

Part II. The Inform Recipe Book

Chapter 1: How to Use The Recipe Book	Chapter 8: Vehicles, Animals and Furniture
Chapter 2: Adaptive Prose	Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
Chapter 3: Place	Chapter 10: Physics: Substances, Ropes, Energy and Weight
Chapter 4: Time and Plot	Chapter 11: Out Of World Actions and Effects
Chapter 5: The Viewpoint Character	Chapter 12: Typography, Layout, and Multimedia Effects
Chapter 6: Commands	Chapter 13: Testing and Publishing
Chapter 7: Other Characters	

-  Start reading here: §1.1. Preface
-  Part I. Writing with Inform
-  Indexes of the examples and definitions

Chapter 1: How to Use The Recipe Book

§1.1. Preface; §1.2. Acknowledgements; §1.3. Disenchantment Bay; §1.4. Information Only

-  Contents of *The Inform Recipe Book*
-  Chapter 2: Adaptive Prose
-  Indexes of the examples

§1.1. Preface

The Inform Recipe Book is one of two interlinked books included with Inform 7: a comprehensive collection of examples, showing the practical use of Inform. The other book is **Writing with Inform**, a systematic manual for the software. If you are reading this within the Inform application, you will see that the Recipe Book pages are on "yellow paper", while the manual is on "white paper".

The Recipe Book assumes that the reader already knows the basics covered in Chapters 1 and 2 of *Writing with Inform*: enough to get simple projects working in the Inform application. It's helpful, but not necessary, to have some familiarity with the main ingredients of Inform. For instance, the reader who can play and test the following source text, and who can take a guess at what it ought to do, should be fine:

"The Power of the Keys"

Afterlife is a room. "Fluffy white clouds gather round you here in the afterlife." The Pearly Gates are a door in Afterlife. "The Pearly Gates - large, white, wrought-iron and splendidly monumental - stand above you." Heaven is a room. The Gates are above the Afterlife and below Heaven.

St Peter is a man in the Afterlife. "St Peter, cheery if absent-minded, studies his celestial clipboard."

Before going through the Pearly Gates:

say "St Peter coughs disarmingly. 'If you'd read your Bible,' he says, 'you might recall Revelation 21:21 saying that the twelve gates were twelve pearls, each gate being made from a

single pearl. I really don't know why people keep imagining it like the entrance to some sort of public park - oh, well. In you go."";
end the story.

Test me with "enter gates".

The *Recipe Book* is not a tutorial - it offers advice and examples to crib from, not theory or systematic teaching. The examples here are provided with the express intention that authors cut and paste useful passages into their own works, modifying as they go. This is an excellent way to get things working quickly.

In the traditional saying: good programmers write good code, but great programmers steal it. (Appropriately enough, nobody seems to know who said this first.) For the avoidance of any doubt - the example text is here to be taken, and this infringes no copyright, and requires no acknowledgement. So steal at will. The examples are a part of Inform itself, and as such, they are available to anyone who accepts the Inform licence.

Many programming languages for conventional computing, such as C, come with elaborate libraries of ready-written code - so elaborate, in fact, that they often need much larger manuals than the language itself, and can be hard to learn. Even expert programmers typically use only a small part of what is available in such libraries, giving up on the rest as too complex to use, or too difficult to find out about, or not quite what they need.

The designers of Inform chose not to go down this road. Rather than providing a general system for liquids (say), which would have to be a quite complicated and opaque program, Inform provides a choice of examples showing how to get different effects. The writer can read the text which achieves these effects, and can simply cut and paste whatever might be useful, and rewrite whatever is not quite wanted.

The wider community of Inform writers has made a great wealth of material available in the form of Extensions, too, and under a Creative Commons Attribution licence requiring only a namecheck: we don't cover the Extensions in this book, because it would grow far too long and be a constant labour to maintain, but it's well worth seeing what is out there.

★ See **Acknowledgements** for a chance to try out the cross-referencing links in the *Recipe Book* - click on the red asterisk or the name of the destination to go there

-
-  Start of Chapter 1: How to Use The Recipe Book
 -  Onward to §1.2. Acknowledgements
 -  Example 1:  **About the examples** An explanation of the examples in this documentation, and the asterisks attached to them. Click the heading of the example, or the example number, to reveal the text.
 -  Example 12:  **Midsummer Day** A few sentences laying out a garden together with some things which might be found in it.
-

§1.2. Acknowledgements

David Fisher's "Past raif topics" pages on the Interactive Fiction Wiki were an invaluable tool during the early design of these examples, as they catalog an enormous assortment of implementation problems encountered by IF authors over the past fifteen years.

Thanks also go to Nick Montfort for several conversations during the development of Inform: these inspired a number of ideas about how the author should be able to control the textual output of a story, and suggested specific problem areas to work on.

Jeff Nyman provided extensive feedback about using *Writing with Inform* in workshops of aspiring IF authors from both programming and conventional fiction writing backgrounds. His observations about the concerns of conventional writers first encountering IF were especially useful, and had a great influence on the organization of the *Recipe Book*. While the results may not meet all the needs he identified, we hope to have taken a step in the right direction.

A few examples were contributed by denizens of rec.arts.int-fiction: Jesse McGrew, Jon Ingold, Mike Tarbert, Eric Rossing, and Kate McKee offered such elegant implementations of various tasks that we have folded their contributions (with permission) into the *Recipe Book*.

Finally, these pages owe much to the questions and suggestions of Inform users on rec.arts.int-fiction and ifMUD.

 [Start of Chapter 1: How to Use The Recipe Book](#)

 [Back to §1.1. Preface](#)

 [Onward to §1.3. Disenchantment Bay](#)

§1.3. Disenchantment Bay

"Disenchantment Bay" is a simple work of IF used as a running example in Chapter 3 of *Writing with Inform* - not so much a tutorial as a convenient hook on which to hang some demonstrations of the basics. Because the resulting examples only use basic features and in the most straightforward way, they make for uninteresting "recipes" - so they are not included in the *Recipe Book* proper. But some readers might like to have all twelve stages of the example gathered on a single page: this is that page.

- ↑ Start of Chapter 1: How to Use The Recipe Book
 - ← Back to §1.2. Acknowledgements
 - Onward to §1.4. Information Only
 - ↓ Example 14: ★ **Disenchantment Bay 1** A running example in this chapter, Disenchantment Bay, involves chartering a boat. This is the first step: creating the cabin.
 - ↓ Example 15: ★ **Disenchantment Bay 2** Disenchantment Bay: creating some of the objects in the cabin's description.
 - ↓ Example 17: ★ **Disenchantment Bay 3** Disenchantment Bay: adding a view of the glacier.
 - ↓ Example 18: ★ **Disenchantment Bay 4** Disenchantment Bay: fleshing out the descriptions of things on the boat.
 - ↓ Example 20: ★ **Disenchantment Bay 5** Disenchantment Bay: adding the door and the deck to our charter boat.
 - ↓ Example 23: ★ **Disenchantment Bay 6** Disenchantment Bay: locking up the charter boat's fishing rods.
 - ↓ Example 25: ★ **Disenchantment Bay 7** Disenchantment Bay: making the radar and instruments switch on and off.
 - ↓ Example 30: ★ **Disenchantment Bay 9** Disenchantment Bay: enter the charter boat's Captain.
 - ↓ Example 33: ★ **Disenchantment Bay 10** Disenchantment Bay: things for the player and the characters to wear and carry.
 - ↓ Example 34: ★ **Disenchantment Bay 11** Disenchantment Bay: making a holdall of the backpack.
 - ↓ Example 28: ★★ **Disenchantment Bay 8** Disenchantment Bay: a pushable chest of ice for the boat.
 - ↓ Example 37: ★★★ **Disenchantment Bay 12** A final trip to Disenchantment Bay: the scenario turned into a somewhat fuller scene, with various features that have not yet been explained.
-

§1.4. Information Only

One last preliminary: a handful of the examples do not show how to do anything at all, but are really sidebars of information. Those examples are gathered below, since they contribute nothing by way of recipes.

- ↑ Start of Chapter 1: How to Use The Recipe Book
 - ← Back to §1.3. Disenchantment Bay
 - Onward to Chapter 2: Adaptive Prose: §2.1. Varying What Is Written
 - ↓ Example 415: ★ **About Inform's regular expression support** Some footnotes on Inform's regular expressions, and how they compare to those of other programming languages.
 - ↓ Example 223: ★★★ **Formal syntax of sentences** A more formal description of the sentence grammar used by Inform for both assertions and conditions.
 - ↓ Example 243: ★★★ **Mathematical view of relations** Some notes on relations from a mathematical point of view, provided only to clarify some technicalities for those who are interested.
 - ↓ Example 244: ★★★ **Graph-theory view of relations** Some notes on relations from the point of view of graph theory.
 - ↓ Example 399: ★★★ **Backus-Naur form for rules** The full grammar Inform uses to parse rule definitions, in a standard computer-science notation.
-

Examples from Chapter 1: How to Use The Recipe Book

-  Start of this chapter
-  Chapter 2: Adaptive Prose
-  Indexes of the examples

1

Example About the examples

WI

An explanation of the examples in this documentation, and the asterisks attached to them. Click the heading of the example, or the example number, to reveal the text.

This is the first of about 400 numbered examples. In a few cases, such as this one, they provide a little background information, but almost all demonstrate Inform source text. The techniques demonstrated tend to be included either because they are frequently asked for, or because they show how to achieve some interesting effect.

The same examples are included in **both** of the books of documentation, but in a different order: in *Writing with Inform*, they appear near the techniques used to make them work; in *The Inform Recipe Book*, they are grouped by the effects they provide. For instance, an example called "Do Pass Go", about the throwing of a pair of dice, appears in the "Randomness" section of *Writing with Inform* and also in the "Dice and Playing Cards" section of *The Inform Recipe Book*. Clicking the italicised WI and RB buttons at the right-hand side of an example's heading switches between its position in each book.

Many computing books quote excerpts from programs, but readers have grown wary of them: they are tiresome to type in, and may only be fragments, or may not ever have been tested. The authors of Inform have tried to avoid this. All but two dozen examples contain entire source texts. A single click on the paste icon  (always placed just left of the double-quoted title) will write the complete source text into the Source panel. All that is then required is to click the Go button, and the example should translate into a working game.

In most cases, typing the single command TEST ME will play through a few moves to show off the effect being demonstrated. (You may find it convenient to create a "scratch" project file for temporary trials like this, clearing all its text and starting again with each new test.)

As part of the testing process which verifies a new build of Inform, each example in turn is extracted from this documentation, translated, played through, and the resulting transcript mechanically checked. So the examples may even work as claimed. But the flesh is weak, and there are bound to be glitches. We would welcome reports, so that future editions can be corrected.

Each example is loosely graded by difficulty: if they were exercises in a textbook, the asterisks would indicate how many marks each question scores. As a general rule:

-  - A simple example, fairly easily guessed.
-   - A complicated or surprising example.
-    - An example needing detailed knowledge of many aspects of the system.
-     - A complete scenario, containing material not necessarily relevant to the topic being demonstrated.

In general, the main text of *Writing with Inform* tries never to assume knowledge of material which has not yet appeared, but the trickier examples almost always need to break this rule.

Example Midsummer Day

12 A few sentences laying out a garden together with some things which might be found in it. WI

"Midsummer Day"

East of the Garden is the Gazebo. Above is the Treehouse. A billiards table is in the Gazebo. On it is a trophy cup. A starting pistol is in the cup. In the Treehouse is a container called a cardboard box.

Test me with "up / x box / d / e / x table / x cup / x pistol / get cup".

14 **★ Example Disenchantment Bay 1** WI
A running example in this chapter, Disenchantment Bay, involves chartering a boat. This is the first step: creating the cabin.

To begin with the title:

"Disenchantment Bay"

There are many Disenchantment Bays across the world, named by eighteenth-century ships' captains - one in Antarctica, another in Tasmania, for instance. The most famous is probably the one where Lewis and Clark's expedition broke through to the Pacific. But ours is the one in Alaska, named in 1791 by a Spanish navigator who had hoped it might lead to the fabled Northwest Passage, and all of this history is beside the point since the game is set in the present day.

The Cabin is a room. "The front of the small cabin is entirely occupied with navigational instruments, a radar display, and radios for calling back to shore. Along each side runs a bench with faded blue vinyl cushions, which can be lifted to reveal the storage space underneath. A glass case against the wall contains several fishing rods.

Scratched windows offer a view of the surrounding bay, and there is a door south to the deck. A sign taped to one wall announces the menu of tours offered by the Yakutat Charter Boat Company."

We might want to start with the glass case.

The Cabin contains a glass case. In the glass case is a collection of fishing rods.

Now Inform will have guessed that the case is a container, but its default idea of a container is something like a bucket: permanently open and not able to be opened and shut. We can change that:

The case is closed, transparent, and openable.

We get a similar set of guesses if we write

The bench is in the cabin. On the bench are some blue vinyl cushions.

Using "some" rather than "a" or "the" tells Inform that the cushions are to be referred to as a plural object in the future. And because of the "on the bench..." phrase, Inform will guess

that the bench is a supporter and that it is fixed in place and cannot be moved from room to room. We do have to tell it that the bench can be sat on, though:

The bench is enterable.

And now a short script, so that if we type TEST ME, we experiment with the case and bench:

Test me with "examine case / get rods / open case / get rods / sit on bench / take cushions / get up"

15

★ Example Disenchantment Bay 2

WI

Disenchantment Bay: creating some of the objects in the cabin's description.

If we compile our last version of the cabin, we get a room where the glass case and the bench are listed separately from the room description, even though they have already been mentioned once. We can prevent this by making the already-mentioned things scenery:

"Disenchantment Bay"

The Cabin is a room. "The front of the small cabin is entirely occupied with navigational instruments, a radar display, and radios for calling back to shore. Along each side runs a bench with faded blue vinyl cushions, which can be lifted to reveal the storage space underneath. A glass case against the wall contains several fishing rods.

Scratched windows offer a view of the surrounding bay, and there is a door south to the deck. A sign taped to one wall announces the menu of tours offered by the Yakutat Charter Boat Company."

The Cabin contains a glass case. In the glass case is a collection of fishing rods. The case is closed, transparent, and openable. The case is scenery.

The bench is in the cabin. On the bench are some blue vinyl cushions. The bench is enterable and scenery. The cushions are scenery.

Generally speaking, it is a good idea to recognize the player's attempts to interact with any objects mentioned in the room description, so we should also provide

Some navigational instruments, some scratched windows, a sign, a radar display, and some radios are scenery in the cabin.

Test me with "examine instruments / x windows / x sign / x display / x radios".

The door and the view will need to be done as well, but they are special cases which we will get to shortly.

As noted, making something scenery also means that the player will be prevented from picking it up and carrying it away. This is sensible, though: if an object can be removed from the room where it first appears, we should be careful about mentioning it in the main room description; otherwise, it will continue to be described as present even when someone has carried it off.

Example Disenchantment Bay 3

Disenchantment Bay: adding a view of the glacier.

Suppose we wanted to have the glacier visible from the Cabin of our boat, and anywhere else we might add to the game:

The view of the Malaspina glacier is a backdrop. It is everywhere. The description is "The Malaspina glacier covers much of the nearby slope, and -- beyond it -- an area as large as Rhode Island."

Example Disenchantment Bay 4

Disenchantment Bay: fleshing out the descriptions of things on the boat.

Currently we have provided objects for most of what is on the boat, but it's not very interesting to look at. We might want to give some more description to these things.

"Disenchantment Bay"

The Cabin is a room. "The front of the small cabin is entirely occupied with navigational instruments, a radar display, and radios for calling back to shore. Along each side runs a bench with faded blue vinyl cushions, which can be lifted to reveal the storage space underneath. A glass case against the wall contains several fishing rods.

Scratched windows offer a view of the surrounding bay, and there is a door south to the deck. A sign taped to one wall announces the menu of tours offered by the Yakutat Charter Boat Company."

The Cabin contains a glass case. In the glass case is a collection of fishing rods. The case is closed, transparent, and openable. The case is scenery.

The bench is in the cabin. On the bench are some blue vinyl cushions. The bench is enterable and scenery. The cushions are scenery.

Some navigational instruments, some scratched windows, a radar display, and some radios are scenery in the cabin.

The description of the instruments is "Knowing what they do is the Captain's job."

The description of the windows is "They're a bit the worse for wear, but you can still get an impressive view of the glacier through them. There were whales earlier, but they're gone now."

The description of the radar is "Apparently necessary to avoid the larger icebergs."

The description of the radios is "With any luck you will not need to radio for help, but it is reassuring that these things are here."

The order in which we define these things is fairly open. We could also define an object so:

A sign is scenery in the Cabin. The description is "You can get half-day and full-day sight-seeing tours, and half-day and full-day fishing trips."

Where "the description" is assumed to refer to the thing most recently defined, if no object is specified.

The view of the Malaspina glacier is a backdrop. It is everywhere. The description is "The Malaspina glacier covers much of the nearby slope, and -- beyond it -- an area as large as Rhode Island."

Test me with "examine sign / examine glacier / examine instruments / examine windows / examine radar / examine radios / take the cushions / take the glacier".

These last two commands show how scenery and backdrops are automatically impossible for the player to take.

20

★ Example Disenchantment Bay 5

WI

Disenchantment Bay: adding the door and the deck to our charter boat.

We mentioned that there is a door out to the deck in our example. The following two sentences will create both the door and the other room:

The cabin door is south of the Cabin and north of the Deck. It is a door and scenery.

Now Inform has constructed a generic room called "Deck" to the south. It has neither a description nor any contents yet, but we could fix that in time. It does have a view of the glacier, though, since we defined the glacier view to be everywhere.

23

★ Example Disenchantment Bay 6

WI

Disenchantment Bay: locking up the charter boat's fishing rods.

It stands to reason that the captain wouldn't let just anyone meddle with his fishing equipment; maybe he keeps that case locked. We could replace the case description with this one, instead:

The Cabin contains a glass case. In the glass case is a collection of fishing rods. The case is closed, transparent, openable, lockable, and locked. The case is scenery. The small silver key unlocks the case.

Now there's a silver key that will unlock it -- though since we haven't said where the key is, the player will never be able to find it in the game. (If we look at the World index, we find "small silver key" right at the bottom, not inside any of the rooms. That is as good as not existing at all -- though we usually use the term "out of play" -- but as we will later see, it is possible to have things initially out of play but brought into existence later on.)

25

★ Example Disenchantment Bay 7

WI

Disenchantment Bay: making the radar and instruments switch on and off.

If we would like the player to be able to turn instrumentation on and off, we could add a line to this effect:

The radar, the instruments, and the radios are devices.

And since the captain is probably not navigating blind, we might also want to say

The radar and the instruments are switched on.

30

★ **Example Disenchantment Bay 9**

Disenchantment Bay: enter the charter boat's Captain.

WI

Now finally we can put a Captain in the boat:

The Captain is a man in the Cabin. "The captain sits at the wheel, steering the boat and occasionally checking the radar readout."

33

★ **Example Disenchantment Bay 10**

Disenchantment Bay: things for the player and the characters to wear and carry.

WI

At this point we can dress both the Captain and the player with some appropriate props:

The captain wears a baseball cap. The description of the cap is "It says, THE WORST DAY FISHING IS BETTER THAN THE BEST DAY WORKING."

The player is carrying a backpack and a bottle of water. The player is wearing a pair of sunglasses. The description of the sunglasses is "The light off the water and the ice does get pretty bright sometimes."

(At present the backpack can't be worn, but see the next version.)

34

★ **Example Disenchantment Bay 11**

Disenchantment Bay: making a holdall of the backpack.

WI

If we wanted, we could make the player's backpack infinitely capacious, so:

The backpack is a player's holdall.

...And now whenever the player character is unable to hold everything, he will automatically stow some of his possessions therein.

This is only useful if the player doesn't have infinite carrying capacity himself, so perhaps we also need

The carrying capacity of the player is 3.

Perhaps mercifully, items which are worn are not counted against the player's carrying capacity. We might want to let him take advantage of that, too:

The backpack is wearable.

This capacity system makes a compromise between the realistic and the absurd: on the one hand, it acknowledges that people can't carry an infinite number of items in their hands, while at the same time providing a sack that can.

Many games will have no use for object-juggling of this kind at all; others will want to be much more rigorous about questions of capacity and volume. Fortunately, it is easy to leave the whole business out by assigning no carrying capacity to anything.

28

★★ Example Disenchantment Bay 8

WI

Disenchantment Bay: a pushable chest of ice for the boat.

We probably do not need a vehicle to ride around our boat, but there might be a heavy ice chest that can only be pushed from room to room:

The ice chest is a closed openable container in the Deck. "A very heavy ice chest sits on the ground." It is fixed in place and pushable between rooms. A quantity of ice is in the chest. The description is "Ready and waiting just in case there's any fish needing to be kept cool."

This anticipates a later chapter, but it would probably be a good idea to hint to the player, if he tries to take the ice chest, that there is another way to move it:

Instead of taking the chest: say "It's too heavy to lift, but you might be able to push it, and just inch it over the frame of the door."

Otherwise, attempts to pick it up will just reply with "That's fixed in place."

37

★★★★ Example Disenchantment Bay 12

WI

A final trip to Disenchantment Bay: the scenario turned into a somewhat fuller scene, with various features that have not yet been explained.

"Disenchantment Bay"

Include Locksmith by Emily Short.

Use scoring.

The Cabin is a room. "The front of the small cabin is entirely occupied with navigational instruments, a radar display, and radios for calling back to shore. Along each side runs a bench with faded blue vinyl cushions[if the compartment is closed], which can be lifted to reveal the storage space underneath[otherwise], one of which is currently lifted to allow access to the storage compartment within[end if]. A glass case against the wall contains several fishing rods.

Scratched windows offer a view of the surrounding bay, and there is a door south to the deck. A sign taped to one wall announces the menu of tours offered by the Yakutat Charter Boat Company."

The Cabin contains a glass case. In the glass case is a collection of fishing rods. Understand "rod" as the collection. The case is closed, transparent, openable, lockable, and locked. The case is scenery. The small silver key unlocks the case.

The bench is in the cabin. On the bench are some blue vinyl cushions. The bench is enterable and scenery. The cushions are scenery.

A storage compartment is an openable closed container. It is part of the bench. Instead of opening the bench, try opening the storage compartment. Instead of closing the bench, try closing the storage compartment. Instead of pushing or pulling or turning the cushions, try opening the storage compartment. Understand "space" as the storage compartment.

Some nets and a Coke are in the compartment. Understand "net" as the nets. The description of the nets is "They must have something to do with fish as well. Really, you're just here for the sights." The nets are a container.

Some navigational instruments, some scratched windows, a radar display, and some radios are scenery in the cabin. The radar, the instruments, and the radios are devices. The radar and the instruments are switched on.

A screen is part of the radar. The description of the screen is "[if the radar is switched on]Phantom lights move across the screen.[otherwise]The screen is dark.[end if]". Instead of doing something other than examining to the screen, say "It's not good for much but looking at."

The Captain is a man in the Cabin. "The captain sits at the wheel, steering the boat and occasionally checking the radar readout." The captain wears a baseball cap. The description of the cap is "It says, THE WORST DAY FISHING IS BETTER THAN THE BEST DAY WORKING." The captain carries the silver key. The description of the captain is "[The captain] is wearing [a list of things worn by the captain][if the captain carries something] and carrying [a list of things carried by the captain][end if]."

The description of the instruments is "Knowing what they do is the Captain's job." Instead of doing something other than examining to the instruments in the presence of the Captain: say "The Captain glares at you. Clearly you are not welcome to do that."

The description of the windows is "They're a bit the worse for wear, but you can still get an impressive view of the glacier through them. There were whales earlier, but they're gone now." Understand "window" as the windows.

The description of the radar is "Apparently necessary to avoid the larger icebergs."

The description of the radios is "With any luck you will not need to radio for help, but it is reassuring that these things are here."

A sign is scenery in the Cabin. The description is "You can get half-day and full-day sight-seeing tours, and half-day and full-day fishing trips."

The view of the Malaspina glacier is a backdrop. It is everywhere. The description is "The Malaspina glacier covers much of the nearby slope, and -- beyond it -- an area as large as Rhode Island." Understand "view of the surrounding bay" or "surrounding bay" as the view.

The cabin door is south of the Cabin and north of the Deck. It is a door and scenery. The description of the Deck is "The whole back half of the boat is open, allowing you to view the surroundings without intervening windows -- if you can stand the cold."

The ice chest is a closed openable container in the Deck. "A very heavy ice chest sits on the ground." It is fixed in place and pushable between rooms.

A quantity of ice is in the Deck. "All around the boat bob chunks of glacier ice." Understand "glacier ice" as the quantity. The description is "Curiously cooled into funny-shaped chunks." The printed name of the quantity is "glacier ice".

Instead of taking the quantity of ice when the player is not carrying the nets:
if the quantity of ice is handled, continue the action;
say "You are having a hard time fishing out the ice with your bare hands."

Instead of taking the quantity of ice when the player is carrying the nets:
if the quantity of ice is handled or the quantity of ice is in the nets, continue the action;
now the quantity of ice is in the nets;
say "You scoop up the ice with the net."

Instead of taking the chest: say "It's too heavy to lift, but you might be able to push it, and just inch it over the frame of the door."

The player is carrying a backpack. The player is wearing a pair of sunglasses. The description of the sunglasses is "The light off the water and the ice does get pretty bright sometimes."

The backpack is a player's holdall. The carrying capacity of the player is 3. The backpack is wearable.

Instead of asking the Captain for the key:
say "'Sure, you can -- well, get me a drink first, would you?'"

Instead of asking the Captain for the key when the Captain is carrying a cold Coke and the Captain is carrying the key:
move the key to the player;
say "'Here, knock yourself out.'"

Instead of asking the Captain for the key when the Captain is not carrying the key: say "'I already gave it to you. You didn't lose it, did you?'"

Heat is a kind of value. The heats are cold, cool, room temperature, and warm.

A beverage is a kind of thing. A beverage can be open or closed. A beverage can be openable or unopenable. A beverage is always edible and openable. A beverage has a heat. A beverage is usually warm. The Coke is a beverage. The beer is a beverage. The beer is in the backpack.

Instead of giving or showing a beer to the Captain:
say "'I don't drink on the job, thanks,' he says. 'You can help yourself if you want it, though.'"

Instead of giving or showing a Coke to the Captain:
say "'It needs chilling,' the Captain remarks, disgruntled."

Instead of giving or showing a cold Coke to the Captain:
move the Coke to the Captain;
increase the score by 2;
say "'Ah, thank you,' he says. How he can drink an iced soda on a day like today is an open question, but Alaskans are special."

Every turn when the quantity of ice is in the ice chest:
repeat with item running through beverages in the ice chest:

let the current heat be the heat of the item;
if the current heat is not cold, now the heat of the item is the heat before the current heat.

Before printing the name of a beverage (called the drink):
say "[heat of the drink]".

Understand the heat property as describing a beverage.

The maximum score is 5.

After taking the fishing rods:
end the story finally;
increase the score by 3;
say "Success is yours! (Now if only you knew anything about fishing.)"

Test me with "test first / test second / test third".

Test first with "x captain / open case / i / ask captain for the key / give beer to captain / open bench / x nets / get nets / get coke / give coke to captain".

Test second with "s / open chest / drop nets / get glacier ice / get nets / get glacier ice / g / put glacier ice in chest / get coke / put coke in chest".

Test third with "x coke / g / g / g / get coke / n / give coke to captain / ask captain for key / open case / get rods".

415



Example About Inform's regular expression support

WI

Some footnotes on Inform's regular expressions, and how they compare to those of other programming languages.

There is not really any unanimity about what regular expression language is. The unix tools `sed` and `grep` extend on Kleene's original grammar. Henry Spencer's `regex` library extended on this again, and was a foundation for Perl, but Perl once again went further. Philip Hazel's `PCRE`, despite the name Perl Compatible Regular Expressions, makes further extensions still, and so on.

Inform's regular expressions are modelled on those of Perl, as the best de facto standard we have, but a few omissions have been inevitable. Inform's `regex` matcher must occupy source code less than one hundredth the size of `PCRE`, and it has very little memory. Inform aims to behave exactly like Perl except as follows:

(i) Inform allows angle brackets as synonymous with square brackets, for reasons explained above. This means literal angle brackets have to be escaped as `"\<"` and `"\>"` in Inform regular expressions, which is unnecessary in Perl.

(ii) Inform only has single-line mode, not multiline mode: this removes need for the mode-switches `"(?m)"` and `"(?s)"` and the positional markers `"\A"` and `"\Z"`. Multiline mode is idiosyncratic to Perl and is a messy compromise to do with holding long files of text as single strings, yet treating them as lists of lines at the same time: this would not be sensible for Inform. Similarly, because there is no ambiguity about how line breaks are represented in Inform strings (by a single `"\n"`), initial newline convention markers such as `"(*ANYCRLF)"` are unsupported.

(iii) The codes "\a", "\r", "\f", "\e", "\0" for alarm, carriage return, form feed, escape and the zero character are unsupported: none of these can occur in an Inform string.

(iv) Inform does not allow characters to be referred to by character code (whereas Perl allows "\036" for an octal character code, "\x7e" for a hexadecimal one, "\cD" for a control character). This is because we do not want the user to know whether text is internally stored as ZSCII or Unicode.

(v) Inform's character class "\p" (and its negation "\P") have no equivalent in Perl, and Inform's understanding of "\w" is different. Perl defines this as an upper or lower case English letter, underscore or digit, which is good for programming-language identifiers, but bad for natural language - for instance, "é" is not matched by "\w" in Perl, but unquestionably it appears in words. Inform therefore defines "\w" as the negation of "\s" union "\p".

(vi) Inform supports only single-digit grouping numbers "\1" to "\9", whereas Perl allows "\10", "\11", ...

(vii) POSIX named character ranges are not supported. These are only abbreviations in any case, and are not very useful. (Note that the POSIX range "[:punct:]", which is supposedly for punctuation, includes many things we do not want to think of that way - percentage signs, for instance - and so "\p" has a more natural-language-based definition.)

(viii) Character classes can be used inside ranges, so that "<\da-f>" is legal, but not as ends of contiguous runs, so that "<\d-X>" is not legal. (As reckless as this is, it is legal in Perl.)

(ix) For obvious reasons, escapes to Perl code using the notation "(?{...})" are unsupported, and so is the Perl iteration operator "\G".

(x) Perl's extended mode "(?x)", a lexical arrangement which allows expressions to be expanded out as little computer programs with comments, is unsupported. It would look awful syntax-coloured in the Inform interface and is not a style of coding to be encouraged.

Inform further does not support the Python extension of named subexpression groups, nor the Java extension of the possessive quantifier "++". There was only so much functionality we could squeeze in.

As verification of Inform's matching algorithm, we took the Perl 5 source code's notorious "re-test.txt" set of 961 test cases, removed the 316 using features unsupported by Inform (220 tested multiline mode, for instance), and ran the remaining 645 cases through Inform. It agrees with Perl on 643 of these: the two outstanding are -

(i) Perl is able to match "\^(a\1?)\{4}\$" against "aaaaa" but Inform is not - Inform's backtracking is not as good when it comes to repetitions of groupings which are recursively defined. (Note that the optional "\1" match refers to the value of the bracketed expression which contains it, so that the interpretation is different on each repetition. Here to match we have to interpret "?" as 0, 0, 1, 0 repeats respectively as we work through the "\{4}".)

(ii) Perl matches "\((<a-c>)b*\2)*" against "ababbbcbc" finding the match "ababb", whereas Inform finds the match "ababbbcbc". This is really a difference of opinion about whether the outer asterisk, which is greedy, should be allowed three matches rather than two if to do so requires the inner asterisk, which is not greedy, to eat more than it needs on one of those three matches.

Case (i) is a sacrifice to enable Inform's back-tracking to use less memory. Case (ii) simply seems unimportant.

223



Example Formal syntax of sentences

WI

A more formal description of the sentence grammar used by Inform for both assertions and conditions.

An entire grammar for the whole mass of Inform would not be linguistically interesting: it contains many convenient wordings which are not really part of a grand pattern. Inform does, however, have a formal notion of a Sentence, a grammatical structure which we shall call S. It is almost true that conditions ("if the flowerpot is on the wall") have the same grammar as assertions ("The flowerpot is on the wall") and "now" phrases ("now the flowerpot is on the wall"). All three use the S grammar, so we could define an assertion as "S.", say that "if S", "while S", "when S" and so on are conditions, and say that "now S" defines the "now" declaration.

Grammatical sentences do not necessarily make sense, of course. Many perfectly grammatical assertions in fact give rise to problem messages:

The wicker basket is not in the kitchen. (*Unhelpfully negative.*)
The wicker basket has been in the kitchen. (*Talks about a time which never existed.*)
The wicker basket is full. (*Full of what? Too vague.*)
The wicker basket is the ginger cat. (*Demonstrably false.*)

Whereas the first three, at least, would be sensible as conditions. So saying that assertions are "just like" conditions is a little misleading: what they have in common is S, the underlying grammar they each use as a starting-point.

To define S, we break it up into subsidiary structures. The most important is the Description Phrase (DP), examples of which include "the red basket", "somewhere lighted" and "an empty open container". Clearly sentences include DPs, but they also include other ingredients. The general pattern used in Inform is very simple:

1. S = DP + VP
2. VP = Verb + DP

where VP is another structure, the Verb Phrase. For instance:

S (The horseman wears a riding helmet)
= DP (The horseman) + VP (wears a riding helmet)
VP (wears a riding helmet)
= Verb (wears) + DP (a riding helmet)

In that example, the Verb was the single word "wears". More generally, Inform allows a Verb to include adverbs and prepositions, to be negated, and to come in any of four tenses, so the following are all valid examples of Verb in our grammar:

wore
carries
is carried by
had not been inside

Although we are not going through the definition of Description Phrases in detail, it is worth noticing how "which" and "who" behave:

- 3a. DP = DP + which + VP
- 3b. DP = DP + who + VP

Thus "an open container which is in the Ballroom" can be broken down as:

DP (an open container) + which + VP (is in the Ballroom)

To understand compounds like "something in a container", we have to invent a new grammatical structure for "in a container" and similar: let's call this a Relative Phrase (RP).

- 4. DP = DP + RP

Thus "an open container in the Ballroom" is DP (an open container) + RP (in the Ballroom). Relative Phrases have two different forms:

- 5a. RP = Preposition + DP
- 5b. RP = Participle + DP

so that "in a container" is an example of 5a. An example of 5b would be

RP (worn by Mr Darcy) = Participle (worn by) + DP (Mr Darcy)

That is nearly it, but not quite: we must go back to the "almost" in the statement above that assertions and conditions "almost" have the same grammar S. The difference arises from a curious irregularity in English called subject-verb inversion (see the *Oxford English Grammar* at 3.22F), whereby assertions can be reversed but not conditions. For instance,

In the Garden is a sunflower.

This does not follow the pattern S = DP + VP, because "in the garden" is not a DP: indeed, it is not a noun at all. To make sense of this sentence, Inform reverses it to "A sunflower is in the Garden", which does indeed follow DP + VP. Hence the final rule:

- 6 (assertions only). S = RP + Verb + DP

So the condition "if in the garden is a sunflower..." fails because rule 6 does not apply to the grammar for conditions: while occasional poetic uses of subject-verb inversion do turn up in conditions ("If On A Winter's Night A Traveller", say), they are rare in ordinary English usage, and illegal in Inform. That completes the S grammar, so to recap:

- 1. S = DP + VP
- 2. VP = Verb + DP
- 3a. DP = DP + which + VP
- 3b. DP = DP + who + VP
- 4. DP = DP + RP
- 5a. RP = Preposition + DP
- 5b. RP = Participle + DP
- 6 (assertions only). S = RP + Verb + DP



Some notes on relations from a mathematical point of view, provided only to clarify some technicalities for those who are interested.

Inform uses the term "relation" in a broader sense than mathematics. Properly speaking, the term "relation" in its mathematical sense only applies to the case where the domain for the left and right objects are the same: for simplicity's sake, let us talk only about the case where they are.

In mathematics, the properties most often looked for in a relation are that it should be:

(a) Reflexive: $A \Leftrightarrow A$ for every A . This is not especially useful for Inform, and seldom appears in practical examples.

(b) Symmetric: $A \Leftrightarrow B$ if and only if $B \Leftrightarrow A$. Generally, Inform relations are not symmetric, but there are two important cases which are:

Meeting relates people to each other.
Marriage relates one person to another.

These are automatically symmetric, so that to assert one way round is to assert the other as well.

(c) Transitive: $A \Leftrightarrow B$ and $B \Leftrightarrow C$ means that $A \Leftrightarrow C$ as well. Again, Inform relations are not generally transitive. In many relations, there can be long chains of things, each perhaps related to the one in front and the one behind, so that there is some indirect sense in which the two ends of the chain are connected to each other: but they are not related as such. For instance, a journey across the map might pass through ten rooms, each adjacent to the last and next, but the two ends would not themselves be adjacent. The concept we need is the "transitive closure" of the original relation, defined as the smallest transitive relation including the original. If R is a relation between "things", then the following:

TC relates a thing (called A) to a thing (called B) when the number of steps via R from A to B is greater than 0.

is the transitive closure of R . In particular,

Accessibility relates a room (called A) to a room (called B) when the number of moves from B to A is greater than 0. The verb to be accessible from means the accessibility relation.

calculates the transitive closure of adjacency. Here, though, the way we normally understand "accessible from" suggests that it would be better to write:

Accessibility relates a room (called A) to a room (called B) when the number of moves from B to A is at least 0.

which is reflexive as well as transitive. The usefulness of Inform's "next step via R from A to B " construction, in a wide variety of settings, reflects the importance of transitivity as an idea.

A relation which has all three properties of being reflexive, symmetric and transitive is called an "equivalence relation". (If all the map connections are two-way, then the accessibility relation above is symmetric and therefore a full equivalence relation: but if not, it may not be.) Inform has a special construction for making equivalence relations:

Nationality relates people to each other in groups.

This language - "in groups" - relies on the standard theorem that every equivalence relation on a set naturally defines a partition of that set, and vice versa. The "groups" referred to are what are normally called "equivalence classes". (Inform does little with these equivalence classes: it might be interesting to do so, in effect forming quotient kinds.)

244



Example Graph-theory view of relations

WT

Some notes on relations from the point of view of graph theory.

One way to look at a relation is to regard it as a directed graph: that is, a collection of things ("vertices") with arrows drawn between them ("edges"). We write our items A, B, C, ... on a piece of paper: then, if A relates to B, we draw an arrow pointing from A to B, and so on. If we made this drawing for the adjacency relation, we would more or less have reconstructed the map (or at least a simplified one which does not care about precise directions, like the famous diagram of the London Underground). But the drawing can be made for any relation. If we define:

Suspecting relates various people to one person.

then, in the corresponding graph, each "vertex" will have at most one arrow leading away from it - though there could be many (or none) leading towards. Conversely, a one-to-various relation produces a graph where each vertex has at most one arrow coming in. A one-to-one relation means that the picture consists of some vertices on their own, with no arrows, a few perhaps with looped arrows leading from and to themselves, and then a collection of pairs joined by arrows. On the other hand, a various-to-various relation is just a free-for-all, with no restrictions on the arrows. The relations:

Meeting relates people to each other.

Marriage relates one person to another.

always have the property of working both ways round, and these are easiest to visualise by forgetting the direction of the arrows, so that they just become lines joining the vertices.

Inform uses a different algorithm for finding routes ("the next step via R from A to B") in each of these cases, and internally it stores relations in different formats in the different cases, because it makes a big difference to the efficiency of Inform to minimise the storage required for a relation and the time taken to explore it.

All the cases are benign except for "various to various" - the most useful - and for its closely related symmetrical version, "relates... to each other". Inform cannot afford to assume that the relation will be "sparse" (that is: that no vertex will have more than a certain number of arrows, or that the total number of arrows will be small), because it has no idea how the arrows will come and go during play. It therefore uses 1 bit of storage for each pair of objects. This sounds harmless, but if there are 200 rooms, there are 40,000 pairs of rooms, which means a 5000-byte allocation of storage (plus a handful of bytes as overhead). Gratuitous various-to-various relations are therefore not a good idea.

There is a standard algorithm for calculating shortest paths through a directed graph, but Inform does not always use it, because there is not always memory to store the required matrix of partial results. Inform's slow method, likely to be used on the Z-machine, requires a storage overhead which is equal to the number of vertices, not the square of that number, but the worst-case running time can be bad: if there are N vertices, and the diameter of graph

(the longest distance between vertices) is D , then the running time is proportional to D times N . The worst case in finding routes from A to B is when almost every vertex can reach B , some across long trails, but A cannot. In the case of finding routes across the game's map, this must be multiplied further by the number of possible directions - usually 16.

This does not sound too awful, but if one is trying to find (say) "the most distant room from A ", that means a further loop and now the running time will be D times N squared. Extension writers will need to be careful of this kind of thing: it is easy to write highly cool prototypes which work terribly slowly on larger, more realistic maps.

399



Example Backus-Naur form for rules

WT

The full grammar Inform uses to parse rule definitions, in a standard computer-science notation.

Backus-Naur form, or BNF, is a standard notation used by computer scientists to specify more or less precisely what the valid programs are for a given programming language. It tends to provide a good description for a language such as C or Pascal, where contextual rules are limited, but the authors of Inform are doubtful that it is such a good tool for a natural-language system. For those who are interested, though, the following gives a formal specification for Inform's rules.

```
<rule> ::=
  Definition : A/an <kind> is <new adjectival name> if/unless <definition>
  | <preamble> : <phrases>
  | <preamble> , <phrase> (* only allowed for a few cases: see below)

<definition> ::=
  <condition>
  | its/his/her/their <value property name> is/are <value> or less/more
  | : <phrases>

<preamble> ::=
  To <phrase template>
  | To decide if/whether <phrase template>
  | To decide which/what <kind of value> is <phrase template>
  | This is the <rule name>
  | [[A] Rule for] <circumstances> [(this is the <rule name>)]

<circumstances> ::=
  At <time>
  | When <event name>
  | [<placement>] <rulebook reference> [while/when <condition>] [during <scene
name>]

<rulebook reference> ::=
  <rulebook name> [about/for/of/on/rule] [<action pattern>]
  | <object-based-rulebook name> [about/for/of/on/rule] [<description>]

<placement> ::=
  a/an
  | [the] first
  | [the] last

<phrases> ::=
  <phrase>
```

| <phrases> ; <phrase>

The following examples show how Inform breaks down some typical rules using the system above:

```
<rule> = At 2:09 PM: increase the score by 2; say "Progress!"
  <preamble> = At 2:09 PM
    <circumstances> = At 2:09 PM
      At
      <time> = 2:09 PM
    :
  <phrases> = increase the score by 2; say "Progress!"
    <phrase> = increase the score by 2
  ;
  <phrase> = say "Progress"
```

```
<rule> = Instead of eating the ostrich during Formal Dinner (this is the cuisine rule), say "It's greasy!"
  <preamble> = Instead of eating the ostrich during Formal Dinner (this is the cuisine rule)
    <circumstances> = Instead of eating the ostrich during Formal Dinner
      <rulebook reference> = Instead of eating the ostrich
        <rulebook name> = Instead
          of
          <action pattern> = eating the ostrich
        during
        <scene name> = Formal Dinner
      (
        this
        is
        the
        <rule name> = cuisine rule
      )
    ,
  <phrases> = say "It's greasy!"
    <phrase> = say "It's greasy!"
```

```
<rule> = After printing the name of a container: say "!"
  <preamble> = After printing the name of a container
    <circumstances> = After printing the name of a container
      <rulebook reference> = After printing the name of a container
        <object-based-rulebook name> = After printing the name
          of
          <description> = a container
    :
  <phrases> = say "!"
    <phrase> = say "!"
```

(*) The colon dividing a rule preamble from its definition can be replaced by a comma only if the preamble begins with the words "Instead of", "Before", "After", "Every turn" or "When", and if the definition consists only of a single phrase.

Chapter 2: Adaptive Prose

§2.1. *Varying What Is Written*; §2.2. *Varying What Is Read*; §2.3. *Using the Player's Input*

-  Contents of *The Inform Recipe Book*
-  Chapter 1: How to Use The Recipe Book
-  Chapter 3: Place
-  Indexes of the examples

§2.1. Varying What Is Written

Before getting to actual recipes, many recipe books begin with intimidating lists of high-end kitchen equipment (carbon-steel pans, a high-temperature range, a Provencal shallot-grater, a set of six pomegranate juicers): fortunately, readers who have downloaded Inform already have the complete kitchen used by the authors. But the other traditional preliminaries, about universal skills such as chopping vegetables, boiling water and measuring quantities, do have an equivalent.

For us, the most basic technique of IF is to craft the text so that it smoothly and elegantly adapts to describe the situation, disguising the machine which is never far beneath the surface. This means using text substitutions so that any response likely to be seen more than once or twice will vary.

M. Melmoth's Duel demonstrates three basic techniques: an ever-changing random variation, a random variation changing only after the player has been absent for a while, and a message tweaked to add an extra comment in one special case. (Random choices can be quite specifically constrained, as **Ahem** shows in passing.) **Fifty Ways to Leave Your Larva** and **Fifty Times Fifty Ways** show how a generic message can be given a tweak to make it a better fit for the person it currently talks about. **Curare** picks out an item carried by the player to work into a message, trying to make an apt rather than random choice. **Straw Into Gold** demonstrates how to have Inform parrot back the player's choice of name for an object.

Another reason to vary messages is to avoid unnatural phrasing. **Ballpark** turns needlessly precise numbers - another computerish trait - into more idiomatic English. (Likewise **Numberless**, though it is really an example demonstrating how to split behaviour into many cases.) **Prolegomena** shows how to use these vaguer quantifiers any time Inform describes a group of objects (as in "You can see 27 paper clips here.").

Blink, a short but demanding example from the extreme end of *Writing with Inform*, shows how the basic text variation mechanisms of Inform can themselves be extended. **Blackout** demonstrates text manipulation at a lower level, replacing every letter of a room name with "*" when the player is in darkness.

Inform's included extension Complex Listing allows us more control over the order and presentation of lists of items.

For how to change printed text to upper, lower, sentence, or title casing, see **Rocket Man**.

- ⬆ Start of Chapter 2: Adaptive Prose
- ⬅ Back to Chapter 1: How to Use The Recipe Book: §1.4. Information Only
- ➡ Onward to §2.2. Varying What Is Read
- ⬇ Example 71: ★ **Fifty Ways to Leave Your Larva** Using text substitution to make characters reply differently under the same circumstances.
- ⬇ Example 169: ★ **Ahem** Writing a phrase, with several variant forms, whose function is to follow a rule several times.
- ⬇ Example 174: ★ **Numberless** A simple exercise in printing the names of random numbers, comparing the use of "otherwise if...", a switch statement, or a table-based alternative.
- ⬇ Example 178: ★ **M. Melmoth's Duel** Three basic ways to inject random or not-so-random variations into text.
- ⬇ Example 249: ★ **Olfactory Settings** Some adaptive text for smelling the flowers, or indeed, anything else.
- ⬇ Example 252: ★ **Responsive** Altering the standard inventory text for when the player is carrying nothing.
- ⬇ Example 343: ★ **Prolegomena** Replacing precise numbers with "some" or other quantifiers when too many objects are clustered together for the player to count at a glance.
- ⬇ Example 345: ★ **Wesponses** Parser messages that are delivered with a speech impediment.
- ⬇ Example 413: ★ **Rocket Man** Using case changes on any text produced by a "to say..." phrase.
- ⬇ Example 419: ★ **Blackout** Filtering the names of rooms printed while in darkness.
- ⬇ Example 439: ★ **Curare** A phrase that chooses and names the least-recently selected item from the collection given, allowing the text to cycle semi-randomly through a group of objects.
- ⬇ Example 454: ★ **Blink** Making a "by atmosphere" token, allowing us to design our own text variations such as "[one of]normal[or]gloomy[or]scary[by atmosphere]".
- ⬇ Example 245: ★★ **Fun with Participles** Creating dynamic room descriptions that contain sentences such as "Clark is here, wasting time" or "Clark is here, looking around" depending on Clark's idle activity.
- ⬇ Example 246: ★★★ **Variety** Suppose we want all of our action responses to display some randomized variety. We could do this by laboriously rewriting all of the response texts, but this example demonstrates an alternative.
- ⬇ Example 247: ★★★ **Variety 2** This builds on the Variety example to add responses such as "You are now carrying the fedora" that describe relations that result from a given verb, as alternate responses.
- ⬇ Example 250: ★★★ **History Lab** We create phrases such as "the box we took" and "the newspaper Clark looked at" based on what has already happened in the story.
- ⬇ Example 251: ★★★ **Relevant Relations** An example of how to create room descriptions that acknowledge particular relations using their assigned verbs, rather than by the heavily special-cased code used by the standard library.
- ⬇ Example 60: ★★★ **Ballpark** A new "to say" definition which allows the author to say "[a number in round numbers]" and get verbal descriptions like "a couple of" or "a few" as a result.
- ⬇ Example 72: ★★★ **Fifty Times Fifty Ways** Writing your own rules for how to carry out substitutions.
- ⬇ Example 248: ★★★ **Narrative Register** Suppose we want all of our action responses to vary depending on some alterable quality of the narrator, so that sometimes they're slangy, sometimes pompous or archaic.



Example 305:  **Straw Into Gold** Creating a Rumpelstiltskin character who is always referred to as "dwarf", "guy", "dude", or "man" -- depending on which the player last used -- until the first time the player refers to him as "Rumpelstiltskin".

§2.2. Varying What Is Read

Making the printed text adapt to circumstances only makes half of the conversation graceful: the other half is to allow the player's commands to have a similar freedom. The things the player can refer to should always respond to the names which would seem natural to the player. Inform provides a variety of techniques for understanding words always, or only under certain conditions; and, if need be, we can also get direct access to what the player has typed in order to examine it with regular expressions. (This last resort is rarely necessary.)

First Name Basis shows how to assign names to things or to kinds of thing - if, for instance, we want the player to be able to refer to any man as "man" or "gentleman":

Understand "man" or "gentleman" as a man.

We may also sometimes want to give names that are specifically plural, as in

A duck is a kind of animal. Understand "birds" as the plural of duck.

or

Understand "birds" as the plural of the magpie.

Vouvray demonstrates.

A common challenge arises when two objects have names that overlap or are related, and we wish Inform to choose sensibly between them: for instance, a cigarette vs. a cigarette case. If a word should apply to something only as part of a phrase (e.g., "cigarette" alone should never refer to the cigarette case) we can manage the situation as follows:

The case is a closed openable container. The printed name is "cigarette case".
Understand "cigarette case" as the case.

Because "cigarette" here appears only as part of the phrase "cigarette case", it will be understood only in that context; the conflict with the bare cigarette will not arise.

As a variant, we may want one object only to take precedence over another in naming. If we wanted the player to be allowed to refer casually to the cigarette case as "cigarette" when (and only when) the cigarette itself is not in view, we could add

Understand "cigarette" as the case when the cigarette is not visible.

Tricks which consider the visibility of other objects can be bad for performance if used widely; but for adding finesse to the treatment of a few items, they work very well.

(There may still arise cases where the player uses a name which can legitimately refer to two different things in view. To deal with this situation, we may want the Does the player mean...

rules, explained in the chapter on Understanding; and to change the way the story asks for clarification, see the two activities Asking which do you mean and Clarifying the parser's choice of something.)

Names of things which contain prepositions can also be tricky because Inform misreads the sentences creating them: **Laura** shows how some awkward cases can be safely overcome.

A more difficult case is to ensure that if we change the description or nature of something in play, then the names we understand for it adapt, too. "Understand... when..." can be all that's needed:

Understand "king" as Aragorn when we have crowned Aragorn.

Or, similarly, if we want some combination of categories and characteristics to be recognized:

Understand "giant" as a man when the item described is tall.

"The item described" here refers to the thing being named. "...when" can even be useful in defining new commands, and **Quiz Show** demonstrates how to ask open-ended questions that the player can answer only on the subsequent turn.

Properties can also be matched without fuss:

Tint is a kind of value. The tints are green, aquamarine and darkish purple. The wallpaper is fixed in place in the Hotel. The wallpaper has a tint. Understand the tint property as describing the wallpaper.

This allows EXAMINE AQUAMARINE WALLPAPER if, but only if, it happens to be aquamarine at the moment. Relationships can also be matched automatically:

A box is a kind of container. The red box is a box in the Toyshop. Some crayons are in the red box. Understand "box of [something related by containment]" as a box.

which recognises BOX OF CRAYONS until they are removed, when it reverts to plain BOX only.

Greater difficulty arises if, using some variable or property or table to mark that a bottle contains wine, we print messages calling it "bottle of wine". We are then honour-bound to understand commands like TAKE BOTTLE OF WINE in return, not to insist on TAKE BOTTLE. Almost all "simulation" IF runs in to issues like this, and there is no general solution because simulations are so varied.

A converse challenge arises when we want to *avoid* understanding the player's references to an object under some or all circumstances. This is relatively uncommon, but does sometimes occur. For this situation, Inform provides the "privately-named" property, as in

The unrecognizable object is a privately-named thing in the Kitchen.

Here "privately-named" tells Inform not to understand the object's source name automatically. It is then up to us to create any understand lines we want to refer to the object, as in

Understand "oyster fork" as the unrecognizable object when the etiquette book is read.

Of course, if we need an object that the player is never allowed to refer to at all, we can just make this privately-named and then not provide any understand lines at all.

A final source of difficulty is that by default Inform truncates words to nine letters before attempting to identify them. This is no problem in most circumstances and is likely to go unnoticed -- until we have two very long words whose names are nearly identical, such as "north-northwest exit" and "north-northeast exit". (To make matters worse, a punctuation mark such as a hyphen counts as two letters on its own.)

When we are compiling for Glulx, the limit is easily changed with a single line, setting the constant called `DICT_WORD_SIZE`. For instance, if we wanted to raise the limit to 15, we would write

Use `DICT_WORD_SIZE` of 15.

When compiling for the Z-machine, the solution is harder. **North by Northwest** shows how to use the reading a command activity to pre-process very long names, rendering them accessible to the parser again.

Inform also allows the player to refer to the most recently seen objects and people as IT, HIM, HER, and so on. It sets these pronouns by default, but there are times when we wish to override the way it does that. **Pot of Petunias** shows off a way to make Inform recognize an object as IT when it would not otherwise have done so.

★ See **Liquids** for a resolution of this bottle-of-wine issue

★ See **Using the Player's Input** for an example (*Mr. Burns' Repast*) in which a fish can be called by any arbitrary word as long as it ends in the letters *-fish*

★ See **Memory and Knowledge** for a way to refer to characters whom the player knows about but who aren't currently in the room

★ See **Clarification and Correction** for ways to improve guesses about what the player means

★ See **Alternatives To Standard Parsing** for several esoteric variations on the default behavior, such as accepting adverbs anywhere in the command, and scanning the player's input for keywords

- ⬆ Start of Chapter 2: Adaptive Prose
 - ⬅ Back to §2.1. Varying What Is Written
 - ➡ Onward to §2.3. Using the Player's Input
 - ⬇ Example 11: ★ **First Name Basis** Allowing the player to use different synonyms to refer to something.
 - ⬇ Example 323: ★ **Quiz Show** In this example by Mike Tarbert, the player can occasionally be quizzed on random data from a table; the potential answers will only be understood if a question has just been asked.
 - ⬇ Example 325: ★ **Pot of Petunias** Responding sensibly to a pot of petunias falling from the sky.
 - ⬇ Example 19: ★★ **Laura** Some general advice about creating objects with unusual or awkward names, and a discussion of the use of printed names.
 - ⬇ Example 43: ★★★ **Vouvray** Adding synonyms to an entire kind of thing.
 - ⬇ Example 375: ★★★ **North by Northwest** Creating additional compass directions between those that already exist (for instance, NNW) -- and dealing with an awkwardness that arises when the player tries to type "north-northwest". The example demonstrates a way around the nine-character limit on parsed words.
-

§2.3. Using the Player's Input

We may sometimes want to capture specific words the player has used and then feature that text elsewhere in the story.

Terracottissima Maxima demonstrates using text to describe objects; **Mr. Burns' Repast** lets the player refer to a fish by any of a number of names, and changes the way the fish is described as a result.

More specialized effects are also possible: **Xot** shows how to collect the player's erroneous input and store the command line to be printed back later. **Igpay Atinlay** shows how to parrot the player's command back in pig Latin form.

★ See **Animals** for a dog which the player can re-name

★ See **Traits Determined By the Player** for a way to let the player name the player character

- ⬆ Start of Chapter 2: Adaptive Prose
 - ⬅ Back to §2.2. Varying What Is Read
 - ➡ Onward to Chapter 3: Place: §3.1. Room Descriptions
 - ⬇ Example 421: ★ **Igpay Atinlay** A pig Latin filter for the player's commands.
 - ⬇ Example 422: ★★ **Mr. Burns' Repast** Letting the player guess types for an unidentifiable fish.
 - ⬇ Example 317: ★★★ **Terracottissima Maxima** Flowerpots with textual names that might change during play.
 - ⬇ Example 381: ★★★★ **Xot** Storing an invalid command to be repeated as text later in the game.
-

Examples from Chapter 2: Adaptive Prose

- ⬆ Start of this chapter
- ➡ Chapter 3: Place
- ⬇ Indexes of the examples

71

★ Example **Fifty Ways to Leave Your Larva**

WI

Using text substitution to make characters reply differently under the same circumstances.

We can use these substitutions to put together fairly complicated variations within a single piece of text:

"Fifty Ways to Leave Your Larva"

The Beekeeper's Palace is a room. Wasp is a woman in the palace. Drone is a man in the palace.

Instead of kissing someone:

say "[denial], [insult]! [boast]!";

In this context, [denial] is understood to refer to the denial property of the noun -- but we could spell it out with "denial of the noun" if we wanted to.

A person has some text called denial. The denial of a person is usually "Stand back". The denial of Drone is "You forget yourself"

A person has some text called insult. The insult of a person is usually "Grasshopper". The insult of Wasp is "Larva".

A person has some text called boast. The boast of a person is usually "I am ferocious". The boast of Drone is "I have ferocious allies".

And then it would be trivial to insert further rules using these responses:

Instead of attacking someone:
say "Get away, [insult]!"

Test me with "kiss wasp / hit wasp / hit drone / kiss drone".

169

★ Example Ahem

WI

Writing a phrase, with several variant forms, whose function is to follow a rule several times.

As we see in the example here, it is possible to use slashed variations in more than one place in a phrase, and to offer a number of separate forms. The main rule of thumb to remember is that value inputs for the phrase should always be separated by some text; so

To do/follow (chosen rule - a rule) exactly/precisely/just/-- (N - a number)
time/times:

....

would cause a problem when we tried to call it with

follow the throat-clearing rule 2 times.

In general, we probably don't need to make our phrase definitions quite so flexible as this, but it's a good idea to account for "a" vs. "the", and for the possibility of using singular and plural forms, especially when writing extensions or other source to be shared.

"Ahem"

To do/follow (chosen rule - a rule) exactly/precisely/just (N - a number)
time/times:

repeat with index running from 1 to N:
follow chosen rule.

This is the throat-clearing rule:

say "Ahem," says [a random visible person who is not the player]."

After waiting:

do the throat-clearing rule just one time.

Instead of listening:

follow the throat-clearing rule precisely three times.

Instead of smelling:

follow the throat-clearing rule exactly 2 times.

Chateau Marmont is a room. Tom, Jack, Zsa-Zsa, and Wilma-Faye are people in the Chateau. Zsa-Zsa and Wilma-Faye are women.

Example Numberless

A simple exercise in printing the names of random numbers, comparing the use of "otherwise if...", a switch statement, or a table-based alternative.

"Numberless"

The Rambling Warren is a room.

When play begins:

```
let N be a random number between 1 and 5;
if N is 1:
  say "N is one.";
otherwise if N is 2:
  say "N is two.";
otherwise if N is 3:
  say "N is three.";
otherwise:
  say "N is more than the number of your toes."
```

The final "otherwise" here will fire only if none of the earlier conditions applies; we could leave it out and print nothing in the case that N is 4 or 5.

The more compact way to do this is to create a list of values that our number could match; in many programming languages this is called a switch statement. For example:

When play begins:

```
let Y be a random number between 6 and 10;
if Y is:
  -- 6: say "Six is the magic number!";
  -- 7: say "The number of the day is seven!";
  -- otherwise: say "Today's magic number is boring."
```

As a final option, we can use a construction we've seen only briefly before now: a table. The use of tables will be explained more fully in their own chapter, but here we see in brief that we can assign a number of values to one column of a table and then use that table to look up output:

When play begins:

```
let X be a random number between 11 and 14;
if X is a number listed in the Table of Switching, say "[output entry][paragraph
break]";
otherwise say "X is greater than the number of your noses!"
```

Table of Switching

number output

11 "X is eleven!"
12 "X is twelve!"
13 "X is thirteen!"

Test me with "z".

As we shall see, things other than text can be stored in tables, so we could also use a table as a way to look up objects or even rules to carry out.

178

★ Example M. Melmoth's Duel

WI

Three basic ways to inject random or not-so-random variations into text.

"M. Melmoth's Duel"

Saint-Germain-des-Prés is a room. "Haunt of artists, of the coffee-drinking sort, and of cafés, of the artist-haunted sort, you once again find yourself outside M. Melmoth's hotel. Today [one of]the recently-fallen rain runs down the gutters of the 6th[or]sunlight glints even off the blackened windows of the Abbey[or]crowds of vulgar children play chase around the lampposts[at random], and you long to be indoors."

The Hôtel d'Alsace is inside from Saint-Germain-des-Prés. "Typical. Oscar writes you a letter announcing his own imminent demise - 'My wallpaper and I are fighting a duel to the death. One or other of us has got to go.' - and then you get there and he's out, no doubt procuring paint the colour of absinthe, if he isn't procuring the painter."

Tint is a kind of value. The tints are green, aquamarine and darkish purple.

The wallpaper is fixed in place in the Hôtel. The wallpaper has a tint. "In this light, the wallpaper has a distinctly [tint of the wallpaper] wash. [if the tint of the wallpaper is darkish purple]You particularly dislike purple.[end if]"

Before going to the Hôtel: now the wallpaper is a random tint.

After going from the Hôtel, say "You leave, shaking your head. But within twenty-four hours, you are back, as you always knew you would be."

Test me with "in / out / look / in / out / look".

249

★ Example Olfactory Settings

WI

Some adaptive text for smelling the flowers, or indeed, anything else.

While this isn't very interesting as IF, it runs through most of the adaptive-text tricks.

"Olfactory Settings"

The Doghouse is a room. "Not so much a place as a state of being."

The player carries a ticket to the opera, some papers, and a bouquet of flowers. The bouquet is ambiguously plural.

Instead of eating something inedible, say "[The noun] [don't] seem likely to agree with [us] at all. [We] ['d be wiser to leave [regarding the noun][them] alone."

Instead of touching something: say "[regarding the noun][Those] [are] all prickly."

Instead of smelling something: say "[Our] nose [regarding nothing][are] too weak to get much smell from [regarding the noun][those]."

Instead of smelling the bouquet: say "[regarding the noun][They][re] lovely."

Instead of tasting something:
say "Whew, [regarding the noun][are] [those] ever nasty!"

Test me with "x ticket / eat it / eat them / touch it / touch them / smell it / smell them / taste it / taste them / x papers / eat it / eat them / touch them / smell them / taste them / x bouquet / eat it / eat them / touch them / smell them / taste them".

252

Example Responsive

WI

Altering the standard inventory text for when the player is carrying nothing.

The most straightforward way to alter the response text for something in the standard rules is to select the Index tab, then select Actions, then pick the particular action whose text we'd like to alter. Under action details, there will be icons that look like speech bubbles.

Clicking on the speech bubble will show what the current response text is, and give us an option called "set".

If we click "set", this will automatically paste in the response name that we need to change. We can put this inside a "when play begins" rule in order to make that change take effect from the start of the game, like so:

"Responsive"

An Anonymous B613 Cell is a room. "There isn't much to see in this bare room. What there is, you've already seen sometime in the last twenty years."

When play begins:
now print empty inventory rule response (A) is "[We] [have] absolutely nothing."

Test me with "i".

★ Example Prolegomena

Replacing precise numbers with "some" or other quantifiers when too many objects are clustered together for the player to count at a glance.

Room descriptions often make the player character out to be a bit of a savant, able to count whole stacks of items at a glance: "You see 27 paper clips here."

We can adjust this behavior to our liking, though, with the printing a number.. activity, as follows:

"Prolegomena"

The Editor's Office is a room. The desk is a supporter in the Editor's Office.

A red pencil is a kind of thing. 12 red pencils are on the desk.

A letter is a kind of thing. 12 letters are on the desk. Understand "correspondence" as a letter.

Rule for printing the plural name of a letter:
if the listing group size is greater than 7, say "correspondence";
otherwise say "letters".

Rule for printing a number of something (called the target) when the listing group size is greater than 7:
say "[one of]some [or]various [or]an assortment of [at random]";
carry out the printing the plural name activity with the target.

This general rule can of course be overridden by more specific ones; for instance, if we want to take the opportunity to comment on the viewpoint character's appetite for instruments of correction:

Rule for printing a number of red pencils (called the target) when the listing group size is greater than 10:
carry out the printing the plural name activity with the target;
say " in nearly-sufficient quantity".

Test me with "get two letters / look / get a pencil / i / get pencil / g / g / look / i / get all / i".

★ Example Responses

Parser messages that are delivered with a speech impediment.

If we want to change individual responses to player action, then the best thing to do is to use the response facility to modify those selections, as shown in the chapter on Responses.

However, suppose what we want is to give the parser a speech impediment that slightly alters all of the responses it issues. For this purpose, we might need to do a bit of text replacement...

"Responses"

The Office is a room. Barry Kripke is a man in the Office.

The response inhibition is initially false.

Rule for issuing the response text of a response (called R) when response inhibition is false:

```
now response inhibition is true;
let output be "[text of R]";
now response inhibition is false;
replace the text "r" in output with "w";
replace the text "R" in output with "W";
say "[output]".
```

Test me with "i / x barry / listen / waffle / jump"

Notice that this doesn't affect the printed names of objects in the room description or other kinds of output text -- only those that are issued by the response mechanism.

It would also not work to try to give our parser a nervous personality by simply adding "Um, " to the beginning of each response, because responses are not guaranteed to be full standalone sentences. For example, we could imagine writing

Rule for issuing the response text of a response (called R) when response inhibition is false:

```
now response inhibition is true;
let output be "Um, [text of R]";
now response inhibition is false;
say "[output]".
```

but here is what the room description would say as a result:

Office
Um, You Um, can see Barry KripkeUm, here

413

★ Example Rocket Man

WI

Using case changes on any text produced by a "to say..." phrase.

We can now change the case of any text produced by a "to say..." phrase. This is often useful when we would like to make use of a standard say phrase in some new context. Say, for instance, that we would like to "[is-are the list...]" in a context that needs the first letter to be capitalized.

We could write a new say phrase, such as "to say is-are the list of (N - a description of objects) in sentence capitalization"; but there is an easier way, and that is to set a

text variable to the output of the to say phrase, and then print that text in the case of our choice.

For example:

```
"Rocket Man"
```

```
Instead of going somewhere from the spaceport when the player carries something:
```

```
let N be "[is-are the list of things carried by the player] really suitable gear to take to the moon?" in sentence case;  
say "[N][paragraph break]".
```

```
The Spaceport is a room. North of the Spaceport is the Rocket Launch Pad. The player carries a stuffed bear, a chocolate cookie, and a book.
```

```
The description of the book is "It is entitled [italic type]Why Not To Take [sentence cased inventory] To The Moon[roman type]."
```

```
To say sentence cased inventory:
```

```
let N be "[a list of things carried by the player]" in title case;  
say "[N]".
```

```
Test me with "n / x book".
```

419

★ Example Blackout

WI

Filtering the names of rooms printed while in darkness.

In this example, we want the names of rooms to be asterisked out if the player wanders around without the benefit of a candle. We can do this by treating the room names as text, then replacing every letter:

```
"Blackout"
```

```
Tiny Room is a dark room. Absurdly Long-Named Room is a dark room. It is west of Tiny Room.
```

```
The Candle Factory is north of Tiny Room. It contains a beeswax candle. The beeswax candle is lit.
```

```
Rule for printing the name of a dark room:
```

```
let N be "[location]";  
replace the regular expression "\w" in N with "*";  
say "[N]".
```

```
Test me with "w / look / e / n / get candle / s / w".
```

Notice that the hyphen in the Absurdly Long-Named Room does not get replaced. We could replace even that, if we liked, with

replace the regular expression "\S" in N with "*";

which would catch every character that is not a space.

439

★ Example Curare

WI

A phrase that chooses and names the least-recently selected item from the collection given, allowing the text to cycle semi-randomly through a group of objects.

"Curare"

A thing has a number called the last use. The last use of a thing is usually 0.

Definition: a thing is old if its last use is 12 or less.

The actual number chosen in this definition is pretty much irrelevant: the main thing is that we want to establish relative values. The lower the "last use" number of an item, the older that item should be understood to be, as we see here:

To decide which thing is cyclically random (collection - a description of objects):
let choice be the oldest member of the collection;
now the last use of the choice is the turn count;
decide on choice.

This phrase will select, from the collection of objects passed to it, the one that has been mentioned least recently. This means that if we consult it repeatedly about the same collection, it will begin to cycle predictably; but if new items are added to the collection, it will mention these first before returning to the previous cycle. Now we can use this:

After taking inventory:

say "You stare morosely at [the cyclically random thing carried by the player], wondering what you're ever going to find to do with it."

We could have said "You stare morosely at [the oldest thing carried by the player]" here, but doing so would not have set the "last use" property correctly, so we would not get the cycling behavior that we're looking for.

The Evidence Room is a room. Some shelves are scenery supporters in the Evidence Room. A box is a kind of container which is open and not openable. On the shelves is a box. It contains a deformed bullet and a driver's license.

The player carries a steel fish hook, a Chinese passport, a tube of synthetic curare, and an envelope full of Euros.

Test me with "i / i / i / i / i / i / get all from box / i / i / i".

Example Blink

Making a "by atmosphere" token, allowing us to design our own text variations such as "[one of]normal[or]gloomy[or]scary[by atmosphere]".

Suppose we are writing a game in which the mood of the piece changes, and we would like to have lots of descriptions that vary according to its current state. We might in that case want to create our own "by atmosphere" token, to control text variations, like this:

"Blink"

Atmosphere is a kind of value. The atmospheres are normal, melancholy, and creepy.

The current atmosphere is an atmosphere that varies.

To say by atmosphere -- ending say_one_of with marker I7_SOO_ATM:
(- {-close-brace} -).

Since we're operating within the untyped Inform 6, we can make use of the fact that kinds of value are (internally) just constants, enumerated in the same order in which they were originally defined. In other words, "normal" at the I6 level translates to 1, "melancholy" to 2, and "creepy" to 3; so we can return the value of the current atmosphere, and thereby select option 1, 2, or 3:

```
Include (-
[ I7_SOO_ATM oldval count;
  if (count < (+ current atmosphere +)) return count;
  return (+ current atmosphere +); ];
-)
```

And that concludes the hard part. Now to test that it works:

The Flat is a room. "A small [one of]but cozy[or]depressing[or]imprisoning[by atmosphere] flat. Outside the window, the sun is [one of][or][or]apparently [by atmosphere]shining and there is a brisk breeze through the leaves of the birch trees. [one of]It would be quite nice weather for a walk[or]The rest of the world apparently has no appreciation of what you suffer[or]It all looks deceptively normal[by atmosphere]."

Instead of waiting when the current atmosphere is normal:

```
say "Everything stretches wide and flat for just a moment, as though all the
world around you were painted on a thin rubber sheet that is being [italic
type]stretched[roman type]. Then it snaps back into place, leaving your ears
ringing. But that little glitch was enough to warn you. Someone is tampering with
space-time again. Someone very close by.";
now the current atmosphere is creepy.
```

Test me with "look / z / look".



Example Fun with Participles

Creating dynamic room descriptions that contain sentences such as "Clark is here, wasting time" or "Clark is here, looking around" depending on Clark's idle activity.

Mostly the Standard Rules use verbs adapted to finite forms ("he jumped", "we take the hammer", and so on). But Inform can also produce participles to describe actions that are ongoing: "he is carrying the fedora" or "taking the hammer..."

In this example, we give non-player characters actions to perform and then have Inform dynamically describe what they're doing when the player chooses to look.

We start by establishing the idea that a verb can describe a particular action:

"Fun with Participles"

Section 1 - Descriptive Functionality

Describing relates various verbs to various action names. The verb to describe means the describing relation.

To look around is a verb. The verb look around describes the looking action.

To stand about is a verb. The verb stand about describes the waiting action. To look bored is a verb. The verb look bored describes the waiting action. To waste time is a verb. The verb waste time describes the waiting action.

To jump is a verb. To leap is a verb. To pirouette is a verb. The verb jump describes the jumping action. The verb leap describes the jumping action. The verb pirouette describes the jumping action.

Now we need to give every character some sort of idle activity. By default, we'll have people just be waiting, but allow for that idle activity to change into something more interesting if the player has told them to do something else.

A person has an action name called the current idle. The current idle of a person is usually the waiting action.

Rule for writing a paragraph about someone (called chosen person) when a verb describes the current idle of the chosen person:

say "[The chosen person] [are] here, [present participle of a random verb that describes (the current idle of the chosen person)]."

Instead of someone doing something:

now the current idle of the person asked is (the action name part of the current action);
continue the action.

A persuasion rule:
persuasion succeeds.

Section 2 - Scenario

Lab is a room. The fedora is a wearable thing in the Lab. Clark is a man in the Lab.

And just to give past participles a test-drive as well, let's make Clark a bit of a drama king:

After Clark doing something when a verb describes (the action name part of the current action):

say "Fine, have it your way!" Clark exclaims. 'But I have [past participle of a random verb that describes (the action name part of the current action)] for the last time!";

rule succeeds.

Test me with "look / Clark, jump / look / Clark, look / look / Clark, wait".

246



Example Variety

WI

Suppose we want all of our action responses to display some randomized variety. We could do this by laboriously rewriting all of the response texts, but this example demonstrates an alternative.

Verbs can be related to other things by relations. We've seen that it's possible for a verb to "mean" a relationship. But we can also create a relation between verbs and actions. For instance, we can tell Inform that "take", "get", and "acquire" are all valid ways to describe the action of taking, and then allow it to pick a verb randomly to describe whatever action just occurred.

"Variety"

Section 1 - Descriptive Functionality

Describing relates various verbs to various action names. The verb to describe means the describing relation.

To take is a verb. To acquire is a verb. To get is a verb.

The verb take describes the taking action. The verb acquire describes the taking action. The verb get describes the taking action.

To drop is a verb. To put down is a verb. To discard is a verb. The verb drop describes the dropping action. The verb put down describes the dropping action. The verb discard describes the dropping action.

To sniff is a verb. To smell is a verb. The verb sniff describes the smelling action. The verb smell describes the smelling action.

To jump is a verb. To leap is a verb. To pirouette is a verb. The verb jump describes the jumping action. The verb leap describes the jumping action. The

verb pirouette describes the jumping action.

After an actor doing something when the noun is nothing and a verb describes (the action name part of the current action) (this is the apply random verbs to describing nounless actions rule):

say "[The actor] [verb rendering applied to a random verb that describes (the action name part of the current action)].";
rule succeeds.

After an actor doing something to something when a verb describes (the action name part of the current action) (this is the apply random verbs to describing actions rule):

say "[The actor] [verb rendering applied to a random verb that describes (the action name part of the current action)] [the noun].";
rule succeeds.

To decide which text is the rendering of (V - verb) (this is verb rendering):
decide on "[adapt V]".

Section 2 - Scenario

Lab is a room. The table is here. The bat and the ball are on the table.

Test me with "get ball / drop ball / get bat / drop bat / smell ball".

247



Example Variety 2

WI

This builds on the Variety example to add responses such as "You are now carrying the fedora" that describe relations that result from a given verb, as alternate responses.

Some of our default actions establish relations between items in the world, and reporting on the relation ("You are now carrying the fedora") can be a valid response alongside reporting on the action itself ("You take the fedora").

To do this, we need to teach Inform explicitly which relations are the results of actions, then check this when reporting on actions:

"Variety 2"

Section 1 - Descriptive Functionality

Describing relates various verbs to various action names. The verb to describe means the describing relation.

Table of Action Results

related action	relation
the taking action	the carrying relation
the wearing action	the wearing relation
the taking off action	the carrying relation

To take is a verb. To acquire is a verb. To get is a verb.

The verb take describes the taking action. The verb acquire describes the taking action. The verb get describes the taking action.

To drop is a verb. To put down is a verb. To discard is a verb. The verb drop describes the dropping action. The verb put down describes the dropping action. The verb discard describes the dropping action.

To sniff is a verb. To smell is a verb. The verb sniff describes the smelling action. The verb smell describes the smelling action.

To jump is a verb. To leap is a verb. To pirouette is a verb. The verb jump describes the jumping action. The verb leap describes the jumping action. The verb pirouette describes the jumping action.

To don is a verb. The verb don describes the wearing action.

To doff is a verb. The verb doff describes the taking off action.

After an actor doing something when the noun is nothing and a verb describes (the action name part of the current action) (this is the apply random verbs to describing nounless actions rule):

```
say "[The actor] [verb rendering applied to a random verb that describes (the
action name part of the current action)].";
rule succeeds.
```

After an actor doing something to something when a verb describes (the action name part of the current action) (this is the apply random verbs to describing actions rule):

```
let current action name be the action name part of the current action;
if a random chance of 1 in 2 succeeds and the current action name is a
related action listed in the Table of Action Results:
  choose a row with the related action of current action name in the Table of
Action Results;
  let R be the relation entry;
  let subject be the actor;
  let chosen object be the noun;
  say "[The subject] [are] now [present participle of a random verb that
means R] [the chosen object].";
else:
  say "[The actor] [verb rendering applied to a random verb that describes
(the action name part of the current action)] [the noun].";
rule succeeds.
```

To decide which text is the rendering of (V - verb) (this is verb rendering):
decide on "[adapt V]".

To say infinitive of (V - a verb): (- {V}(1); -).

To say past participle of (V - a verb): (- {V}(2); -).

To say present participle of (V - a verb): (- {V}(3); -).

Section 2 - Scenario

Lab is a room. The fedora is a wearable thing in the Lab.

Test me with "wear the fedora / take off the fedora / wear fedora / take off fedora".

250



Example History Lab

WI

We create phrases such as "the box we took" and "the newspaper Clark looked at" based on what has already happened in the story.

The examples Variety and Narrative Register show how verbs can be associated with particular actions. Here, we use the same principle so that we can report to the player what was last done to a particular object, either by the player or by someone else.

To do this, we need to use the idea of stored actions from the Advanced Actions chapter.

"History Lab"

Section 1 - Procedure

An object has an action called the last action.

Describing relates various verbs to various action names. The verb to describe means the describing relation.

To take is a verb. The verb take describes the taking action.

To drop is a verb. The verb drop describes the dropping action.

To look at is a verb. The verb look at describes the examining action.

To examine is a verb. The verb examine describes the examining action.

After an actor doing something to something:

if a verb describes the action name part of the current action:

now the indefinite article of the noun is "the";

now the last action of the noun is the current action;

continue the action.

After printing the name of something (called item):

if the last action of the item is not waiting and the last action of the item is not the current action:

let chosen action-name be the action name part of the last action of the item;

let chosen actor be the actor part of the the last action of the item;

if a verb describes the chosen action-name:

let the chosen verb be a random verb that describes the chosen action-name;

say " [if the chosen actor is the player][we][else][chosen actor][end if] [adapt chosen verb in past tense]";

Section 2 - Scenario

Lab is a room. It contains a box. The box contains a newspaper. Clark is a man in the Lab.

A persuasion rule:
persuasion succeeds.

Test me with "x box / look / x newspaper / look / clark, x newspaper / clark, get box / clark, drop box / look / take box / i / smell box / i".

Notice that smelling the box does not change the box's description because we haven't gotten around to defining a smell or sniff verb.

251

☆☆ Example Relevant Relations

WI

An example of how to create room descriptions that acknowledge particular relations using their assigned verbs, rather than by the heavily special-cased code used by the standard library.

Suppose that we wanted authors to be able to indicate which relations should or should not be included in room descriptions, and have the system dynamically honor that instruction.

Inform already knows about verbs for describing supporting, containment, carrying, and wearing, so we could write a set of instructions to handle such cases. To do this, we're using the "writing a paragraph about" activity, which is described in the chapter on activities.

"Relevant Relations"

Section 1 - Procedure

Rule for writing a paragraph about something (called item):

```
now the current paragraph is { };
say "[one of][regarding item]There [are] [an item] here[or][We] [can see] [an
item] here[at random]. [run paragraph on]";
follow the descriptive rules for the item;
repeat with new item running through the current paragraph:
  now the prior named object is nothing;
  if new item is not the item:
    follow the descriptive rules for the new item;
say paragraph break.
```

Rule for writing a paragraph about someone (called chosen person):

```
now the current paragraph is { };
say "[one of][regarding chosen person][The chosen person] [are] here[or][We]
[can see] [a chosen person] here[at random]. [run paragraph on]";
follow the descriptive rules for the chosen person;
repeat with new item running through the current paragraph:
  now the prior named object is nothing;
  if new item is not the chosen person:
    follow the descriptive rules for the new item;
say paragraph break.
```

The descriptive rules are an object-based rulebook.

Definition: a container is see-through:

if it is transparent:

yes;

if it is open:

yes;

no.

A descriptive rule for a see-through container (called item) (this is the describe contents rule):

describe the containment relation for item.

A descriptive rule for a supporter (called item):

describe the support relation for item.

A descriptive rule for a person (called item):

describe the wearing relation for the item.

A descriptive rule for a person (called item):

describe the carrying relation for the item.

The current paragraph is a list of things that varies.

Before printing the name of something (called mentioned target) while writing a paragraph about something:

add the mentioned target to the current paragraph, if absent.

To describe (R - a relation of objects) for (item - a thing):

if a thing to which item relates by R is a thing:

say "[The item with pronoun] [verb rendering applied to a random verb that means R] [the list of things to which item relates by R with indefinite articles].
[run paragraph on]"

To decide which text is the rendering of (V - verb) (this is verb rendering):

decide on "[adapt V]".

To say (T - a thing) with pronoun:

if T is the prior named object:

say "[regarding T][They]";

else:

say "[The T]"

Section 2 - Scenario

The Space Elevator is a room. "Mercifully, there aren't any windows. The ability to see how far up you are would almost certainly make you ill."

The luggage rack is a supporter in the Space Elevator. The suitcase is a closed openable container on the luggage rack. The bouquet is on the luggage rack.

Clark is a man in the Space Elevator. Clark is carrying a box of cupcakes. Clark is wearing a t-shirt. The description of the box of cupcakes is "They're the latest confection from Red Velvet Planet, the Martian bakery."

Persuasion rule: persuasion succeeds.

We can if we like then add alternate names for these relations that will be randomly swapped in some of the time. For instance:

To sport is a verb. The verb to sport means the wearing relation.

To hold up is a verb. The verb to hold up means the support relation.

Test me with "clark, drop the box / look / clark, take the suitcase / look / clark, get bouquet".

One might, hypothetically, imagine going even further than this and simply designating relations as either "important" or "unimportant" -- perhaps changing the relation's designation at runtime. Relations are not themselves allowed to have properties, however.

60



Example Ballpark

WI

A new "to say" definition which allows the author to say "[a number in round numbers]" and get verbal descriptions like "a couple of" or "a few" as a result.

Sometimes it is more sensible to describe numbers roughly than in exact terms. For instance, we might want to have our player perceive "many people" rather than "forty-two people" on entering a room. To achieve this, we might write our own "to say" phrase.

"Ballpark"

To say (count - a number) in round numbers:
repeat through the Table of Numerical Approximation:
if count is less than threshold entry:
say "[approximation entry]";
rule succeeds.

Phrases will be explained more thoroughly in a later chapter, but as we have already seen in the examples, we can make a "To say..." phrase that will allow us to create our own text substitutions. In this case, we are going to replace the specific number with a vaguer one chosen from a chart, so:

Table of Numerical Approximation

threshold	approximation
1	"no"
2	"one"
3	"a couple of"
6	"a few"
11	"some"
21	"many"
1000	"lots and lots of"

The idea here is that we will work our way through the table until we hit a line where the threshold number is higher than the number we want to express, and then print that output: so if we have less than one item, we'll print "no"; if we have more than none but less than two, we'll print "one"; if we have less than three, we'll print "a couple of"; if we have three, four, or five (but not six), we'll print "a few."

A room has a number called the population. The population of a room is usually 0. The description of a room is usually "You observe [population of the location in round numbers] [if population of the location is 1]person [otherwise]people [end if]here."

The Stadium is a room. The Hot Dog Stand is west of the Stadium. The Women's Restroom is south of the Stadium.

The population of the Stadium is 500. The population of the Hot Dog Stand is 3. The population of the Restroom is 750.

Test me with "w / e / s".

72



Example Fifty Times Fifty Ways

WI

Writing your own rules for how to carry out substitutions.

There is only so much we can cram into a text property, so being able to swap in properties is useful but limited. Fortunately, we can also, if we want, create new phrases for how to say things in brackets:

"Fifty Times Fifty Ways"

The Beekeeper's Palace is a room. Wasp is a woman in the palace. Drone is a man in the palace.

A person can be fierce or mellow. Wasp is fierce. Drone is mellow. A person can be calm or angry. A person is usually calm. A person has some text called insult. The insult of a person is usually "Grasshopper". The insult of Wasp is "Larva".

Instead of kissing someone:

say "[denial for the noun], [insult for the noun]! [boast]!";

Now to provide some meaning to these bracketed forms. We'll start with the easy one:

To say boast:

say "I have ferocious allies".

This is a "to say" phrase; we will learn more about phrases in a later chapter, but for now it may be enough to observe that whatever we write after "to say..." becomes a valid substitution in bracketed speech. In this particular case there is no advantage to using the boast token rather than spelling the text out in the quotation, but we might in theory add further instructions to randomize the output, for instance.

To say phrases can be more complex, as well, since we can have them incorporate extra information:

To say insult for (speaker - a person):
if speaker is angry, say "[the insult of the noun]";
otherwise say "small one".

Here where we have (speaker - a person), we are leaving a slot which we can later fill in, madlibs-like, with any person we like. That is why we can write "insult for the noun": we are summoning the To say phrase and telling it to fill in the identity of the unknown speaker with the noun.

This differs from "insult of the noun" in the previous example; in that case, each person had his own insult property, and were merely printing that property out. Here we are actually telling Inform to calculate anew what the insult should be, and giving it some instructions about how to do that.

Our instructions can also get arbitrarily complex:

To say denial for (speaker - a person):
if speaker is calm:
say "You must not";
otherwise if speaker is female:
say "Stand back";
otherwise:
say "You forget yourself".

Instead of attacking someone:
now the noun is angry;
say "'Get away, [insult]!"

Test me with "kiss wasp / hit wasp / kiss wasp / kiss drone / hit drone / kiss drone".

So the effects we can get with text substitutions are quite flexible. We could even, if we wanted, fill in the substitutions by random choice, or by selecting items from a long list or table, should we have so bellicose a set of characters that they cannot make do with one or two insulting remarks apiece.



Suppose we want all of our action responses to vary depending on some alterable quality of the narrator, so that sometimes they're slangy, sometimes pompous or archaic.

As we saw in "Variety", we can associate verbs with particular actions and call them up as needed. If we do that, though, we can also store additional information about those verbs and use that information to select the ideal verb to use in a particular situation.

In this example, we create a table of verbs and their meanings, together with some connotative information. Each time we report an action, we then score all the available verbs to decide which is the most suitable to use at the moment. This allows us to change the narrator's diction change mid-game and have the action descriptions change as well.

Moreover, because we're using adaptive verbs, these responses will automatically inflect properly even if we change the story tense and viewpoint.

"Narrative Register"

Section 1 - Descriptive Functionality

Describing relates various verbs to various action names. The verb to describe means the describing relation.

To take is a verb. To acquire is a verb. To get is a verb. To gain is a verb. To obtain is a verb. To pick up is a verb. To bag is a verb. To procure is a verb. To score is a verb. To grab is a verb. To snag is a verb. To snatch is a verb. To collect is a verb.

To drop is a verb. To put down is a verb. To discard is a verb. To throw away is a verb. To dispose of is a verb. To set down is a verb. To toss aside is a verb. To ditch is a verb. To abandon is a verb. To dump is a verb. To jettison is a verb. To abjure is a verb. To forsake is a verb. To dispense with is a verb.

After an actor doing something to something when a verb describes (the action name part of the current action) (this is the apply random verbs to describing actions rule):

- score the relevant verbs;
- sort the Table of Verb Meanings in reverse relevance order;
- choose row 1 in the Table of Verb Meanings;
- let top score be the relevance entry;
- sort Table of Verb Meanings in random order;
- repeat through the Table of Verb Meanings:
 - if relevance entry is top score:
 - say "[The actor] [verb rendering applied to (word entry)] [the noun].";
 - erase relevance;
 - rule succeeds.

To decide which text is the rendering of (V - verb) (this is verb rendering):
decide on "[adapt V]".

To score the relevant verbs:
repeat through the Table of Verb Meanings:

- if the meaning entry is (the action name part of the current action):
 - increase relevance entry by 1;
 - repeat with chosen connotation running through connotations entry:
 - if the chosen connotation is listed in the current register:
 - increase relevance entry by 1;
 - otherwise:
 - decrease relevance entry by 1.

To erase relevance:
repeat through Table of Verb Meanings:

now relevance entry is 0.

A tonality is a kind of value. The tonalities are pompous, archaic, slangy, upbeat, downbeat.

Connoting relates various verbs to various tonalities. The verb to connote means the connoting relation.

The current register is a list of tonalities that varies. The current register is { }.

When play begins:

repeat through the Table of Verb Meanings:

now the word entry describes the meaning entry;

now relevance entry is 0;

repeat with chosen tone running through the connotations entry:

now the word entry connotes the chosen tone.

Table of Verb Meanings

word	meaning	connotations	relevance (a number)
the verb take	the taking action	{ }	
the verb acquire	the taking action	{ pompous }	
the verb get	the taking action	{ }	
the verb gain	the taking action	{ }	
the verb obtain	the taking action	{ pompous }	
the verb pick up	the taking action	{ }	
the verb bag	the taking action	{ slangy }	
the verb score	the taking action	{ slangy, upbeat }	
the verb procure	the taking action	{ archaic }	
the verb grab	the taking action	{ slangy }	
the verb snag	the taking action	{ slangy }	
the verb snatch	the taking action	{ slangy }	
the verb collect	the taking action	{ }	
the verb discard	the dropping action	{ pompous }	
the verb drop	the dropping action	{ }	
the verb put down	the dropping action	{ }	
the verb toss aside	the dropping action	{ }	
the verb ditch	the dropping action	{ slangy }	
the verb throw away	the dropping action	{ }	
the verb dispose of	the dropping action	{ }	
the verb set down	the dropping action	{ }	
the verb abandon	the dropping action	{ downbeat }	
the verb dump	the dropping action	{ downbeat }	
the verb abjure	the dropping action	{ archaic }	
the verb foresake	the dropping action	{ archaic }	
the verb jettison	the dropping action	{ pompous }	
the verb dispense with	the dropping action	{ pompous }	

Section 2 - Changing Tone Mid-Game

Understand "new tone" as changing the tone. Changing the tone is an action out of world.

Carry out changing the tone:

now the current register is { };

if a random chance of 1 in 4 succeeds:

say "Your narrator will now adopt an ordinary tone.";

rule succeeds;

let rando be a random tonality;

add rando to the current register, if absent;

say "Your narrator will now be [rando]."

Section 3 - Scenario

Lab is a room. The table is here. The bat and the ball are on the table.

Test me with "get ball / drop ball / get bat / drop bat / new tone / get all / drop all / new tone / get all / drop all".

305



Example Straw Into Gold

WI

Creating a Rumpelstiltskin character who is always referred to as "dwarf", "guy", "dude", or "man" -- depending on which the player last used -- until the first time the player refers to him as "Rumpelstiltskin".

"Straw Into Gold"

The Cell is a room. Rumpelstiltskin is an improper-named man in the Cell. Rumpelstiltskin can be identified or unidentified. Rumpelstiltskin is unidentified.

R-name is a kind of value. The R-names are dwarf, guy, dude, and man-thing. Rumpelstiltskin has an R-name. Understand "[R-name]" as Rumpelstiltskin.

Our example is slightly complicated by the fact that "man" is a name already known to Inform, so we can't re-use it as a kind of value. This is possible to work around, though:

Understand "man" as man-thing.

Now we borrow from the Activities chapter to look at the exact wording of the player's command:

After reading a command:

if the player's command includes "[R-name]",
now the R-name of Rumpelstiltskin is the R-name understood;
if the player's command includes "Rumpelstiltskin":
now Rumpelstiltskin is identified;
now Rumpelstiltskin is proper-named.

Rule for printing the name of Rumpelstiltskin when Rumpelstiltskin is unidentified:

if the R-name of Rumpelstiltskin is man-thing:
say "man";
otherwise:
say "[R-name]".

Test me with "x dwarf / x guy / x dude / look / x rumpelstiltskin / look / x man".

11



Example First Name Basis

WI

Allowing the player to use different synonyms to refer to something.

Sometimes we create objects that we want the player to be able to call by different names: a television that should also answer to "tv" and "telly", for instance, or a refrigerator the player might also call "fridge". In this case, we can use instructions like

Understand "tv" and "telly" as the television.

to add extra names to the object we've defined.

"First Name Basis"

The Crew Lounge is a room. "Deliberately spartan: the crew feels weight restrictions here first, so there aren't any chairs, just a few thin pads on the ground."

The holographic projector is a device in the Crew Lounge. "The one major source of entertainment is the holographic projector, a top of the line Mithon 9000, on which you view every beam you can get." Understand "holo" or "holograph" or "Mithon" or "9000" as the projector.

The description of the projector is "[if switched on]The projector is now playing a documentary about the early politics of the Mars colony.[otherwise]The air above the projector is disappointingly clear.[end if]".

(This description is for local color; we will learn more about devices, and conditions like "if switched on", later in this chapter.)

By default, Inform does not understand the names of an object's kind as referring to that object, unless the object has no other name of its own. We can change this, if we like, by defining names that should be applied to everything of a given kind:

Lewis and Harper are men in the Crew Lounge. Understand "man" or "guy" or "chap" or "lad" or "male" as a man. Understand "men" or "chaps" or "lads" or "guys" or "males" as the plural of a man.

The description of Lewis is "A wiry, excitable engineer who just signed aboard last week." The description of Harper is "Harper's a good guy: taciturn when sober, affectionate when drunk, but rarely annoying in either state."

Test me with "x holo / x man / lewis / x guy / harper / turn on projector / x holo projector / get men".

Inform's naming abilities go considerably further, in fact: we can also instruct it to understand words only under certain circumstances, or only when they appear with other words. Fuller details may be found in the chapter on Understanding.

on random data from a table; the potential answers will only be understood if a question has just been asked.

"Quiz Show" by Mike Tarbert

Use scoring.

Answer mode is a truth state that varies.
Current state is a text that varies.

Guessing is an action applying to one topic.
Understand "[text]" as guessing when answer mode is true.

Because of the "...when" part of this line, random text is only treated as an answer when a question is being asked.

Check guessing (this is the default wrong answer rule):
if the topic understood is not a topic listed in the Table of Dates of Statehood:
say "Wrong!";
now answer mode is false.

Carry out guessing a topic listed in the Table of Dates of Statehood:
if state entry is the current state:
say "Correct! ([comment entry], to be exact!);
increase the score by one;
otherwise:
say "Wrong!";
now answer mode is false.

This next rule allows a player to do something other than answer the question, but then makes him wait for another question before answering.

Before doing anything other than guessing:
if answer mode is true:
say "(ignoring the question)[line break]";
now answer mode is false.

Section 2 - Scenario

The Lab is a room. Sam is a man in the lab.

Every turn when the player is in the lab:
if a random chance of 3 in 5 succeeds:
choose a random row in the Table of Dates of Statehood;
say "Sam asks you, 'In what year was [state entry] admitted into the Union?";
now current state is state entry;
now answer mode is true.

Table of Dates of Statehood

State	Topic	Comment
"Florida"	"1845"	"March 3rd"

"Delaware" "1787" "December 7th"
"Hawaii" "1960" "July 4th"

Test me with "1845 / z / z / 1787 / 1792 / z / 1845 / g".

Note that the situation will become a little more complicated if we have two or more identical topics in our trivia list; in that case, we would need to loop through the Table of Dates of Statehood explicitly, and only mark the player wrong if none of the lines were found to match. (See the chapter on Tables for many more ways to manipulate table behavior.)

325

Example Pot of Petunias

WI

Responding sensibly to a pot of petunias falling from the sky.

Suppose we have an object that makes a dramatic entrance on the scene, like so:

"Pot of Petunias"

Wide Open Field is a room. "A big field under a big sky. The clouds are puffy, the trees are handsome."

Some clouds and some trees are scenery in Wide Open Field. The description of the clouds is "That one looks like Yoda's head." The description of the trees is "You've never been much good at botany, so it's anyone's guess what kind they are."

A rock is in Wide Open Field. The description of the rock is "It looks like it's been here from the dawn of time."

The broken flower pot is a thing. The description of the broken flower pot is "It contains the remains of some abused petunias."

At 9:01 am:

move the broken flower pot to the location;
say "Quite unexpectedly, a flower pot falls from the sky and breaks open on the ground. Good thing you weren't standing six inches to the left.";
set pronouns from the broken flower pot.

Test me with "x it / x it / x it".

If we leave out the "set pronouns..." line here, we'll wind up with the following very unsatisfactory end to our test transcript:

Quite unexpectedly, a flower pot falls from the sky and breaks open on the ground. Good thing you weren't standing six inches to the left.

>[3] x it
It looks like it's been here from the dawn of time.



Some general advice about creating objects with unusual or awkward names, and a discussion of the use of printed names.

Occasionally it is useful to give something a printed name because we want to call it something extremely long-winded; give one thing a name that is the subset of the name of something else; or use words such as "with" or "and" that are likely to confuse Inform into thinking that the object name ends before it actually does.

Often it is enough to preface these ambiguously-titled things with "a thing called..." or "a supporter called..." or the like, as here:

South of Spring Rolls is a room called Hot and Sour Soup.

prevents Inform from trying to read "Hot and Sour Soup" as two separate rooms, while

The player carries an orange ticket. The player carries a thing called an orange.

creates two objects instead of the one orange ticket that would result if the second sentence were merely "The player carries an orange."

Really long names can be a bit cumbersome. For example:

The player carries a thing called an incriminating photograph of a woman with blonde hair.

So we might instead give the photograph a printed name:

"Laura"

The City of Angels is a room. The incriminating photograph is carried by the player. The printed name of the incriminating photograph is "incriminating photograph of a woman with blonde hair".

Now we've gotten around any awkwardness with printing the name -- but we also need to understand when the player refers to the photograph. When we define the names of objects under normal circumstances, Inform takes care of this automatically, but if we have especially set the printed name, we must also specially define the appropriate terms for the player to use. For this we need "understand", which will be explained in much more depth in a later chapter:

Understand "woman" or "with" or "blonde" or "hair" or "of" or "a" as the incriminating photograph.

Test one with "x photograph / x incriminating photograph of a woman with blonde hair / x hair / x blonde / x woman with blonde hair / x incriminating photograph of a woman".

That's probably as far as we really need to go, and if you are satisfied with this behavior, there is no need to read on.

One possible objection to this solution is that Inform will accept some nonsensical formulations as applying to the photograph: for instance, it will allow >EXAMINE PHOTOGRAPH OF, >X BLONDE PHOTOGRAPH WOMAN INCRIMINATING, or even >X OF ...though in the case there were two items with "of" names, the game would disambiguate with a question such as "Which do you mean, the incriminating photograph of a woman with blonde hair or the essence of wormwood?"

Traditionally, Inform has tended to be fairly flexible about word order, preferring to err in the direction of leniency. On the other hand, there are times when we need more exacting rules in order to distinguish otherwise similar cases.

Two features allow us to specify more exactly if we so desire. The first is that, if we specify a whole phrase as the name of something, all the words in that phrase are required, in the order given. Thus "Understand "blonde hair" as the photograph" would require that both "blonde" and "hair" be present, and would not recognize >X BLONDE, >X HAIR BLONDE, or >X HAIR.

Second, we can create tokens, such as "Understand "blonde hair" or "hair" as "[hair]", and then use these tokens in match phrases. This saves a good deal of time when we want to specify a number of different but fussy alternatives. So, for instance, here is a drawing that would not respond to >X OF, or >X BROWN EYES, but would respond to >X DRAWING OF MAN WITH BROWN EYES, >X MAN WITH BROWN EYES, and so on:

The drawing is carried by the player. The printed name of the drawing is "drawing of a man with brown eyes".

Understand "eyes" or "brown eyes" as "[brown eyes]". Understand "man" or "man with [brown eyes]" or "brown-eyed man" as "[man]". Understand "[man]" or "drawing of [man]" or "drawing of a [man]" as the drawing.

Test me with "test one / test two".

Test two with "x drawing / x man / x of / x drawing of man / x drawing of a man / x drawing of a man with brown eyes / x drawing of a brown-eyed man / x brown eyes".

Further refinements are possible: the "privately-named" attribute tells Inform not to try to understand the source name of an object at all, so if we write

The purple rabbit is a privately-named thing.

...the player will not be able to refer to it as "purple" or "rabbit" or "purple rabbit".

There are also ways to make names to refer to entire kinds of objects (so "dude" will refer to any man in the game); to specify names that only refer to objects in the plural (so GET PICTURES will pick up several pictures together); to reflect an object's properties (so "red apple" works only as long as the apple is in fact red); or even to refer to the object's relationships to other objects (so "bottle of wine" works only

when wine is indeed in the bottle). All these refinements are discussed in the chapter on Understanding.

43

Example Vouvray

WI

Adding synonyms to an entire kind of thing.

The Understanding chapter lays out ways to change how the player can refer to objects, but we may not want to wait that long for some of the basic features. Here, for instance, is how to add synonyms that the player can use to refer to an entire kind of object:

"Vouvray"

The Wine Emporium is a room. "Set aside, you rather suspect, for tourists: this chamber is barrel-vaulted stone, lined on each side with casks of aging wine. Discarded brochures here and there advertise Wine Tours of the Loire Valley in three different languages, none of them French."

A cask is a kind of thing. A cask is always fixed in place. Understand "cask" or "barrel" as a cask. Understand "casks" or "barrels" as the plural of cask.

The Vouvray cask and the Muscadet cask are casks in the Wine Emporium.

Test me with "get barrels / get barrel / muscadet / x casks / x muscadet cask".

375

Example North by Northwest

WI

Creating additional compass directions between those that already exist (for instance, NNW) -- and dealing with an awkwardness that arises when the player tries to type "north-northwest". The example demonstrates a way around the nine-character limit on parsed words.

Suppose we wanted to add intermediate compass directions such as north-northwest to our game. Because of the limitations of the index map, we won't be able to view these connections on the world map, but we can certainly create them, and use them in route-finding, just like other directions.

Here's how we'd set up such a thing:

"North by Northwest"

Section 1 - Procedure

The north-northwest is a direction. North-northwest has opposite south-southeast. Understand "n-nw" or "nnw" as north-northwest.

The north-northeast is a direction. North-northeast has opposite south-southwest. Understand "n-ne" or "nne" as north-northeast.

The south-southwest is a direction. South-southwest has opposite north-northeast. Understand "s-sw" or "ssw" as north-northeast.

The south-southeast is a direction. South-southeast has opposite north-northwest. Understand "s-se" or "sse" as south-southeast.

The west-northwest is a direction. West-northwest has opposite east-southeast. Understand "w-nw" or "wnw" as west-northwest.

The east-northeast is a direction. East-northeast has opposite west-southwest. Understand "e-ne" or "ene" as east-northeast.

The west-southwest is a direction. West-southwest has opposite east-northeast. Understand "w-sw" or "wsw" as west-northwest.

The east-southeast is a direction. East-southeast has opposite west-northwest. Understand "e-se" or "ese" as east-southeast.

A complication arises because we reach the 9-character limit: Inform truncates the names of objects to nine characters before trying to understand them. To make matters worse, the hyphen (and other punctuation marks) count as two letters. So both north-northwest and north-northeast will get truncated to "north-no", and be indistinguishable when the player types them.

When we are compiling for Glulx, the limit is easily changed with a single line, setting the constant called `DICT_WORD_SIZE`. For instance, if we wanted to raise the limit to 15, we would simply write "Use `DICT_WORD_SIZE` of 15."

If we're compiling to the Z-machine, however, we'll have to resort to some manipulation of the player's command. The general solution is that when the player's name for an object is going to have to be longer than we can correctly read, we can substitute an unambiguous abbreviation for the thing the player typed. In this case, it will be simplest and most efficient always to condense the player's direction names to single letters, thus:

```
After reading a command:
  let N be "[the player's command]";
  replace the text "north" in N with "n";
  replace the text "east" in N with "e";
  replace the text "south" in N with "s";
  replace the text "west" in N with "w";
  change the text of the player's command to N.
```

For more on the use of text, see the Advanced Text chapter.

Section 2 - Scenario

The Empty Field is north-northwest of the Deserted Road.

A crop-dusting plane is a backdrop. It is not scenery. It is in the Deserted Road and Empty Field. The initial appearance of the crop-dusting plane is "[one of]In the distance[or]Approaching faster and faster[or]Flying ominously low and directly towards you[or]Immediately overhead[or]Circling around for another approach[cycling] is a standard crop-dusting plane."

After looking:

say "From here you can run to [the list of adjacent rooms]."

Rule for printing the name of a room (called the target) which is not the location while looking:

let chosen direction be the best route from the location to the target;

say "[chosen direction]".

Test me with "sse / north-northwest".

In practice, this is going to be overkill for almost all games: most players already find eight compass directions plus up and down to be enough (or more than enough) to keep track of. But the option exists, in case there is a compelling reason to use it.

(Note also that we are allowed to use multi-word direction names, so we could have called the directions "north by northwest", "north by northeast", and so on. This example deliberately takes the hard way in order to show how to resolve the nine-character problem.)

421

★ Example Igpay Atinlay

WI

A pig Latin filter for the player's commands.

For the sake of argument, suppose we want to parrot back all the player's commands in pig Latin:

"Igpay Atinlay"

Armfoy is a room.

After reading a command:

let N be "[the player's command]";

replace the regular expression "\b(<aeiou>+)(\w*)" in N with "\1\2ay";

replace the regular expression "\b(<bcdfghjklmnpqrstvwxyz>+)(\w*)" in N with "\2\1ay";

say "[N][paragraph break]";

reject the player's command.

Test me with "nix on the stupid".

422

★★ Example Mr. Burns' Repast

WI

Letting the player guess types for an unidentifiable fish.

Suppose we have an unhappily mutated fish that the player can refer to by any of a number of species names, or any word followed by -fish. We want to reject these commands, but preserve a memory of what the player last tried to call the thing:

"Mr. Burns' Repast"

Wharf is a room.

There is an unknown fish in the Wharf. The unknown fish has some a text called the supposed name. The description of the unknown fish is "The victim of heavy mutagens, this thing is not really recognizable as any species you know."

Fish variety is a kind of value. The fish varieties are salmon, albacore, mackerel.

Rule for printing the name of the unknown fish:

if the supposed name of the unknown fish is "", say the printed name of the unknown fish;
otherwise say the supposed name of the unknown fish.

After reading a command:

if the unknown fish is visible and player's command matches the regular expression "\b\w+fish":

let N be "[the player's command]";
replace the regular expression ".*(?=\b\w+fish)" in N with "";
now N is "[N](?)";
now the supposed name of the unknown fish is N;
respond with doubt;
reject the player's command;

otherwise if the unknown fish is visible and the player's command includes "[fish variety]":

now supposed name of the fish is "[fish variety understood](?)";
respond with doubt;
reject the player's command.

To respond with doubt:

say "You're not *sure* you're seeing any such thing."

Test me with "get swordfish / look / touch monkfish / look / listen to tunafish / x fish / x salmon / look".

317



Example Terracottissima Maxima

Flowerpots with textual names that might change during play.

WI

Inform can also understand text properties:

"Terracottissima Maxima"

A flowerpot is a kind of thing. A flowerpot has a text called pattern. Understand the pattern property as describing a flowerpot. The printed name of a flowerpot is usually "[pattern] flowerpot". The printed plural name of a flowerpot is usually "[pattern] flowerpots".

The Herb Garden is a room. In the Herb Garden is a flowerpot with pattern "blue willow". In the Herb Garden is a flowerpot with pattern "striped". In the Herb Garden is a flowerpot with pattern "striped".

Test me with "x blue willow / get striped / look".

This may not seem very much different from having the pattern be a kind of value -- except that texts can, of course, hold almost anything. Further exploration of these possibilities may be found in the chapter on Advanced Text.

381



Example Xot

WI

Storing an invalid command to be repeated as text later in the game.

In Hitchhiker's Guide to the Galaxy, any erroneous command the player types can return to haunt him later in the game. We could do the same, if we liked, by storing the player's command whenever we print a parser error.

"Xot"

Humiliation Chamber is a room. "A grim, grey-walled room. Cameras watch you from every angle; convex mirrors reflect your actions; and up near the ceiling, where you can't disable it, is a loudspeaker."

The last error is a text that varies. The last error is "xot".

Before printing a parser error:
now the last error is the player's command.

Every turn when a random chance of 1 in 2 succeeds:
say "Over the loudspeaker comes some distorted nonsense. If you listen carefully, it sounds as though some fool is saying '[last error], [last error], [last error]!'"

Test me with "wiggle / z / z / z / z / z / z".

Chapter 3: Place

§3.1. *Room Descriptions*; §3.2. *Map*; §3.3. *Position Within Rooms*;
§3.4. *Continuous Spaces and The Outdoors*; §3.5. *Doors, Staircases, and Bridges*;
§3.6. *Windows*; §3.7. *Lighting*; §3.8. *Sounds*;
§3.9. *Passers-By, Weather and Astronomical Events*

-  Contents of *The Inform Recipe Book*
-  Chapter 2: Adaptive Prose
-  Chapter 4: Time and Plot
-  Indexes of the examples

§3.1. Room Descriptions

The printing of a room description is a more delicate business than it might initially seem to be: Inform has to consider all the objects that the player might have brought into the room or dropped there, and all the objects on visible supporters, and decide how to group and list them.

All of this behavior is handled by the looking command, so we find the relevant rules in the carry out looking rulebook. To go through the elements step by step:

Looking begins by printing the name and description of the room we're in. We can introduce variations into room names and descriptions by changing their printed name and description properties, as in

now the printed name of the Church is "Lightning-Struck Ruin";
now the description of the Church is "The beams overhead have been burnt away and the pews are charred. Only the stone walls remain.";

If we need more drastic effects, we can turn off or change either of these features by altering the rules in the carry out looking rulebook. For instance, to remove the name of the location entirely from room descriptions, we would write

The room description heading rule is not listed in the carry out looking rules.

(A word of warning: there is one other context in which the story prints a room name — when restoring a save or undoing a move. To omit the room title here too, add

Rule for printing the name of a room: do nothing.)

Ant-Sensitive Sunglasses demonstrates how to use activities to make more flexible room description text.

Next, the story determines what items are visible to the player and need to be described. These never include the player himself, or scenery, but other things in the environment will

be made "marked for listing". This is also the stage at which Inform chooses the order in which items will be listed.

We are allowed to meddle by changing the priorities of objects, in case we want some things to be described to the player first or last in the room description; **Priority Lab** goes into detail about how. We can also force things to be left out entirely: **Low Light** handles the case of an object that can only be seen when an extra lamp is switched on, even though the room is not otherwise considered dark. **Copper River** implements the idea of "interesting" and "dull" objects: the story determines which items are currently important to the puzzles or narrative and mentions those in the room description, while suppressing everything else.

Then Inform carries out the writing a paragraph about... activity with anything that provides one; anything it prints the name of, it tags "mentioned". Thus

Rule for writing a paragraph about Mr Wickham:
say "Mr Wickham looks speculatively at [list of women in the location]."

will count Wickham and everyone he looks at as all having been mentioned, and will not refer to them again through the rest of the room description. More complicated uses of writing a paragraph abound. A developed system for handling supporters that don't list contents appears in **The Eye of the Idol**.

Inform then prints the initial appearances of objects that are marked for listing but not already mentioned; and then it performs the listing nondescript items activity, collating the remaining objects into a paragraph like

You can see a dog, a hen, ...

We can pre-empt items from appearing in this paragraph or change their listing by intervening with a Before listing nondescript items... rule, as in

Before listing nondescript items when the player needs the watch:
if the watch is marked for listing:
say "The watch catches your eye.";
now the watch is not marked for listing.

If we wanted the watch always to be listed this way, it would be better to give it an initial appearance, but for conditional cases, the listing nondescript items activity is a good place to intervene. For instance, **Rip** uses this activity to incorporate changeable or portable items into the main description text for a room when (and only when) that is appropriate.

The listing nondescript items activity also allows us to replace the "You can see..." tag with something else more fitting, if for instance we are in a dimly lit room.

When the story compiles the list of nondescript items, it adds tags such as "(open)" or "(empty)" or "(on which is a fish tank)" to the names of containers and supporters. We can suppress or change the "(empty)" tag with the printing room description details of activity, as in

Rule for printing room description details: stop.

And we can suppress the "(open)" and "(on which is...)" sorts of tags with the "omit the contents in listing" phrase, as in

Rule for printing the name of the bottle while not inserting or removing:
if the bottle contains sand, say "bottle of sand";
otherwise say "empty bottle";
omit contents in listing.

Finally, the looking command lists visible non-scenery items that sit on scenery supporters, as in

On the table is a folded newspaper.

These paragraphs can be manipulated with the printing the locale description activity and the printing a locale paragraph about activity.

Another common thing we may want to do is change the description of a room depending on whether we've been there before (as in **Slightly Wrong**) or on how often we've visited (as in **Infiltration**). **Night Sky**, meanwhile, changes the description of a room when we've examined another object, so that the player's awareness of his environment is affected by other things the character knows.

★ See **Looking** for ways to change the default length of room descriptions

- ⬆ Start of Chapter 3: Place
 - ⬅ Back to Chapter 2: Adaptive Prose: §2.3. Using the Player's Input
 - ➡ Onward to §3.2. Map
 - ⬇ Example 147: ★ **Night Sky** A room which changes its description depending on whether an object has been examined.
 - ⬇ Example 152: ★ **Infiltration** A room whose description changes depending on the number of times the player has visited.
 - ⬇ Example 334: ★ **Ant-Sensitive Sunglasses** What are activities good for? Controlling output when we want the same action to be able to produce very flexible text depending on the state of the world -- in this case, making highly variable room description and object description text.
 - ⬇ Example 355: ★ **Rip Van Winkle** A simple way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description.
 - ⬇ Example 358: ★ **Priority Lab** A debugging rule useful for checking the priorities of objects about to be listed.
 - ⬇ Example 359: ★ **Low Light** An object that is only visible and manipulable when a bright light fixture is on.
 - ⬇ Example 4: ★★ **Slightly Wrong** A room whose description changes slightly after our first visit there.
 - ⬇ Example 357: ★★★ **The Eye of the Idol** A systematic way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description, and to control whether supporters list their contents or not.
 - ⬇ Example 362: ★★★ **Copper River** Manipulating room descriptions so that only interesting items are mentioned, while objects that are present but not currently useful to the player are ignored.
-

§3.2. Map

A work of IF contains many spectacles and activities, and these must not all present themselves at once, or the player will be overwhelmed. One way to spread them out is in time, by having them available only as a plot develops, but another is to spread them out literally in space. The player has to walk between the Library and the Swimming Pool, and thus bookish and athletic tasks are not both presenting themselves at once. There have been valiant "one-room" IFs, and it forms a respectable sub-genre of the art, but most works of any size need a map.

Inform, following IF conventions, divides the world up into locations called "rooms", connected together by so-called "map connections" along compass bearings. Thus:

[The Library is east of the Swimming Pool.](#)

The example **Port Royal 1** develops a medium-sized map from such sentences. This develops in **Port Royal 2** to include connections which bend around, allowing the rooms not to lie on an imaginary square grid.

Because it is useful to group rooms together under names describing whole areas, Inform also allows rooms to be placed in "regions". Thus:

The Campus Area is a region. The Library and the Swimming Pool are in the Campus Area.

Port Royal 3 demonstrates this further. **A&E** shows how regions can be used to write simple rules which regulate access to and from whole areas of the map.

Many old-school IF puzzles involve journeys through the map which are confused, randomised or otherwise frustrated: see **Bee Chambers** for a typical maze, **Zork II** for a randomised connection, **Prisoner's Dilemma** for a change in the map occurring during play. A completely random map takes us away from traditional IF and more towards a different sort of old-school game, the computerised role-playing game with its endless quests through dungeons with randomly generated treasures and monsters. This style of map - building itself one step at a time, as the player explores - can sometimes be useful to provide an illusion of infinite expanse: see **All Roads Lead To Mars**.

While the standard compass directions are conventional in IF, there are times when we may want to replace them without other forms of directional relationship. **Indirection** renames the compass directions to correspond to primary colors, as in Mayan thinking. **The World of Charles S. Roberts** substitutes new ones, instead, introducing a hex-grid map in place of the usual one.

★ See **Going, Pushing Things in Directions** for ways to add more relative directions, such as context-sensitive understanding of *OUT* and *IN*

★ See **Room Descriptions** for ways to modify the room description printed

★ See **Ships, Trains and Elevators** for rooms which move around in the map and for directions aboard a ship

- ⬆ Start of Chapter 3: Place
 - ⬅ Back to §3.1. Room Descriptions
 - ➡ Onward to §3.3. Position Within Rooms
 - ⬇ Example 5: **★ Port Royal 1** A partial implementation of Port Royal, Jamaica, set before the earthquake of 1692 demolished large portions of the city.
 - ⬇ Example 8: **★ Port Royal 2** Another part of Port Royal, with less typical map connections.
 - ⬇ Example 10: **★ Port Royal 3** Division of Port Royal into regions.
 - ⬇ Example 78: **★ All Roads Lead to Mars** Layout where the player is allowed to wander any direction he likes, and the map will arrange itself in order so that he finds the correct "next" location.
 - ⬇ Example 125: **★ Bee Chambers** A maze with directions between rooms randomized at the start of play.
 - ⬇ Example 134: **★ Zork II** A "Carousel Room", as in Zork II, where moving in any direction from the room leads (at random) to one of the eight rooms nearby.
 - ⬇ Example 286: **★ Indirection** Renaming the directions of the compass so that "white" corresponds to north, "red" to east, "yellow" to south, and "black" to west.
 - ⬇ Example 40: **★★ Prisoner's Dilemma** A button that causes a previously non-existent exit to come into being.
 - ⬇ Example 41: **★★ The World of Charles S. Roberts** Replacing the ordinary compass bearings with a set of six directions to impose a hexagonal rather than square grid on the landscape.
 - ⬇ Example 101: **★★ A&E** Using regions to block access to an entire area when the player does not carry a pass, regardless of which entrance he uses.
-

§3.3. Position Within Rooms

Inform's division of geography into "rooms" is a good compromise for most purposes. The rooms are cut off from each other by (imaginary or actual) walls, while all of the interior of a given room is regarded as the same place.

Suppose we want things to happen differently in different corners of the same room? Inform can already do this a little, in that the player can be inside an enterable container or on an enterable supporter. For instance:

Instead of opening a door when the player is on the bed, say "You can't reach the handle from the bed."

If we need to have divided-up areas of the floor itself, the standard approach is to define a small number of named positions. We then need to remember at which of these locations the player (or something else) currently stands.

Further Reasons Why All Poets Are Liars allows the player to be in different parts of a room by standing on a box which can be in different places: thus only the box needs an internal position, not the player, simplifying matters neatly.

Another interesting case is when one room is entirely inside another (such as a hut in a field, or a booth in a large convention hall), so that the exterior of the room should be visible from

another location. **Starry Void** gives a simple demonstration of a magician's booth that can be examined from the outside, opened and closed, and entered to reach a new location.

★ See **Continuous Spaces and The Outdoors** for making the space between rooms continuous

★ See **Combat and Death** for the use of position in a room in determining combat maneuvers

★ See **Entering and Exiting, Sitting and Standing** for automatically getting up from chairs before going places

★ See **The Human Body** for letting the player take different postures on furniture or on the floor

★ See **Furniture** for cages, beds, and other kinds of enterable supporters and containers



Start of Chapter 3: Place



Back to §3.2. Map



Onward to §3.4. Continuous Spaces and The Outdoors



Example 199: ★ **Further Reasons Why All Poets Are Liars** The young William Wordsworth, pushing a box about in his room, must struggle to achieve a Romantic point of view.



Example 7: ★★★ **Starry Void** Creating a booth that can be seen from the outside, opened and closed, and entered as a separate room.

§3.4. Continuous Spaces and The Outdoors

Suppose we want to blur the boundaries between rooms, in an environment where there are no walls: out of doors, for instance?

The simplest cases involve making something exceptional visible in more than one place. **Carnivale** features an exceptionally large landmark seen by day; **Eddystone** an exceptionally bright one by night. **Waterworld** allows a very distant object (the Sun) to be seen throughout many rooms, but never approached. **View of Green Hills** gives the player an explicit command for looking through into an adjacent room.

Three systematic examples then present outdoor landscapes with increasing sophistication. **Tiny Garden** gives the multiple rooms of an extended lawn descriptions which automatically adapt to say which directions lead into further lawn area. **Rock Garden** provides a relation, "connected with", between rooms, allowing items in one to be seen from the other: an attempt to interact with a visible item in a different area of the garden triggers an implicit going action first. **Stately Gardens** provides a much larger outdoor area, where larger landmarks are visible from further away, and room descriptions are highly adaptive.

In an outdoor environment, the distinction between a one-move journey and a multiple-move journey is also blurred. **Hotel Stechelberg** shows a signpost which treats these equally.

★ See **Position Within Rooms** for making the space within a room continuous

★ See **Windows** for another way to see between locations

★ See **Doors, Staircases, and Bridges** for still a third way to be told at least what lies adjacent

★ See **Passers-By, Weather and Astronomical Events** for more on describing the sky

- ↑ Start of Chapter 3: Place
 - ← Back to §3.3. Position Within Rooms
 - Onward to §3.5. Doors, Staircases, and Bridges
 - ↓ Example 216: ★ **Waterworld** A backdrop which the player can examine, but cannot interact with in any other way.
 - ↓ Example 62: ★★ **Tiny Garden** A lawn made up of several rooms, with part of the description written automatically.
 - ↓ Example 79: ★★★ **Hotel Stechelberg** Signposts such as those provided on hiking paths in the Swiss Alps, which show the correct direction and hiking time to all other locations.
 - ↓ Example 212: ★★★ **Carnivale** An alternative to backdrops when we want something to be visible from a distance but only touchable from one room.
 - ↓ Example 213: ★★★ **Eddystone** Creating new commands involving the standard compass directions.
 - ↓ Example 366: ★★★ **Rock Garden** A simple open landscape where the player can see between rooms and will automatically move to touch things in distant rooms.
 - ↓ Example 80: ★★★ **A View of Green Hills** A LOOK [direction] command which allows the player to see descriptions of the nearby landscape.
 - ↓ Example 367: ★★★ **Stately Gardens** An open landscape where the player can see landmarks in nearby areas, with somewhat more complex room descriptions than the previous example, and in which we also account for size differences between things seen at a distance.
-

§3.5. Doors, Staircases, and Bridges

Inform's "door" kind provides for a tangible thing which comes between one room and another. A door can be open or closed, and openable or not: it can be locked or unlocked, and lockable or not. Here we create a conventional door, a natural gap in the rocks, and a (fixed in place) wooden ladder:

The fire door is an open door. The fire door is east of the Projection Booth and west of the Fire Escape.

The narrow crevice is an open unopenable door. The crevice is east of the Col du Prafleuri and west of Rocky Knoll Above Arolla.

The wooden ladder is an open unopenable door. The ladder is above the Stableyard and below the Hay Loft.

Most doors are visible from both sides: they are single objects but present in two rooms at once, which raises a number of complications. Inform normally uses the same description looking from each way, which is not very interesting: **When?** and **Whence?** demonstrate neat ways to describe the two sides differently, and **Whither?** adds the option for the player to refer to doors as "the west door" and "the east door" automatically.

Neighbourhood Watch goes further by making a door behave differently on each side: from the "outside" you need a key, but "inside" it opens on a latch. Finally, **Garibaldi 1** shows how to access information about the two sides of a door.

Higher Calling demonstrates doors which automatically open as needed: though using the Inform extension Locksmith by Emily Short is probably easier and better. **Elsie**, conversely, demonstrates a door that closes one turn after the player has opened it.

Certain complications apply when characters other than the player have to see and interact with doors that exist in other rooms. **Wainwright Acts** demonstrates the syntax needed to handle this technically quirky situation.

Something Narsty and **Hayseed** provide a "staircase" kind useful for vertically arranged, always-open doors like staircases and (fixed in place) ladders.

One Short Plank implements a precarious plank bridge across a chasm as an open unopenable door.

★ See **Windows** for *climbing through a window from one room to another*

★ See **Ropes** for *portable connections between rooms, much of the development of which could be adapted to handle portable ladders. "Doors" are never allowed to move*

★ See **Magic (Breaking the Laws of Physics)** for *a hat that lets the player walk through closed doors*

★ See **Modifying Existing Commands** for *ways to allow the player to unlock with a key he isn't currently holding*

- ⬆ Start of Chapter 3: Place
 - ⬅ Back to §3.4. Continuous Spaces and The Outdoors
 - ➡ Onward to §3.6. Windows
 - ⬇ Example 45: ★ **Something Narsty** A staircase always open and never openable.
 - ⬇ Example 63: ★ **When?** A door whose description says "...leads east" in one place and "...leads west" in the other.
 - ⬇ Example 89: ★ **Hayseed** A refinement of our staircase kind which can be climbed.
 - ⬇ Example 128: ★ **Higher Calling** All doors in the game automatically attempt to open if the player approaches them when they are closed.
 - ⬇ Example 236: ★ **Wainwright Acts** A technical note about checking the location of door objects when characters other than the player are interacting with them.
 - ⬇ Example 321: ★ **Whither?** A door whose description says where it leads; and which automatically understands references such as "the west door" and "the east door" depending on which direction it leads from the location.
 - ⬇ Example 24: ★★ **Neighborhood Watch** A locked door that can be locked or unlocked without a key from one side, but not from the other.
 - ⬇ Example 106: ★★★ **One Short Plank** A plank bridge which breaks if the player is carrying something when he goes across it. Pushing anything over the bridge is forbidden outright.
 - ⬇ Example 151: ★★★ **Elsie** A door that closes automatically one turn after the player opens it.
 - ⬇ Example 22: ★★★ **Garibaldi 1** Providing a security readout device by which the player can check on the status of all doors in the game.
 - ⬇ Example 64: ★★★ **Whence?** A kind of door that always automatically describes the direction it opens and what lies on the far side (if that other room has been visited).
-

§3.6. Windows

Calvin Coolidge once described windows as "rectangles of glass." For us, they have two purposes: first, they offer a view of landscape beyond. In the simplest case the view is of an area which will not be interacted with in play, and therefore does not need to adapt to whatever may have changed there:

The window is scenery in the Turret. "Through the window you see miles and miles of unbroken forest, turning from green to flame in the hard early autumn."

More interesting is to adapt the view a little to provide a changing picture: a forest may not change much, but a street scene will. **Port Royal 4** allows us to glimpse random passers-by.

The trickiest kind of window allows the player to see another room which can also be encountered in play, and to interact with what is there. **Dinner is Served** presents a shop window, allowing people to see inside from the street, and even to reach through.

Vitrine handles the complication of a window misting up to become opaque, and thus temporarily hiding its view.

Second, windows provide openings in walls and can act as conduits. **Escape** shows how a "door" in the Inform sense can become a window. **A Haughty Spirit** provides a general kind

of window for jumping down out of: ideal for escapers from Colditz-like castles.

★ See **Doors, Staircases, and Bridges** for a door which can be partially seen through

- ↑ Start of Chapter 3: Place
 - ← Back to §3.5. Doors, Staircases, and Bridges
 - Onward to §3.7. Lighting
 - ↓ Example 116: ★ **Vitrine** An electrochromic window that becomes transparent or opaque depending on whether it is currently turned on.
 - ↓ Example 21: ★★ **Escape** Window that can be climbed through or looked through.
 - ↓ Example 217: ★★★ **Dinner is Served** A window between two locations. When the window is open, the player can reach through into the other location; when it isn't, access is barred.
 - ↓ Example 272: ★★ **Port Royal 4** A cell window through which the player can see people who were in Port Royal in the current year of game-time.
 - ↓ Example 181: ★★★ **A Haughty Spirit** Windows overlooking lower spaces which will prevent the player from climbing through if the lower space is too far below.
-

§3.7. Lighting

At any place (room, or inside a container) light is either fully present or fully absent. Inform does not usually try to track intermediate states of lighting, but see **The Undertomb 2** for a single lantern with varying light levels and **Zorn of Zorna** for multiple candles that can be lit for cumulative changes to the light level.

Light can be added to, but not taken away: rooms and things can act as sources of light, by having the "lighted" and "lit" properties respectively, but they cannot be sinks which drain light away. The reason darkness is not a constant hazard in Inform-written games is that rooms always have the "lighted" property unless declared "dark". (We assume daylight or some always-on electric lighting.) A "dark" room may well still be illuminated if a light source happens to be present:

The Deep Crypt is a dark room. The candle lantern is a lit thing in the Deep Crypt.

Hymenaeus allows us to explicitly refer to torches as "lit" or "unlit", or (as synonyms) "flaming" or "extinguished".

For light produced electrically we might want a wall switch, as in **Down Below**, or a portable lamp, as in **The Dark Ages Revisited**.

The fierce, locally confined light thrown out by a carried lamp has a quality quite unlike weak but ambient daylight, and **Reflections** exploits this to make a lantern feel more realistic.

When the player experiences darkness in a location, Inform is usually very guarded in what it reveals. ("It is pitch dark, and you can't see a thing.") **Hohmann Transfer** gives darkness a quite different look, and **Four Stars** heightens the other senses so that a player in darkness

can still detect her surroundings. The first of the two examples in **Peeled** allows exploration of a dark place by touch.

It is sometimes useful to check whether a room that is not the current location happens to contain a light source or be naturally lighted. This poses a few challenges. **Unblinking** demonstrates one way of doing this, so long as there are no backdrop light sources.

Cloak of Darkness is a short and sweet game based on a light puzzle.

★ See **Room Descriptions** for an item that can only be seen in bright light, when an extra lamp is switched on

★ See **Looking Under and Hiding** for a looking under action which is helped by the fiercer brightness of a light source

★ See **Going, Pushing Things in Directions** for making it hazardous to walk around in the dark

★ See **Electricity and Magnetism** for batteries to power a torch or flashlight

★ See **Fire** for a non-electrical way to produce light

↑ Start of Chapter 3: Place

← Back to §3.6. Windows

→ Onward to §3.8. Sounds

↓ Example 194: ★ **The Dark Ages Revisited** An electric light kind of device which becomes lit when switched on and dark when switched off.

↓ Example 312: ★ **Hymenaeus** Understanding "flaming torch" and "extinguished torch" to refer to torches when lit and unlit.

↓ Example 352: ★ **Reflections** Emphasizing the reflective quality of shiny objects whenever they are described in the presence of the torch.

↓ Example 363: ★ **Peeled** Two different approaches to adjusting what the player can interact with, compared.

↓ Example 26: ★★ **Down Below** A light switch which makes the room it is in dark or light.

↓ Example 51: ★★★ **The Undertomb 2** Flickering lantern-light effects added to the Undertomb.

↓ Example 348: ★★★ **Hohmann Transfer** Changing the way dark rooms are described to avoid the standard Inform phrasing.

↓ Example 81: ★★★★ **Unblinking** Finding a best route through light-filled rooms only, leaving aside any that might be dark.

↓ Example 347: ★★★★★ **Zorn of Zorna** Light levels vary depending on the number of candles the player has lit, and this determines whether or not he is able to examine detailed objects successfully.

↓ Example 349: ★★★★★ **Four Stars 1** An elaboration of the idea that when light is absent, the player should be given a description of what he can smell and hear, instead.

↓ Example 291: ★★★★★ **Cloak of Darkness** Implementation of "Cloak of Darkness", a simple example game that for years has been used to demonstrate the features of IF languages.

§3.8. Sounds

It is too easily assumed that room descriptions are what the player sees, but as **The Undertomb** demonstrates, they might just as easily include ambient sounds.

So Inform's "listening to" action is the audio equivalent of "examining", rather than "looking". Despite this the player can type LISTEN, which Inform understands as listening to the everything in the location at once. A simple but effective way to handle this is shown in **The Art of Noise**.

Four Stars 2 adjusts the idea of "visibility" to make it behave differently for listening purposes: this introduces a formal idea of "audibility".

★ See **Lighting** for heightened hearing in darkness, and the rest of "Four Stars"

↑ Start of Chapter 3: Place

← Back to §3.7. Lighting

→ Onward to §3.9. Passers-By, Weather and Astronomical Events

↓ Example 50: ★ **The Undertomb 1** A small map of dead ends, in which the sound of an underground river has different strengths in different caves.

↓ Example 364: ★ **Four Stars 2** Using "deciding the scope" to change the content of lists such as "the list of audible things which can be touched by the player".

↓ Example 95: ★★ **The Art of Noise** Things are all assigned their own noise (or silence). Listening to the room in general reports on all the things that are currently audible.

§3.9. Passers-By, Weather and Astronomical Events

Out of doors, nature is seldom still. Clouds scull by at random, as in **Weathering**, and provide some variety in what would otherwise be lifelessly static room descriptions. In much the same way, passers-by and other diversions make a city street a constant bustle: see **Uptown Girls** for this human breeze. A more nagging sense of atmosphere can be experienced in **Full Moon**.

Orange Cones offers traffic that is present on every road in the story unless a room is marked off with orange cones -- and this is allowed to change during play.

Night and Day and **Totality** each schedule celestial events to provide a changing display in the sky above, and this time running like clockwork rather than at random.

★ See **Scene Changes** for meteors and a moon-rise

- ↑ Start of Chapter 3: Place
 - ← Back to §3.8. Sounds
 - Onward to Chapter 4: Time and Plot: §4.1. The Passage Of Time
 - ↓ Example 131: ★ **Weathering** The automatic weather station atop Mt. Pisgah shows randomly fluctuating temperature, pressure and cloud cover.
 - ↓ Example 157: ★ **Full Moon** Random atmospheric events which last the duration of a scene.
 - ↓ Example 164: ★ **Night and Day** Cycling through a sequence of scenes to represent day and night following one another during a game.
 - ↓ Example 144: ★★ **Totality** To schedule an eclipse of the sun, which involves a number of related events.
 - ↓ Example 120: ★★★ **Orange Cones** Creating a traffic backdrop that appears in all road rooms except the one in which the player has laid down orange cones.
 - ↓ Example 132: ★★★ **Uptown Girls** A stream of random pedestrians who go by the player.
-

Examples from Chapter 3: Place

- ↑ Start of this chapter
- Chapter 4: Time and Plot
- ↘ Indexes of the examples

147

★ Example **Night Sky**

WI

A room which changes its description depending on whether an object has been examined.

Sometimes a nice effect is to change the way things are described depending on the information the player has gained in the course of play. We could for instance write this:

"Night Sky"

The Planetarium is a room. "[if we have examined the sinister message]A dark room where it seems something is about to jump out at you![otherwise]A tranquil dark room with a ceilingful of stars.[end if]"

The sinister message is a thing in the Planetarium. "A message is taped to the wall." The description is "BEWARE."

Test me with "look / x message / look".

On the other hand, beware that this would not work as desired:

"Night Sky"

The Planetarium is a room. "[if we have listened to the sinister message]A dark room where it seems something is about to jump out at you![otherwise]A tranquil

dark room with a ceilingful of stars.[end if]"

The sinister message is a thing in the Planetarium. "A message plays very softly, so that you would have to listen to hear it." Instead of doing anything other than listening to the message: say "It's only a sound, after all.". Instead of listening to the sinister message: say "A voice whispers, 'BEWARE'."

Test me with "listen to message / look".

The reason is that our Instead rule has pre-empted normal listening, so Inform considers that we have never successfully heard the message. The moral here is that "if we have..." is useful for tracking events that otherwise proceeded completely normally (picking up ordinary objects, examining things); if we have used instead to make some change, we will have to use a different approach to record that the event did occur as scheduled.

152

★ Example Infiltration

WI

A room whose description changes depending on the number of times the player has visited.

Suppose we have a location that makes the player uncomfortable, and we want its description to change slightly each time he goes there, to reflect his increasing unease. We also want the door to that room to show whether he is going there for the first time, or whether this is a repeat visit.

We start with an ordinary room:

"Infiltration"

The Wasteland is a room. "In its more distant reaches, the Wasteland has a kind of austere beauty, but here beside the Secure Zone it is the worst of all possible worlds. Barrels of toxins are stacked the regulation hundred and fifty feet out; more traditional garbage has simply been flung over the wall, and this category includes one or two corpses roughly and inadequately disguised by black plastic bags. The wall itself has become a canvas for outcasts and exiles, and is covered with obscene paintings, lewd remarks about the inhabitants of the Secure Zone, and a few maudlin epitaphs."

Now the door, which will change from saying "leads inside..." to "leads back inside..." when this becomes appropriate:

The portal is a door. It is inside from the Wasteland and outside from the Secure Zone. "[if the player is in the Wasteland]To the west, a[otherwise]A[end if] portal in the cinder-block and barbed wire wall leads[if the player is in the Wasteland and the Zone is visited] back[end if] [if the player is in the Wasteland]inside[otherwise]outside[end if]."

Here we haven't used any conditions that we didn't know about in previous sections: the portal line only reflects whether the Zone has been visited never or visited once. But the Secure Zone itself makes use of the number of times visited:

The Secure Zone has the description "[if the player is in the Zone for the second time]Re-entering the Zone has not made you any more comfortable inside. [end if]Despite your carefully-chosen outfit and the walk you have been practicing, you are sure those inside can tell you don't belong. Not that very many people are visible here[if the player is in the Zone for more than the second time] -- the place seems more desolate than ever[end if]."

Instead of going west in the Wasteland, try going inside. Instead of going east in the Secure Zone, try going outside.

And finally, to be sure that the player does see our fancy changing descriptions:

Use full-length room descriptions.

Test me with "look / open portal / w / look / e / look / w / e / w".

Notice that the description of the Secure Zone changes from visit to visit, but that looking repeatedly during a single visit changes nothing.

334

★ Example Ant-Sensitive Sunglasses

WI

What are activities good for? Controlling output when we want the same action to be able to produce very flexible text depending on the state of the world -- in this case, making highly variable room description and object description text.

Suppose we want to create an object -- or maybe even a series of objects -- that warp the player's perception of every room description and object around him.

We've already seen some ways to create variations in text. For instance, we could make a room description with if substitutions in it, like so:

The Kitchen is a room. "[if the player is wearing the sunglasses]Are ants coming out of the sink? No, probably no.[otherwise]A small kitchen tucked into a corner of the vacation house. There is storage space for five or six cups, a sink, a two-ring stove; nothing else to speak of.[end if]"

That works fine if we have one or two variations we want to add; it's not so good if we're going to have several items that work like the sunglasses, or if we want the sunglasses to override the description of every room in the house.

A slightly more flexible method is to use a substitution that calls out to a say phrase, like this:

The Kitchen is a room. "[kitchen-description]"

To say kitchen-description:

if the player is wearing the sunglasses:

say "Are ants coming out of the sink? No, probably no.";

otherwise:

say "A small kitchen tucked into a corner of the vacation house. There is

storage space for five or six cups, a sink, a two-ring stove; nothing else to speak of."

But again this doesn't handle the case of overriding multiple rooms at once very well.

When we reach a point where we need a given piece of text to be very flexible depending on the world model, it's time to use an activity.

Activities offer several advantages. One, we can create an activity like this:

Printing the room-description of something is an activity.

and then write a rule that applies to multiple rooms at once, like:

Rule for printing the room-description of a room when the player wears the sunglasses:
say "The walls look like they're covered with ants. Just a coincidence, I'm sure."

Inform's usual rule-ranking also means that more-specific rules will override less-specific ones, so we could add

Rule for printing the room-description of the Kitchen when the player wears the sunglasses:
say "Are ants coming out of the sink? No, probably not."

and have that rule override the behavior of the activity just in the kitchen. Meanwhile, our base room descriptions remain straightforward and uncluttered by if-statements.

Several other examples will show how to hook activities into existing actions: Crusoe goes into detail about how to make the descriptions of things more variable, and Aftershock demonstrates activities for describing the behavior of switchable devices.

Here, we preview all of those methods, just to get a sense of how they work and why they might be useful in controlling a game. Subsequent chapters go into more detail about the syntax of creating activities and the list of activities that are already defined by Inform.

"Ant-Sensitive Sunglasses"

Part 1 - Procedure

To add a new activity to an existing Inform rule, we need to do three things:

- 1) Define our new activity.
- 2) Give a basic rule that says what is supposed to happen when that activity occurs, as in "Rule for..."

3) Replace the existing rule in Inform's rulebooks with a new one that calls on our activity.

Here we do this with examining:

Section 1 - Item Description

Printing the description of something is an activity.

Now, by default, we want to print the description property; we just want the option to write some extra rules overriding that property. So we tell Inform that our most basic rule for printing the description of something is just to give that description text:

```
Rule for printing the description of something (called item):  
  if the description of the item is not "":  
    say "[description of item][paragraph break]";  
  otherwise:  
    say "You see nothing special about [the item].".
```

Next, we need the standard examining rule to look at our printing-the-description activity:

The activity-based examining rule is listed instead of the standard examining rule in the carry out examining rules.

```
This is the activity-based examining rule:  
  carry out the printing the description activity with the noun;  
  rule succeeds.
```

Now we do the same thing to room descriptions.

Section 2 - Room Description

Printing the room-description of something is an activity.

```
Rule for printing the room-description of a room (called item):  
  if the description of the item is not "":  
    say "[description of item][paragraph break]";  
  otherwise:  
    do nothing instead.
```

The activity-based room description body text rule is listed instead of the room description body text rule in the carry out looking rules.

Our replacement rule this time around is a little bit trickier just because the rule that we're replacing is a complicated one: describing a room already checks to see whether there's light to see by, whether the player has turned off room descriptions when he enters a room for the second time, and whether the player character is (say) inside a closed box he can't see out of.

But all of those details are re-copied from the standard rules, and the important thing is that, at the end, we again carry out our activity.

This is the activity-based room description body text rule:

- if the visibility level count is 0:
 - if set to abbreviated room descriptions, continue the action;
 - if set to sometimes abbreviated room descriptions and abbreviated form allowed is true and darkness witnessed is true, continue the action;
 - begin the printing the description of a dark room activity;
 - if handling the printing the description of a dark room activity, say "It is pitch dark, and you can't see a thing.";
 - end the printing the description of a dark room activity;
- otherwise if the visibility ceiling is the location:
 - if set to abbreviated room descriptions, continue the action;
 - if set to sometimes abbreviated room descriptions and abbreviated form allowed is true and the location is visited, continue the action;
 - carry out the printing the room-description activity with the location.

Section 3 - Device Description

Showing action of something is an activity.

Rule for showing action of something (called item):

- if the item is switched on, say "[The item] is switched on.";
- otherwise say "[The item] is switched off."

The activity-based described devices rule is listed instead of the examine devices rule in the carry out examining rules.

This is the activity-based described devices rule:

- if the noun is a device:
 - carry out the showing action activity with the noun;
 - now examine text printed is true.

Report switching on something:

- say "You flip a switch. ";
- carry out the showing action activity with the noun instead.

Part 2 - Scenario

The Kitchen is a room. "A small kitchen tucked into a corner of the vacation house. There is storage space for five or six cups, a sink, a two-ring stove; nothing else to speak of."

The microwave is a device in the Kitchen.

South of the Kitchen is the Living Area. The description of the Living area is "A whitewashed living/dining/reclining area in what used to be a shepherd's stone hut, but now costs vacationers 600 euros a week. It offers no mod cons, only a straight view of the Mediterranean and a wobbly writing table."

Rule for printing the room-description of a room when the player wears the sunglasses:

- say "The walls look like they're covered with ants. Just a coincidence, I'm sure[antsy]."

Rule for printing the room-description of the Kitchen when the player wears the sunglasses:

say "Are ants coming out of the sink? No, probably not[antsy]."

Rule for printing the description of something (called the item) when the player wears the sunglasses:

say "[The item] [are] [one of]ant-colored[or]ant-legged[or]covered in ants[at random][antsy]."

Rule for showing action of the microwave:

say "The microwave hums meaningfully to itself."

Rule for showing action of the microwave when the player wears the sunglasses:

say "The microwave hums as though inhabited by a billion ants[antsy]."

The player carries sunglasses of freakiness and an apple. The apple is edible. The sunglasses are wearable.

ant-paranoia is a number that varies.

To say antsy:

increase ant-paranoia by 1;

Every turn:

if the ant-paranoia is greater than 3:

say "Augh! AUUUGH! GET THEM OFF--";

end the story saying "You have lost your mind."

Test me with "look / turn on microwave / turn off microwave / x apple / x sunglasses / s / wear sunglasses / look / x apple / n / turn on microwave".

355

★ Example Rip Van Winkle

WI

A simple way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description.

There are times when, for greater elegance of prose, we'd like to mention an object in the main body text of a room. For instance:

Here is a lovely, secluded fold in the mountains, far from civilization: as though to prove it, Rip Van Winkle is sleeping under a tree.

As we've already seen, that's no problem if Rip is scenery. He'll stay there motionless.

But what if something in the game allows Rip to wake up? Or what if we want to use the same technique on a portable object that the player should be allowed to take? Clearly in that case it's not appropriate to make the mentioned thing be scenery, and at the same time, we need to keep Inform from adding a superfluous

You can see Rip Van Winkle here.

to the end of our description.

Here is how:

"Rip Van Winkle"

A person can be asleep.

The Catskills is a room. "Here is a lovely, secluded fold in the mountains, far from civilization[if Rip Van Winkle is asleep]: as though to prove it, Rip Van Winkle is sleeping under a tree[end if]."

A tree is scenery in the Catskills.

Rip Van Winkle is a man in the Catskills. Rip Van Winkle is asleep.

Before listing nondescript items of the Catskills:
if Rip Van Winkle is marked for listing:
now Rip Van Winkle is not marked for listing;
if Rip Van Winkle is not asleep,
say "Rip Van Winkle stands here, looking mightily confused."

Instead of waiting:
say "Rip Van Winkle wakes up with a snort.";
now Rip Van Winkle is not asleep.

Test me with "look / z / look".

358

★ Example Priority Lab

WI

A debugging rule useful for checking the priorities of objects about to be listed.

When it comes time to start manipulating the priorities of items, it is useful to be able to check the table for debugging purposes; the problem is that printing the names of the objects can itself affect the way the room description is generated, foiling our debugging efforts.

What follows is a rule to help with debugging safely, and a sample of how priorities work:

"Priority Lab"

Section 1 - Procedure

Before printing the locale description (this is the dump locale table rule):
say "Locale Priority list:";
repeat through Table of Locale Priorities:
let the flag be whether or not the notable-object entry is mentioned;

say "[line break] [notable-object entry]: [locale description priority entry]";
if the flag is false, now the notable-object entry is not mentioned;
say line break.

Now, let's look at some items put in a specific order. Things with low priority numbers list towards the beginning; things with high priority numbers list towards the end. (It helps to think of it as though we were making a numbered list of the paragraphs to appear in the description.) Anything numbered 0 doesn't appear at all, and the default priority of an object is 1.

A thing can be early-described, late-described, latest-described, never-described, sightline-described, or ordinarily-described. A thing is usually ordinarily-described.

After choosing notable locale objects (this is the apply early and late description rule):

- repeat with item running through early-described things:
 - if there is a notable-object of item in the Table of Locale Priorities:
 - set the locale priority of the item to 1; [list before everything else -- this would work with any number lower than 5 and higher than 0]
 - repeat with item running through late-described things:
 - if there is a notable-object of item in the Table of Locale Priorities:
 - set the locale priority of the item to 10; [list after everything else -- this would work with any number larger than 5]
 - repeat with item running through never-described things:
 - set the locale priority of the item to 0; [don't list at all]
- continue the activity.

An important cautionary note: priorities are only honored if the objects are going to get their own paragraphs (with "writing a paragraph about..." or because they have initial appearances). Priorities do not affect the order in which items appear in the final "You can see..." list, except that items with priority 0 or lower are omitted. (If we want to order the items in that list, we may want to resort to the Complex Listing extension by Emily Short.)

There are further refinements available to us: for instance, we could make some things that are only visible if the player is raised above ground level.

After choosing notable locale objects (this is the sightline-described things are visible from supporters rule):

- if the player is not on a supporter:
 - repeat with item running through sightline-described things:
 - if there is a notable-object of item in the Table of Locale Priorities:
 - set the locale priority of the item to 0; [remove objects that can only be seen from higher objects.]
- continue the activity.

It may also be useful to know about the "parameter-object", which refers to the thing whose contents we are currently describing: the standard rules consider how to describe the contents of the location and then also check the contents of any supporter or container the player may be inside, so in the first case "parameter-object" would be the location, and then in the second the supporter in question.

In practice this is rarely useful, but should we need to change priorities in the case of both player and object being inside a particular container, we might make use of it, for instance:

A thing can be tasteful or icky. A thing is usually tasteful.

After choosing notable locale objects (this is the icky things next to players rule):
if the player is on the parameter-object:
repeat with item running through icky things :
if there is a notable-object of item in the Table of Locale Priorities:
set the locale priority of the item to 10; [remove objects that can only be seen from higher objects.]
continue the activity.

The other thing to note is that by default that final collection of generic objects ("You can also see...") appears at the end, regardless of the priority of everything else. If we really wanted to, though, we could force something to appear even after that paragraph, by adding a new listing rule to the locale description rules:

After choosing notable locale objects (this is the latest-described items priority rule):
repeat with item running through latest-described things:
if the item is a notable-object listed in the Table of Locale Priorities:
now the item is mentioned;
now the item is marked for late listing.

The late listing rule is listed after the you-can-also-see rule in the for printing the locale description rules.

A thing can be marked for late listing. A thing is usually not marked for late listing.

This is the late listing rule:
if something is marked for late listing:
say "Oh! And also [a list of things which are marked for late listing].";
now everything is not marked for late listing;
continue the activity.

Section 2 - Scenario

The Priority Lab is a room. The early bird, the worm, the leaf, the unseen object, the pebble, the twig, and the late edition are things in the Priority Lab.

The early bird is early-described. The late edition is late-described. The unseen object is never-described.

The worm is icky.

The high window is in Priority Lab. It is sightline-described and fixed in place. The initial appearance of the high window is "There's a tiny high window up near the ceiling that you can't see unless you're on top of something."

In order for the priorities we just set to be interesting, let's give out some initial appearances and writing a paragraph rules:

The initial appearance of the worm is "A worm inches along the ground."
The initial appearance of the late edition is "Finally, the late edition lies at your feet."

Rule for writing a paragraph about the early bird when the early bird is in a room: say "The early bird always appears first, and here it is."

Rule for writing a paragraph about the leaf: say "Look, there's [a leaf][unless the leaf is in the location] on [the holder of the leaf][end if]!"

Rule for writing a paragraph about an icky thing (called icky item) which is on something which supports the player: say "Ew, [an icky item] is right next to you."

This procedure also means (as you can test by experiment) that after the late edition has been picked up and dropped again, it lists in no special order in the "you can see..." paragraph (since initial appearances only print when the object has not yet been moved).

The afterthought is a thing in the Priority Lab. It is latest-described.

The bar stool is an enterable supporter in Priority Lab.

Test me with "get leaf / drop leaf / look / x unseen object / get pebble / look / get twig / look / get afterthought / look / drop twig / look / get late edition / look / drop late edition / sit on bar stool / look / get all / put all on stool / look".

359

★ Example Low Light

WI

An object that is only visible and manipulable when a bright light fixture is on.

Suppose we want a different treatment of lighting than the usual: the room isn't totally dark, but there's something we can't see unless we turn on a bright light.

"Low Light"

First we make our environment and its light:

The Workroom is a room. The desk is in the Workroom. The brilliant lamp is a device on the desk.

To decide whether the light level is high:
if the brilliant lamp is switched off, no;
if the player cannot see the brilliant lamp, no;
yes.

To decide whether the light level is low:
if the light level is high, no;
yes.

Now we make a shadow so that the player can only refer to it if the shadow is in inventory or the light is on:

The shadow is a privately-named thing on the desk.

Understand "barely-visible" or "barely visible" or "shadow" as the shadow when the light level is high. Understand "invisible" or "shadow" as the shadow when the player encloses the shadow.

And finally a couple of extra touches to make it clear why we're able to interact with the shadow when it's in inventory, even if the light is low:

Before printing the name of the shadow:
if the light level is high:
say "barely-visible ";
otherwise if the player encloses the shadow:
say "invisible (but tangible) "

After dropping the shadow when the light level is low:
say "You let it go and it fades into the ambient gloom."

To handle the appearance of the object, we want to set its locale priority to 0: that will prevent it being named in room descriptions.

After choosing notable locale objects:
unless the light level is high:
set locale priority of the shadow to 0.

Test me with "look / get shadow / turn on lamp / look / get shadow / i / turn off lamp / i / drop shadow / look / get shadow / turn on lamp / look".

4 Example Slightly Wrong

WI

A room whose description changes slightly after our first visit there.

A fairly common effect in interactive fiction is a room which is described differently on the first visit than on subsequent visits. We can produce this effect as follows:

"Slightly Wrong"

Awning is a room. "A tan awning is stretched on tent poles over the dig-site, providing a little shade to the workers here; you are at the bottom of a square twenty feet on a side, marked out with pegs and lines of string. Uncovered in the south face of this square is an awkward opening into the earth."

Slightly Wrong Chamber is south of the Awning. "[if unvisited]When you first step into the room, you are bothered by the sense that something is not quite right: perhaps the lighting, perhaps the angle of the walls. [end if]A mural on the far wall depicts a woman with a staff, tipped with a pine-cone. She appears to be watching you."

Test me with "look / s / look".

Note the "[if unvisited]..." in the description of the Slightly Wrong Chamber. A room is considered to be "unvisited" until after the player has seen its description for the first time.

The bracketed text creates a special rule for printing; we will learn more about these in the sections on text with variations and text with substitutions.

Some further fine print: we might write our condition as "if unvisited", "if the location is unvisited", or "if the Chamber is unvisited" -- all of these constructions would be acceptable, but in the absence of more specifics, the condition is understood to apply to the object whose description it is.

357



Example The Eye of the Idol

WI

A systematic way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description, and to control whether supporters list their contents or not.

"The Eye of the Idol"

Section 1 - Reusable Material

We start by defining relations that let us know where items "belong", with the understanding that if something is where it belongs, it will be described in the main room description and therefore should not be separately listed. Thus:

Positioning relates various things to various things. The verb to be placed in means the positioning relation. The verb to be placed on implies the positioning relation.

Room-positioning relates various things to various rooms. The verb to be room-placed in means the room-positioning relation.

We can't make relations relate various objects to various objects, and rooms are not things, so two separate cases are necessary. An alternative approach would be to say "A thing has an object called the initial placement", which would allow a thing to have an initial placement that was a room, a supporter, or a container; an advantage of using relations, though, is that that way we can if we like specify multiple placements for the same object, so that, e.g., a sparkling diamond can be described in the main description paragraph as "half-buried in dust" in the beginning of the game, and then at the end as "in the eye of the idol" at the end.

Now we define, based on these relations, an "in-place" adjective, which will identify whether something is in a location which will specially describe it:

Definition: a thing (called prop) is in-place:
if the prop is in the location and the prop is room-placed in the location, yes;

if the holder of the prop is a thing and the prop is placed in the holder of the prop, yes;
no.

Definition: a thing is out-of-place if it is not in-place.

With that done, removing these items automatically from the room description is actually pretty easy:

Before listing nondescript items:
now every marked for listing in-place thing is not marked for listing.

One tricky case remains: when something is placed on a supporter that is scenery, it can be mentioned even if we have marked that object "not marked for listing". What matters here is not whether the object itself is marked for listing but whether the supporter has been "mentioned". (A fuller description of how room descriptions are assembled is available in the Looking section of the Commands chapter in the Recipe Book.) So let's also add a feature whereby we can easily suppress the descriptions of these supporters when appropriate:

A supporter can be quiet.

A quiet supporter is one that is never mentioned itself and which only mentions its contents if they are out of place. This allows for maximum flexibility in incorporating it into the body of room descriptions.

Rule for writing a paragraph about a quiet supporter (called chosen table):
if an out-of-place thing is on the chosen table:
if an in-place thing is on the chosen table,
say "On [the chosen table], in addition to [the list of in-place things on the chosen table], [is-are a list of out-of-place things which are on the chosen table].";
otherwise say "On [a chosen table] [is-are a list of out-of-place things which are on the chosen table].";
now the chosen table is mentioned.

Notice that we can still override this with writing a paragraph rules about specific supporters in our game, if we decide that we want something a little different in some cases.

Now, an example to test this out:

Section 2 - A Sample Scenario

The Sand-Floored Chamber is a room. "The constant wind has filled this chamber with a layer of fine red sand, as soft as powder snow[if the diamond is in the Sand-floored Chamber]. Something sparkling is half-buried in the corner[end if]. A doorway lies open to the north."

The sparkling diamond is in the Sand-floored Chamber. The sparkling diamond is room-placed in the Sand-floored Chamber. The description is "It is a vast diamond; the front is faceted, the back smoothed to fit in some sort of socket."

The Hexagonal Temple is north of the Sand-Floored Chamber. "The temple walls are great ashlar blocks rising to a hundred feet overhead, perhaps more; the roof is a scarlet awning only, through which the sun filters down in blood hues. Overseeing all is a sculpture in stone and ivory[if the sparkling diamond is in the idol's eye], in whose single eye a vast diamond gleams[end if][mat-and-incense text]."

To say mat-and-incense text:

if the mat is in the Temple and the incense stick is on the pedestal:
say ". A prayer mat at the idol's feet, and an incense stick still burning on the pedestal, indicate that someone was only recently consigning her grievances to the care of the deity";

otherwise if the mat is in the Temple:

say ". At the idol's feet, some worshipper has left a prayer mat";

otherwise if the incense stick is on the pedestal:

say ". At the idol's side is a pedestal, on which incense still smolders".

We could have done all this with text conditions in the main room description, but it becomes difficult to read when there are too many conditions operating in the same text property, so we break it out into a clearer set of conditions.

The idol is scenery in the Hexagonal Temple. Understand "sculpture" or "stone" or "ivory" as the idol. The description is "The idol is perhaps three times the height of an ordinary man."

The idol's eye is part of the idol. It is a container. The description is "[if the diamond is in the idol's eye]It gleams with purpose and righteous wrath[otherwise]A round socket in the center of the idol's forehead from which something seems to be missing[end if]."

The pedestal is a quiet supporter in the Hexagonal Temple. On the pedestal is an incense stick. The incense stick is placed on the pedestal.

A mat is in the Hexagonal Temple. It is room-placed in the Hexagonal Temple. The description is "Woven of assorted grasses."

Test me with "get diamond / look / n / get mat / look / drop diamond / look / get diamond / put diamond in eye / look / get incense / look / drop mat / look / get mat / put mat on pedestal / look / put incense on pedestal / look".

362



Example Copper River

WI

Manipulating room descriptions so that only interesting items are mentioned, while objects that are present but not currently useful to the player are ignored.

In a very dense environment, we might want to offer the player room descriptions in which only the currently-interesting items are mentioned, while other objects are suppressed even if they are present. In effect, this takes the idea of scenery and makes it more flexible: different things might become background objects or foreground objects at different times during play.

There are a wide range of possible reasons to do this -- to shift the narrative emphasis, to change the mood of the game by highlighting different parts of the environment, to show the game from the perspective of different viewpoint characters -- but in the following example, our goal is to show the player only the objects that are currently useful for puzzles.

To do this, we need some notion of what puzzles are currently available and unsolved, so we make an "unsolved" adjective; we also need to know which things solve the puzzle, so we create a "resolving" relation, to indicate which objects resolve which problems.

Given that information, we can create rules about which objects in the game world are currently interesting, which are currently dull, and describe accordingly:

"Copper River"

Use scoring.

Section 1 - Procedure

Resolving relates various things to various things. The verb to resolve means the resolving relation.

Definition: a thing is interesting if it is not dull.

Definition: a person is dull:
no.

Definition: a thing is dull:
if it is unsolved, no;
if it resolves an unsolved thing, no;
yes.

Definition: a supporter is dull:
if it is unsolved, no;
if it resolves an unsolved thing, no;
if it supports an interesting thing, no;
yes.

Definition: a container is dull:
if it is unsolved, no;
if it resolves an unsolved thing, no;
if it contains an interesting thing, no;
yes.

After choosing notable locale objects:
repeat with item running through unsolved things:
set the locale priority of the item to 1.

For printing a locale paragraph about a dull thing (called item):
now the item is mentioned.

Before printing a locale paragraph about a supporter (called item):
now every dull thing on the item is mentioned.

Before printing a locale paragraph about a container (called item):
now every dull thing on the item is mentioned.

Instead of searching a supporter:
if the noun supports something interesting:
say "[A list of interesting things on the noun] [are] on [the noun]";
if the noun supports something dull:
say " (alongside [a list of dull things on the noun])";
say " .";
otherwise if the noun supports something dull:
say "There's nothing very useful here, only [a list of dull things on the noun].";
otherwise:
say "[The noun] [are] completely bare."

Instead of searching a container:
if the noun contains something interesting:
say "[A list of interesting things in the noun] [are] in [the noun]";
if the noun contains something dull:
say " (alongside [a list of dull things in the noun])";
say " .";
otherwise if the noun contains something dull:
say "There's nothing very useful here, only [a list of dull things in the noun].";
otherwise:
say "[The noun] [are] completely empty."

Before listing contents when not taking inventory: group dull things together.

Rule for grouping together dull things: say "assorted dull items".

Section 2 - Scenario World and Objects

The Kitchen is a room. "Your Aunt Fiona's kitchen looks as though it has been at the eye of a glitter storm. Fine, sparkling grit dusts every surface. The appliances are slightly askew, too, as though they hadn't quite settled after a vigorous earthquake."

The shelf is a scenery supporter in the Kitchen. On the shelf is a can of beans, a can of potato leek soup, and a tin of deflating powder.

The cabinet is a scenery container in the Kitchen. In the cabinet is a book of matches, a bottle of descaling solution, a fish hook, and a rusty knife. It is openable and closed.

The counter is a scenery supporter in the Kitchen. On the counter is an espresso machine, a blender, and a mortar. The blender and the mortar are containers. In the mortar is a pestle. Understand "countertop" as the counter.

The stove is a scenery supporter in the Kitchen. The oven is part of the stove. The oven is a closed openable container.

The refrigerator is a fixed in place container in the Kitchen.

Understand "fridge" as the refrigerator.

The description is "The refrigerator is a dull blue-green, and has a puffy, marshmallow texture on the outside, which means that it's no good for sticking magnets to. Aunt Fiona has never been willing to explain where she got it." The refrigerator is openable and closed.

In the refrigerator are a bottle of ice wine, a bag of carrot sticks, and an egg.

Aunt Fiona is a woman in the Kitchen. Aunt Fiona can be inflated or deflated. Aunt Fiona is inflated. "[if Aunt Fiona is inflated]Aunt Fiona stands nearby. Or perhaps 'stands' is the wrong word: she has been sort of puffed up in her own skin like a balloon, and is now propped in a corner of the room with her head lolling back[otherwise]Aunt Fiona stands -- on her own two slender legs -- at the center of the room[end if]."

Every turn when Fiona is unsolved and Fiona can see the player:

if a random chance of 1 in 3 succeeds:

say "[one of]Aunt Fiona's eyes follow you, wide and desperate, but it doesn't look like she's able to do anything[or]Aunt Fiona is still looking reproachful[or]A faint gurgling comes from Aunt Fiona[or]Aunt Fiona makes a funny croak noise[or]Aunt Fiona is still having trouble speaking. Perhaps her throat is as swollen as the rest of her[or]Aunt Fiona twitches[stopping]."

There is a thing called a salmon. Understand "fish" as the salmon. The salmon can be scaly or prepared. The salmon is scaly. The description is "[if scaly]It looks delicious, but is still covered with scales[otherwise]The salmon has been scaled and is ready to eat[end if]."

Before printing the name of the salmon when the salmon is scaly:

say "very scaly".

Section 3 - Scenario Puzzles

Definition: Aunt Fiona is unsolved if she is inflated.

Definition: the salmon is unsolved:

if the salmon is off-stage, no;

if the salmon is scaly, yes;

no.

The deflating powder resolves Aunt Fiona.

Instead of putting the deflating powder on Aunt Fiona:

try throwing the deflating powder at Aunt Fiona.

Instead of giving the deflating powder to Aunt Fiona:

try throwing the deflating powder at Aunt Fiona.

Instead of throwing the deflating powder at Aunt Fiona:

if Aunt Fiona is inflated:

say "You toss some of the powder in Aunt Fiona's direction, and with a sudden gaseous HUFF! she returns to her usual shape and size. [paragraph break]'Well!' she says, brushing herself off. 'That was bracing!' [paragraph break]You give her an embarrassed smile, to apologize for not curing her faster.";

now Aunt Fiona is deflated;

increase the score by 2;
otherwise:
say "[one of]You throw another hefty dose of the powder at your aunt.
[paragraph break]'Thank you, child,' she says, sneezing. 'But I think you've done
enough now.'[or]You throw another hefty dose of the powder at your aunt.
[paragraph break]'You're too kind,' she wheezes, through a cloud of glittering
dust.[or]You've probably done enough with the powder.[stopping]".

Every turn when Aunt Fiona is deflated and the salmon is off-stage:
move the salmon to the counter;
say "'At least they didn't get this,' she says, producing from somewhere on her
person a fresh-caught salmon. An odd pattern around its eye sockets makes it
looks comically as though it wears spectacles. 'It's the Salmon of Knowledge,'
she explains casually. 'We just need to scale and cook it.'"

The bottle of descaling solution resolves the salmon.

Does the player mean putting the descaling solution on the fish hook: it is
unlikely.
Does the player mean putting the descaling solution on the salmon: it is very
likely.

Instead of putting the bottle of descaling solution on the salmon:
if the salmon is scaly:
now the salmon is prepared;
say "With just a single squirt of the descaling solution (which confusingly
has a picture of bathroom tiles on the label), you remove the scales from the
salmon, leaving its pink flesh ready for preparation.";
increase the score by 2;
otherwise:
say "'Don't do that,' Aunt Fiona warns you. 'Excessive applications could
damage the flesh.'"

Test me with "look / get powder / drop powder / look / look in cabinet / get
powder / put powder on fiona / look / open cabinet / look in cabinet / get solution
/ open fridge / put solution in fridge / look / get solution / put solution on salmon /
look".

5

★ Example Port Royal 1

WI

A partial implementation of Port Royal, Jamaica, set before the
earthquake of 1692 demolished large portions of the city.

"1691"

Fort James is a room. "The enclosure of Fort James is a large, roughly
hexagonal court walled with heavy stone. The walls face the entrance to Port
Royal Harbour, and the battery of guns is prepared to destroy any enemy ship
arriving."

Unless we arrange otherwise, this will be the first room in the game because it is the
first we have defined.

For subsequent rooms, we do not have to say explicitly that they are rooms, as long as they are connected to a room on the map. For instance, this will automatically make Thames Street End a room:

Thames Street End is south of Fort James. "The ill-named Thames Street runs from here -- at the point of the peninsula -- all the way east among houses and shops, through the Fish Market, edging by the round front of Fort Carlisle, to the point where the town stops and there is only sandy spit beyond. Lime Street, wider and healthier but not as rich, runs directly south, and to the north the road opens up into the courtyard of Fort James."

Water Lane is east of Thames Street End. "Here Thames Street -- never very straight -- goes steeply southeast for a portion before continuing more directly to the east.

Water Lane runs south toward Queen Street, and facing onto it is the New Prison -- which, in the way of these things, is neither. It did serve in that capacity for a time, and in a measure of the villainy which has been usual in Port Royal from its earliest days, it is nearly the largest building in the town."

If we have some concern that the room name will be confused with an existing name, we can be more explicit about it using "called":

East of Water Lane is a room called Thames Street at the Wherry Bridge. Thames Street at the Wherry Bridge has the description "To the southwest is the fishmarket; directly across the street is the entrance to a private alley through a brick archway."

The Private Alley is south of Thames Street at the Wherry Bridge. "You're just outside the tavern the Feathers. To the north, under a pretty little archway, is the active mayhem of Thames Street, but the alley narrows down to a dead end a little distance to the south."

And now we get "inside", which generates a space treated as its own area on the map.

The Feathers is inside from the Private Alley. "Newly built with brick, replacing the older Feathers tavern that used to stand here. It sells wines in quantity, as well as serving them directly, and the goods are always of the best quality. There's a room upstairs for those wanting to stay the night." The Feathers Bedroom is above the Feathers.

And if we like we can declare a number of rooms for which we will come back and write the descriptions later. There is no obligation for the description to occur at the first definition of the room.

Lime Street is south of Thames Street End.

For efficiency, we can also write multiple sets of connections at once:

Queen Street East is east of Queen Street Middle and south of Private Alley.

Clicking Go will translate this description into a sketchy but working simulation of Port Royal, in which we can type movement commands like EAST or SOUTH to

explore the streets. Looking at the World tab of the Index, we can also see a schematic map of the simulation as it currently stands. Like the rest of the Index, this is provided entirely for the author's benefit, and is not visible to the player. (Though if we do decide that we want players to have access to a printed map while they play, Inform can help: we will return to the layout of Port Royal in the chapter on Publishing.)

The following Test command allows us to type TEST ME and explore the map we just devised:

Test me with "s / e / e / s / in".

8 **★ Example Port Royal 2**

WI

Another part of Port Royal, with less typical map connections.

"1691"

Thames Street End is a room.

If we check out a map of historic Port Royal, we find that Thames Street End bends around the northwest tip of the peninsula and becomes the (very) roughly north/south Fisher's Row. We can't put Fisher's Row south of Thames Street End, though, because Lime Street is already going that way. So instead, let's have a map connection that bends around from west to north:

West of Thames Street End is north of Fisher's Row.

Now continuing west along Thames Street, or north along Fisher's Row, will bring us around the corner in question. Asymmetric map connections should be used carefully. They're good for representing the layout of the real world, which tends not to be laid out on a convenient square matrix, but if exits are not described clearly they can be disorienting for the player. So let's be sure to make things clear:

The description of Fisher's Row is "A waterfront street that runs south towards Chocolata Hole, where the small craft are harboured. It also continues north around the tip of the peninsula from here, turning into the east-west Thames Street."

Meanwhile, suppose Fort James is in a prominent position, raised a bit from its surroundings; maybe the player should be able to go down from there, as well as south, to get to Thames Street End.

Thames Street End is down from Fort James. Thames Street End is south from Fort James.

But we don't want the upward direction to work:

Up from Thames Street End is nowhere.

Test me with "n / d / u / w / e / n / s".

10

Example Port Royal 3

WI

Division of Port Royal into regions.

"1691"

We should go ahead and do all our room definitions first...

Fort James is a room. "The enclosure of Fort James is a large, roughly hexagonal court walled with heavy stone. The walls face the entrance to Port Royal Harbour, and the battery of guns is prepared to destroy any enemy ship arriving."

Thames Street End is south of Fort James. "The ill-named Thames Street runs from here -- at the point of the peninsula -- all the way east among houses and shops, through the Fish Market, edging by the round front of Fort Carlisle, to the point where the town stops and there is only sandy spit beyond. Most of that stretch is full of people at all hours. Imported goods are moved off of ships and taken to distributors; exported goods are brought to be loaded; and there is one public house and brothel for every ten inhabitants.

Lime Street, wider and healthier but not as rich, runs directly south, and to the north the road opens up into the courtyard of Fort James."

Lime Street is south of Thames Street End. West of Thames Street End is north of Fisher's Row. The description of Fisher's Row is "A waterfront street that runs south towards Chocolata Hole, where the small craft are harboured. It also continues north around the tip of the peninsula from here, turning into the east-west Thames Street."

Thames Street End is down from Fort James. Up from Thames Street End is nowhere.

Water Lane is east of Thames Street End. "Here Thames Street -- never very straight -- goes steeply southeast for a portion before continuing more directly to the east.

Water Lane runs south toward Queen Street, and facing onto it is the New Prison -- which, in the way of these things, is neither. It did serve in that capacity for a time, and in a measure of the villainy which has been usual in Port Royal from its earliest days, it is nearly the largest building in the town."

East of Water Lane is a room called Thames Street at the Wherry Bridge. Thames Street at the Wherry Bridge has the description "To the southwest is the fishmarket; directly across the street is the entrance to a private alley through a brick archway."

The Fishmarket is southwest of Thames Street at the Wherry Bridge.

The Private Alley is south of Thames Street at the Wherry Bridge. "You're just outside the tavern the Feathers. To the north, under a pretty little archway, is the active mayhem of Thames Street, but the alley narrows down to a dead end a little distance to the south."

The Feathers is inside from the Private Alley. "Newly built with brick, replacing the older Feathers tavern that used to stand here. It sells wines in quantity, as well as serving them directly, and the goods are always of the best quality. There's a room upstairs for those wanting to stay the night." The Feathers Bedroom is above the Feathers.

Thames Street by the King's House is east of Thames Street at the Wherry Bridge. "The King's House is reserved for the use of the Governor, but he does not live in it, and it is frequently being rented out to some merchant so that the government will at least derive some value from it. It is nearly the least interesting establishment on Thames Street, and the crowd -- which, to the west, is extremely dense -- here thins out a bit."

Thames Street before Fort Carlisle is east of Thames Street by the King's House. "Here Thames Street, formerly a respectable width, narrows to a footpath in order to edge around the front of Fort Carlisle, underneath the mouths of the cannon.

There are no buildings on the harbour side of Thames Street at this point, which means that you have an unusually good view of the ships at dock, water beyond, and the Blue Mountains rising on the other side of the harbour."

South of Thames Street before Fort Carlisle is a room called Fort Carlisle. The description of Fort Carlisle is "Handsomely arrayed with cannons which you could fire at any moment -- though of course there are ships at dock which might be in the way."

Queen Street End is south of Lime Street.

Queen Street Middle is east of Queen Street End.

Queen Street East is east of Queen Street Middle and south of Private Alley.

Queen Street at the Prison is east of Queen Street East.

Now, if we like, we can create regions to distinguish the coast from the portions of town that aren't on the water:

Inland is a region. Queen Street End, Queen Street Middle, Queen Street East, Private Alley, Lime Street, and Queen Street at the Prison are in Inland.

Waterfront is a region. Thames Street before Fort Carlisle, Thames Street by the King's House, Thames Street at the Wherry Bridge, Water Lane, Fishmarket, Fisher's Row, and Thames Street End are in Waterfront.

There's no rule that regions must be contiguous, so we could if we like make a region consisting just of the two forts:

Military Holdings is a region. Fort Carlisle and Fort James are in Military Holdings.

And we might make the Feathers Tavern part of the Inland area, but within its own subcategory:

Tavern is a region. It is in Inland. Feathers and Feathers Bedroom are in Tavern.

Now the index map will be colored to reflect our regions, and later in the game development we would be able to make rules that affect just one region at a time.

78

Example All Roads Lead to Mars

WI

Layout where the player is allowed to wander any direction he likes, and the map will arrange itself in order so that he finds the correct "next" location.

Suppose we want to allow the player to wander freely in any direction, but ourselves maintain control over the order in which he encounters the rooms. This sort of effect emphasizes the order of the story-telling over any kind of rigorous simulation of space; on multiple play-throughs, the player might not find all the same rooms in the same locations.

"All Roads Lead to Mars"

Before going a direction (called way) when a room (called next location) is not visited:

- let further place be the room the way from the location;
- if further place is a room, continue the action;
- change the way exit of the location to the next location;
- let reverse be the opposite of the way;
- change the reverse exit of the next location to the location.

The Open Plain is a room. "A wide-open grassy expanse, from which you could really go any way at all."

The Hilly Place is a room. "The grassland gives way to a somewhat more hilly area, though there is still very little to guide you any particular way."

The Stream is a room. "This is the third place you've been today, and so the stream is welcome. How refreshing!"

Test me with "n / s / e / e".

If we wanted still to be able to find routes between places, we could define a relationship of connection between rooms, which we would add to as we went along.

125

Example Bee Chambers

WI

A maze with directions between rooms randomized at the start of play.

Mazes are a traditional element of interactive fiction, often consisting of apparently identical rooms with exits that do not work reciprocally and which cause confusion.

The methods of mapping mazes are now fairly well understood and mazes themselves tend to be regarded as tiresome rather than enjoyable by a large portion of the playing audience. However, if we did want to ignore the common wisdom and create a maze, randomly generated at the start of play, here would be one way to go about it:

"Maze of Gloom"

A Bee Chamber is a kind of room. The printed name of a Bee Chamber is usually "Hexagonal Room". The description of a Bee Chamber is usually "Waxy, translucent walls surround you on six sides; the floor and ceiling are made of the same material, gently uneven. There are exits in every direction, cut into the faces or the corners."

Bee1, Bee2, Bee3, Bee4, Bee5, Bee6, Bee7, Bee8, Bee9, and Bee10 are Bee Chambers.

When play begins:

now right hand status line is "[number of visited rooms]/[number of rooms]";

repeat with place running through Bee Chambers:

now a random Bee Chamber is mapped north of place;
now a random Bee Chamber is mapped northwest of place;
now a random Bee Chamber is mapped west of place;
now a random Bee Chamber is mapped southwest of place;
now a random Bee Chamber is mapped south of place;
now a random Bee Chamber is mapped southeast of place;
now a random Bee Chamber is mapped east of place;
now a random Bee Chamber is mapped northeast of place;
now a random Bee Chamber is mapped above place;
now a random Bee Chamber is mapped below place;
now a random Bee Chamber is mapped inside place;
now a random Bee Chamber is mapped outside place.

Test me with "in / out / up / down / n / ne / nw / e / w / sw / se / s".

134

★ Example Zork II

A "Carousel Room", as in Zork II, where moving in any direction from the room leads (at random) to one of the eight rooms nearby.

WI

All we need to do is select the player's destination for him at random:

"Zork II"

The Carousel Room is a room.

Instead of going from the Carousel Room:
move the player to a random adjacent room.

To avoid infringing the original game too much, let's try a somewhat different setting:

The Games of Chance is north of the Carousel Room. The Haunted Funhouse is northwest of the Carousel Room. The Ferris Wheel is east of the Carousel Room. The Topsy-Turvy is northeast of the Carousel Room. The Reproduction Henge is south of the Carousel Room. The Women's Toilet is southwest of the Carousel Room. The Men's Toilet is southeast of the Carousel Room. The Cotton Candy Shop is west of the Carousel Room.

Test me with "s".

And the following means that the test runs consistently even though the numbers are theoretically random. To make them truly random, remove this line.

When play begins, seed the random-number generator with 1234.

Or if we want to add the refinement that the Carousel Room can be switched off:

"Zork II"

The Carousel Room is a room. The spinning machine is a switched on device in the Carousel Room.

And then

Instead of going from the Carousel Room when the spinning machine is switched on:
move the player to a random adjacent room.

The Games of Chance is north of the Carousel Room. The Haunted Funhouse is northwest of the Carousel Room. The Ferris Wheel is east of the Carousel Room. The Topsy-Turvy is northeast of the Carousel Room. The Reproduction Henge is south of the Carousel Room. The Women's Toilet is southwest of the Carousel Room. The Men's Toilet is southeast of the Carousel Room. The Cotton Candy Shop is west of the Carousel Room.

Test me with "turn off machine / s / n / turn on machine / s".

When play begins, seed the random-number generator with 1234.

In Mayan culture, colours seem to have been used as names for the primary directions: for instance, "red" implies east as the colour of sunrise. So the following might be a stylish touch for a game in which the player has to get inside the Mayan world-view:

"Indirection"

Understand "white" and "sac" as north. Understand "red" and "chac" as east. Understand "yellow" and "kan" as south. Understand "black" and "chikin" as west.

We could also use a colour as a verb:

Understand "turquoise" and "yax" as looking.

And now a few extra rooms to try it out in:

The Square Chamber is a room. "A sunken, gloomy stone chamber, ten yards across. A shaft of sunlight cuts in from the steps above, giving the chamber a diffuse light, but in the shadows low lintelled doorways to east and south lead into the deeper darkness of the Temple."

The Wormcast is east of the Square Chamber. The Corridor is south of the Square Chamber.

Test me with "kan / white / chac / black".

40



Example Prisoner's Dilemma

WI

A button that causes a previously non-existent exit to come into being.

We can change the directions in the map in mid-game, though in practice this is rarely necessary. But suppose we do not want a door or any sign of a door to exist before the player takes some action, in this case pressing a button:

"Prisoner's Dilemma"

Challenger's Waiting Room is a room. "The challenge is this: to wait as long as you can endure to do so in a room with no features and no clock. If you wait longer than all the other contestants, you win."

The button is fixed in place in the Challenger's Waiting Room. "The only item in view is a black recessed button."

Amid the Cheering Throng is a room.

Instead of pushing the button for the first time:

change the east exit of the Challenger's Waiting Room to Amid the Cheering Throng;

change the west exit of the Cheering Throng to the Challenger's Waiting Room;
say "With a groan of gears, the east wall swings open! If you've lost now, well, you've lost..."

Test me with "e / push button / e / w".

Our instructions about pushing the button will be further explained in the chapter on Actions, but the thing to note here is that we can "change (whatever) exit" in order to set or re-set map directions. Notice that we have to set both directions explicitly: changing the east exit of the Waiting Room does not automatically also change the west exit of Amid the Cheering Throng.

This allows greater flexibility in our games but does require an extra line or so of work.

41



Example The World of Charles S. Roberts

WI

Replacing the ordinary compass bearings with a set of six directions to impose a hexagonal rather than square grid on the landscape.

Wargaming is an ancient pursuit, but its modern form began as a professional training exercise in 19th-century Prussian staff colleges; since at least as early as H. G. Wells's "Little Wars" (1913) it has been a hobby of "boys from twelve years of age to one hundred and fifty and for that more intelligent sort of girl who likes boys' games and books." The free-form tabletop game used miniature figures and tape-measured movements, and remains the dominant form today. But in the mid-20th century, map grids on printed sheets gave the hobby a sudden new lease of life. They were easier to set up, more interesting to look at, cheaper to sell by mail-order. 1970s sales figures for "Strategy and Tactics", the leading US subscription-based wargame distributor, were very similar to those of Infocom's IF games in the 1980s. And like classical IF, the grid-based wargame parceled up a continuous world into locations.

Grids were initially square, as on a chessboard, but square cells have several disadvantages. Four directions of movement (N, E, S, W) is too few, yet allowing movement in the diagonal directions means allowing tanks to travel about 1.4 times faster northeast than they do north. Square grids also only conform cleanly to man-made landscape features such as buildings in one orientation, and they never fit hills well. (A compromise measure to fix this, cutting the squares into octagons to leave smaller diamond squares at corner intersections, has never caught on.)



But following Charles S. Roberts's American Civil War designs for Avalon Hill of 1958-61 (notably "Chancellorsville" and the second edition of "Gettysburg"), a hexagonal grid became the new standard. Each hexagon is the same distance from the centre of all six of its neighbours, which are at equal angular spacings; and clumps of hexagons fit the shape of lakes, contoured hills, and so forth, much more naturally than clumps of squares do. Hexes also have a certain mystique - an air of "I don't belong in the children's department".

But hexes are tricky for IF, not least because English lacks words for "the direction 60 degrees around from front". Our cognitive view of the world tends to be square, perhaps because our two eyes both face front, in a direction at right angles to the plane of our arms, legs, pelvis and eyes. We reach out sideways at right angles to our walking. Even early hex-grid wargames called the cells "squares", though "hexes" eventually caught on. Still and all:

"The World of Charles S. Roberts"

Forward is a direction. Forward has opposite backward. Understand "f" as forward.

Backward is a direction. Backward has opposite forward. Understand "b" and "back" as backward.

Forward left is a direction. Forward left has opposite backward right. Understand "fl" as forward left.

Forward right is a direction. Forward right has opposite backward left. Understand "fr" as forward right.

Backward left is a direction. Backward left has opposite forward right. Understand "bl" as backward left.

Backward right is a direction. Backward right has opposite forward left. Understand "br" as backward right.

Now to forbid the use of the compass directions:

A direction can be hexagonal or squared-off. A direction is usually squared-off. Forward, backward, forward left, forward right, backward left and backward right are hexagonal.

Before going a squared-off direction, say "In this hexagonally-divided landscape, squared-off directions are not allowed." instead.

A slight nuisance is that, with things as they are above, typing BACKWARD produces the response "Which do you mean, backward, backward left or backward right?" To avoid that silly question, we write:

Does the player mean going backward: it is very likely. Does the player mean going forward: it is very likely.

And now a clump of 37 hexes, in six columns of six or seven rooms each. There are many ingenious ways we could put this map together automatically, but instead we will take a deep breath and write:

E1 is forward of E2. "Open farmland." E2 is forward of E3. "The edge of woods." E3 is forward of E4. "Deep woodland." E4 is forward of E5. "Deep woodland." E5 is forward of E6. "The rear edge of woods." E6 is forward of E7. "The start of a road leading forward right." E7 is a room. "Grassland."

F1 is forward of F2. "The edge of farmland." F2 is forward of F3. "The edge of woods." F3 is forward of F4. "Clearing in woods." F4 is forward of F5. "Deep woodland." F5 is forward of F6. "A road runs backward left to forward right." F6 is a room. "The edge of grassland."

G1 is forward of G2. "Grassland." G2 is forward of G3. "The edge of farmland." G3 is forward of G4. "A copse of trees." G4 is forward of G5. "The backward edge of woodland." G5 is forward of G6. "A bend in the road, from backward left to backward right." G6 is forward of G7. "Open farmland." G7 is a room. "Open farmland."

H1 is forward of H2. "Grassland, bordered by a hedge to the right." H2 is forward of H3. "The edge of farmland, with a hedge to forward right." H3 is forward of H4. "A copse of trees." H4 is forward of H5. "Open farmland." H5 is forward of H6. "A passing place on the road, which bends forward left to forward right." H6 is a room. "Open farmland."

I1 is forward of I2. "The end of a forward road, blocked by hedges on all sides except backward." I2 is forward of I3. "A straight road runs forward to backward, with long hedges to left and right." I3 is forward of I4. "A straight road runs forward to backward, alongside a long hedge to right." I4 is forward of I5. "A straight road runs forward to backward, alongside a long hedge to right." I5 is forward of I6. "Where three roads, forward, backward left and backward right, meet. Forward right is a thick hedge." I6 is forward of I7. "Open farmland." I7 is a room. "Open farmland."

J1 is forward of J2. "Dense woodland, with a hedge to left." J2 is forward of J3. "Grassland, with a hedge to left." J3 is forward of J4. "The edge of farmland, with a hedge to left." J4 is a room. "Open farmland, with a long hedge blocking movement forward left, backward left or backward." J5 is forward of J6. "A road

running forward left to backward right, alongside a hedge." J6 is a room. "Open farmland."

F1 is forward right of E2 and backward right of E1. F2 is forward right of E3 and backward right of E2. F3 is forward right of E4 and backward right of E3. F4 is forward right of E5 and backward right of E4. F5 is forward right of E6 and backward right of E5. F6 is forward right of E7 and backward right of E6.

G1 is forward right of F1. G2 is forward right of F2 and backward right of F1. G3 is forward right of F3 and backward right of F2. G4 is forward right of F4 and backward right of F3. G5 is forward right of F5 and backward right of F4. G6 is forward right of F6 and backward right of F5.

H1 is forward right of G2 and backward right of G1. H2 is forward right of G3 and backward right of G2. H3 is forward right of G4 and backward right of G3. H4 is forward right of G5 and backward right of G4. H5 is forward right of G6 and backward right of G5. H6 is forward right of G7 and backward right of G6.

I3 is forward right of H3 and backward right of H2. I4 is forward right of H4 and backward right of H3. I5 is forward right of H5 and backward right of H4. I6 is forward right of H6 and backward right of H5.

J5 is forward right of I6 and backward right of I5. J6 is forward right of I7 and backward right of I6.

And now we have a hexagonally-gridded world. Route-finding will work; prepositional forms like "to be mapped backward left of" exist, just as they should; and in general these directions are just as good as the square ones. (The only thing which doesn't look good is the Index map, where Inform is just unable to draw a picture because it assumes a square grid. But that has no effect on play.)

The landscape is much easier to navigate with a little diagram:

```
To say legend (D - direction):
  let destination hex be the room D from the location;
  if the destination hex is nothing, say " ";
  otherwise say the destination hex.
```

```
Carry out looking:
  say "[fixed letter spacing] \ [legend forward] /[line break][legend forward left] ---
- [legend forward right][line break] / \[line break]--< [location] >--[line break] \
/[line break][legend backward left] ---- [legend backward right][line break] /
[legend backward] \[variable letter spacing][line break]".
```

And finally:

The player is in I5.

Test me with "f / forward / backward left / bl / br / br / f".



Using regions to block access to an entire area when the player does not carry a pass, regardless of which entrance he uses.

Rules about going to regions make it easy to exclude the player from a large portion of the map, even if there are many connecting paths to the region. For instance, in this story it would be annoying to have to write a rule about all four exits by which the player could reach the film set area:

"A&E"

Winding Street is a room. Winding Street is west of Duck Pond. Sloping Street is north of Winding Street, northwest of Duck Pond, west of Stately Lawn, and southwest of Stately Home. Stately Lawn is north of Duck Pond. Stately Home is north of Stately Lawn.

Film Set is a region. Duck Pond, Stately Lawn, and Stately Home are in Film Set.

Instead of going to Film Set when the player does not carry the VIP Pass: say "A burly studio guard materializes in your path, convincing you that you would prefer to be elsewhere."

The VIP Pass is in the garbage can. The garbage can is in Sloping Street.

After going to the Film Set:

say "Success! At last you are inside the set of 'Prouder and More Prejudiced'.

Next step: locating Mr Firth.";

end the story finally.

Test me with "e / n / e / get pass / e".

199

★ Example Further Reasons Why All Poets Are Liars

WI

The young William Wordsworth, pushing a box about in his room, must struggle to achieve a Romantic point of view.

We begin with the location and its fittings, and we create a kind of value which names the different internal positions we will allow.

"Further Reasons Why All Poets Are Liars"

Nook Obscure is a room. "Above the College kitchens, which make a humming sound, less tuneable than bees, but hardly less industrious, with shrill notes of sharp command and scolding intermixed: and below Trinity's loquacious clock, who never lets the quarters, night or day, slip by him unproclaimed, and tells the hours twice over with a male and female voice. In short, the kind of rubbish room they give to a northern villager. But you get a bed and a high shelf all of your own. And you long to find some Romantic way to look out of the window."

The window, the shelf and the bed are scenery in the Nook Obscure. The shelf and the bed are supporters. The bed is enterable.

Internal position is a kind of value. The internal positions are nowhere at all, over by the window, under the shelf and near the bed.

The box is an enterable supporter in Nook Obscure. The current box position is an internal position that varies. The current box position is near the bed. "Your packing case, stamped W. WORDSWORTH (KENDAL), is [current box position]." Instead of taking the box, say "It is filled with your peerless rock collection and too heavy to lift, but could be pushed." Instead of opening the box, say "It is securely nailed shut."

We create an action, "pushing it over to", for pushing a box around on the floor of a single location. (Calling this "pushing it over to" prevents clashes with the existing "pushing it to" action, which is for pushing things from one room to another.) Almost half of the text which defines the action is concerned with the two action variables, but they make the implementation of everything else so much easier that we end up writing less than if we hadn't used them.

Understand "push [box]" as a mistake ("You can push the box to the window, the bed or the shelf.").

Understand "push [something] to [something]" as pushing it over to. Pushing it over to is an action applying to two things.

The pushing it over to action has an internal position called the old position. The pushing it over to action has an internal position called the new position.

Setting action variables for pushing something over to something:

- now the old position is the current box position;
- now the new position is nowhere at all;
- if the second noun is the window, now the new position is over by the window;
- if the second noun is the bed, now the new position is near the bed;
- if the second noun is the shelf, now the new position is under the shelf.

Check pushing it over to:

- if the noun is not the box, say "That's not something you can push." instead;
- if the player is on the bed, say "You can't reach from here." instead;
- if the player is on the noun, say "Not while you are standing on [the noun]." instead;
- if the new position is nowhere at all, say "You can only push [the noun] to the window, the bed or the shelf." instead;
- if the new position is the old position, say "The [noun] is already [new position]." instead.

Carry out pushing it over to:

- now the current box position is the new position.

Report pushing it over to:

- say "With some effort, you shove [the noun] from [old position] to [new position]."

Everything which remains simply provides a couple of puzzles to test this arrangement.

Euclid's Elements is on the shelf. Understand "euclid" or "book" as the Elements. Instead of taking something (called the item) which is on the shelf:

if the player is on the box and the current box position is under the shelf, continue the action;

say "You cannot reach [the item], which is up on the shelf."

Instead of examining the window:

say "This window opens rather unpromisingly onto the chapel wall opposite, so even granted the moonlight it is dark in here. Still, surely there's a poem here somewhere?"

Instead of examining the window when the player is on the bed:

say "Just a blank patch of chapel wall."

Instead of examining the window when the player is on the box:

if the current box position is near the bed:

say "Tantalisingly, you are not quite able to spy the statue.";

otherwise if the current box position is under the shelf:

say "All you can see is the antechapel wall, and the dull silver gleam of the pealing organ.";

otherwise:

say "At last! You can just, standing on tiptoes on the box right up at the window, make out the top of the statue! Of such epiphanies are Poesy born. Let's see now... oh yes...[paragraph break]And from my pillow, looking forth by light[line break]Of moon or favouring stars, I could behold[line break]The antechapel where the statue stood[line break]Of Newton with his prism and silent face,[line break]The marble index of a mind for ever[line break]Voyaging through strange seas of Thought, alone.";

end the story finally.

Test me with "get on bed / x window / get off / x window / get elements / get on box / x window / get elements / push box to shelf / get off / push box to shelf / get on box / get elements / x window / get off / push box to window / get on box / x window".

7



Example Starry Void

WI

Creating a booth that can be seen from the outside, opened and closed, and entered as a separate room.

Sometimes we may want a room to be visible from the outside in one location, but treated as a separate location when we are inside. The simplest way to do this is to make the exterior form of the object into a door object, and to describe it differently from different vantage points. (Doors in general are described more fully in the Doors section of the Things chapter.)

"Starry Void"

The Center Ring is a room.

The magician's booth is a door. "[if the player is in Center Ring]A magician's booth stands in the corner, painted dark blue with glittering gold stars.[otherwise

if the magician's booth is closed]A crack of light indicates the way back out to the center ring.[otherwise]The door stands open to the outside.[end if]".

Here we've arranged for the booth to be described in the initial room description in different ways depending on where the player is when viewing it. We might like to do the same if the player takes a closer look:

Instead of examining the magician's booth in the Center Ring:

say "It is dark blue and glittering with gold stars. [if the booth is open]The door currently stands open[otherwise]It has been firmly shut[end if]."

Instead of examining the magician's booth in the Starry Void:

say "The booth door is [if the magician's booth is open]wide open[otherwise]shut, admitting only a thin crack of light[end if]."

And now we put it in place:

The magician's booth is inside from Center Ring and outside from Starry Void.

...and make sure that the booth-and-door object responds to all the names we have used for it in different places:

Understand "door" or "of" or "the" or "light" or "crack" or "thin crack" as the booth.

Test me with "examine booth / open door of the booth / in / examine door / close door / look / examine crack of light".

A final nice touch, if we're so inclined, is to borrow from the Basic Actions chapter and make the player automatically open the booth door before trying to enter:

Before going through the closed magician's booth:

say "(first opening the door of the booth)[command clarification break]";
silently try opening the booth.

For the contrasting case of a space that is nested inside another place and is not its own room -- say a stall at an open-air market, or a rowboat on a lake -- see the example "Tamed".

216

★ Example Waterworld

WI

A backdrop which the player can examine, but cannot interact with in any other way.

It's tempting to handle the player's inability to interact with something with a simple instead rule:

"Waterworld 1"

A view is a kind of backdrop. Instead of doing something other than examining to a view, say "You are too far from [the noun] to do anything but look."

The sun is a view. It is everywhere. The description is "A blazing sun makes you wish you had never been born."

The Sahara is a room. North of the Sahara is More Sahara. North of More Sahara is Yet Further Sahara.

Test me with "x sun / get sun / n / x sun / n / x sun".

Unfortunately, the rule does not address the case where the object in question is the second noun; so for instance the following example reveals the difficulty:

"Waterworld 2"

A view is a kind of backdrop. Instead of doing something other than examining to a view, say "You are too far from [the noun] to do anything but look."

The player carries a rope.

The sun is a view. It is everywhere. The description is "A blazing sun makes you wish you had never been born."

The Sahara is a room. North of the Sahara is More Sahara. North of More Sahara is Yet Further Sahara.

Test me with "x sun / get sun / n / x sun / n / x sun / tie rope to the sun".

...where the response here behaves as though the sun is in reach. If we had a fully implemented tying action, the player would (even more disastrously) be allowed to lasso celestial objects.

We could add a second instead rule as well:

"Waterworld 3"

A view is a kind of backdrop.

Instead of doing something other than examining when the noun is a view:

say "You are too far from [the noun] to do anything but look."

Instead of doing something other than examining when the second noun is a view:

say "You are too far from [the second noun] to do anything but look."

The player carries a rope.

The sun is a view. It is everywhere. The description is "A blazing sun makes you wish you had never been born."

The Sahara is a room. North of the Sahara is More Sahara. North of More Sahara is Yet Further Sahara.

Test me with "x sun / get sun / n / x sun / n / x sun / tie rope to sun".

This produces acceptable output again, but there is a more elegant way, one that works better with Inform's existing world model. Currently the default model assumes that accessibility -- whether the player can reach something or not -- is checked between the Before... rules and the Instead... rules. We can add our own accessibility rules, including this one to govern whether views are accessible. So for instance:

"Waterworld 4"

A view is a kind of backdrop.

The can't touch views rule is listed before the access through barriers rule in the accessibility rulebook.

Accessibility rule (this is the can't touch views rule):

if the action requires a touchable noun and the noun is a view:
say "You are too far from [the noun] to do anything but look." instead;
if the action requires a touchable second noun and the second noun is a view:
say "You are too far from [the second noun] to do anything but look."
instead;

The player carries a rope.

The sun is a view. It is everywhere. The description is "A blazing sun makes you wish you had never been born."

The Sahara is a room. North of the Sahara is More Sahara. North of More Sahara is Yet Further Sahara.

Test me with "x sun / get sun / n / x sun / n / x sun / tie rope to sun".

Now our new accessibility rule fits into its proper stage.

62

☆☆ Example Tiny Garden

WI

A lawn made up of several rooms, with part of the description written automatically.

Sometimes we want to make a list of something too complicated to express in a say list... phrase. When this happens, we can instead mark all the items we want to mention as "marked for listing".

In this case, we have a lawn area made up of four rooms. We want each room to automatically describe the directions leading to the other parts of the lawn. To do this, we will first determine which directions are relevant and mark those for listing, then list them.

"Tiny Garden"

The Herb Garden is a room. "Along this side of the house run your great-aunt's herb beds."

A Grassy Room is a kind of room. The printed name of a Grassy Room is usually "Lawn". The description of a Grassy Room is "The grass underfoot is thick and green. The lawn extends to [grassy directions] from here."

The following phrase goes through all the directions in the compass and marks the ones that are interesting to us at the moment.

To say grassy directions:
repeat with that way running through directions:
if the room that way from the location is a grassy room,
now that way is marked for listing;
say "[a list of directions which are marked for listing]";
now every direction is not marked for listing.

Lawn1 is west of the Herb Garden. It contains a picnic table and a wicker basket. Lawn2 is south of Lawn1 and southeast of Lawn4. Lawn3 is southwest of Lawn1, west of Lawn2, and south of Lawn4. Lawn4 is west of Lawn1. Lawn4 contains a birdbath. The birdbath is fixed in place.

Lawn1, Lawn2, Lawn3, and Lawn4 are Grassy Rooms.

Test me with "w / s / w / n".

79



Example Hotel Stechelberg

WI

Signposts such as those provided on hiking paths in the Swiss Alps, which show the correct direction and hiking time to all other locations.

The following rule appends a paragraph to every room description. We need not worry about doors (despite the pass in the Bernese Oberland known figuratively as the "Little Door").

"Hotel Stechelberg"

After looking:
say "Yellow arms on the signpost point:-[line break]";
repeat with destination running through interesting rooms:
let the way be the best route from the location to the destination;
if the way is a direction, say " [way] for [the destination]: [number of moves from the location to the destination] Std."

Hotel Stechelberg is a room. "The wooden hiking inn at the end of the road, with flowerboxes, canton flags, outdoor tables and a triangular paddock for the cows contesting the annual Miss Stechelberg competition. Otto and Marianne do cheerful innkeeper things, while the sun blazes from a gentian-blue sky."

A room can be dull or interesting. A room is usually dull.

North of Hotel Stechelberg is Trummelbachfalle. North of Trummelbachfalle is Lauterbrunnen. Lauterbrunnen is interesting.

Southeast of Hotel Stechelberg is Trachsellaenen. Trachsellaenen is interesting.

Test me with "look".

With a bit more work, the result might be:

Hotel Stechelberg

The wooden hiking inn at the end of the road, with flowerboxes, canton flags, outdoor tables and a triangular paddock for the cows contesting the annual Miss Stechelberg competition. Otto and Marianne do cheerful innkeeper things, while the sun blazes from a gentian-blue sky.

Yellow arms on the signpost point:-

north for Lauterbrunnen: 2 Std.

west for Sefinental: 2 Std.

west for Schilthorn: 6 Std.

southeast for Trachsellaenen: 1 Std.

southeast for Oberhornsee: 3 Std.

212



Example Carnivale

WI

An alternative to backdrops when we want something to be visible from a distance but only touchable from one room.

Suppose we want to make an object that (unlike a backdrop) is definitely located in one room, but can be seen from far off. We want to allow the player to interact with it from a distance, but only using those actions that require visibility. Other actions should be denied:

"Carnivale"

The Fairground is a region. Park Entrance, By the Wheel, and Candy Stand are in Fairground. Candy Stand is north of By the Wheel. Park Entrance is west of Candy Stand and northwest of By the Wheel.

The ferris wheel is scenery in By the Wheel. "It is extravagantly tall and carries several dozen glass gondolas for riders."

The description of By the Wheel is "You stand at the foot of an enormous ferris wheel, which turns far too quickly and never seems to stop for new riders."

The description of Park Entrance is "You are now just inside the gates. Behind you snakes a triple line of fairgoers all the way down the length of the valley to the railway station. Roughly southeast of here is the ferris wheel, towering over the other attractions."

The description of Candy Stand is "A hut in pale pink and baby blue dispenses marshmallow death's-heads, sugar-beetles, and other such treats. The giant ferris wheel is just off to the south from here."

As the descriptions make clear, the ferris wheel should be visible from everywhere in the fair, so we'll borrow a line from the Activities chapter to make that happen:

After deciding the scope of the player:
if the location is in Fairground, place the ferris wheel in scope.

"Scope" determines what the player can interact with; by writing this rule, we make Inform understand all commands that refer to the ferris wheel when the player is anywhere in the fairground, instead of responding with

You can't see any such thing.

as it normally would.

Now, by default, if the player were to type TOUCH FERRIS WHEEL while in another room, he would get the response

You can't reach into By the Wheel.

This may not be quite what we want, but we can replace this text with our own reaching inside rule:

Rule for reaching inside a room:
say "You can only look from this distance."
deny access.

And because our accessibility rules are considered before the "Instead" phase, we can write the following rule confident that it will apply only when the player is in fact in range to touch the ferris wheel:

Instead of touching the ferris wheel:
say "You don't dare: it's spinning too fast."

Test me with "x ferris wheel / touch ferris wheel / se / x ferris wheel / touch ferris wheel".



Using the compass directions in commands is a little bit finicky because directions are forbidden to figure in any interactions involving touch. (Really, directions are more a concept than an object; this is a compromise situation.) In any case, if we want to write a new command involving these, we need to be sure to specify that the direction is a visible thing. For instance:

"Eddystone"

The Lighthouse is a room. "A lonely place, but in these tense times, no one but the lighthouse keeper and a few trusted agents are allowed on the grounds at all, for fear of sabotage."

The light is a fixed in place thing in the Lighthouse. "At the center of the room is the light itself, a 1000-Watt tungsten halogen light powered by diesel generator, and having a visible range of twenty-six nautical miles." Understand "lamp" as the light. It is lit. The light has a a direction called heading. The heading of the light is north.

A room is usually dark.

Understand "turn [something] [a direction]" as reorienting it to. Reorienting it to is an action applying to two things.

will give us

>turn light northeast
You must name something more substantial.

To avoid this mystifying result:

"Eddystone"

The Lighthouse is a room. "A lonely place, but in these tense times, no one but the lighthouse keeper and a few trusted agents are allowed on the grounds at all, for fear of sabotage."

The light is a fixed in place thing in the Lighthouse. "At the center of the room is the light itself, a 1000-Watt tungsten halogen light powered by diesel generator, and having a visible range of twenty-six nautical miles." Understand "lamp" as the light. It is lit. The light has a direction called heading. The heading of the light is north.

A room is usually dark.

Understand "turn [something] [a direction]" as reorienting it to. Reorienting it to is an action applying to one thing and one visible thing.

Instead of turning the light, say "Try turning the light to the direction of your choice."

Check reorienting it to: if the noun is not the light, say "You couldn't do so meaningfully." instead; if the second noun is up or the second noun is down, say "The light only points in compass directions." instead.

And now that's done, we have a little fun calculating where the beam hits:

Carry out reorienting it to:
now the heading of the light is the second noun;
let way be the heading of the light;

let place be the room way from the Lighthouse;
while place is a room and place is lower than Lighthouse:
 let place be the room way from the place;
 if place is not a room, now the beam is nowhere;
 otherwise move beam to the place.

Report reorienting it to: say "The light now points [heading of the light][if the beam is in a room], spotlighting [the holder of the beam][otherwise], into empty space[end if]."

The beam is a lit thing. Understand "light" or "brilliant" as the beam. "Brilliant light from the lighthouse floods the whole area." It is fixed in place. Instead of doing something other than examining to the beam: say "The light is, of course, intangible." The description is "The light is coming from the lighthouse, since the lamp is apparently pointed this way."

Altitude is a kind of value. 200 ft specifies an altitude. A room has an altitude. The altitude of a room is usually 50 ft. The altitude of the Lighthouse is 100 ft.

Definition: a room is low if its altitude is 20 ft or less.

The Jetty is south of the Lighthouse. "During daylight hours, a fine place to catch almost unlimited supplies of crayfish. Less entertaining by night."

North of the Lighthouse is the Uphill Road. The altitude of Uphill Road is 75 ft. North of Uphill Road is Hilltop. The altitude of Hilltop is 110 ft. The description of Hilltop is "The highest natural point around for miles; sometimes you will sit up here and watch for the lighthouse supply ship, the Lady Loch."

Northeast of the Lighthouse is Open Field. East of the Lighthouse is Stanley Creek Valley. The description of Stanley Creek is "This place used to have some other name meaning Ghost Valley in the aboriginal language, but it was piously renamed by missionaries." Train Trestle is east of the Stanley Creek Valley. "The now-abandoned track of the Bush Pacific Railway runs here, above Stanley Creek." The altitude of Train Trestle is 100 ft.

Before going from a room (called source) to a room (called destination):
 if source is lower than destination:
 say "It's an uphill climb...";
 otherwise:
 if destination is lower than source, say "You're heading downhill now...";
 otherwise say "It's a straight shot."

Test me with "turn lamp / turn lamp down / turn lamp east / e / e".

All very loosely based on the Eddystone Point lighthouse of Tasmania, built in 1889, and forbiddingly remote even today. George Isaacs, a child growing up in the lighthouse, remembers the plentiful crayfish.



A map of linked rooms works well for modeling enclosed or indoor space, and somewhat less well for modeling large open spaces, where a person should reasonably be able to see things which are much too far away to touch. With some modifications to scoping, though, we can create an environment where objects in nearby rooms are described and viewable, and where the player will automatically move towards distant items before interacting with them physically.

"Rock Garden"

Section 1 - General Rules

Intervisibility relates rooms to each other in groups. The verb to be connected with means the intervisibility relation.

Definition: a room is inter-visible if it is connected with more than one room.

After deciding the scope of the player when the location is an inter-visible room:
repeat with other place running through rooms which are connected with the location:
unless the other place is the location, place the other place in scope.

Rule for reaching inside a room (called target) which is connected with the location:

let way be the best route from the location to the target;
if the way is not a direction:
say "You can't get over to [the target] from here.";
deny access;
say "(first heading [way])[command clarification break]";
try going way;
if the player is in the target, allow access;
otherwise deny access.

After looking when the location is an inter-visible room:

repeat with other place running through rooms which are connected with the location:
if the other place is not the location, describe locale for other place.

Section 2 - The Scenario

Rock Garden West is west of Rock Garden East. Rock Garden East contains a rake. Rock Garden West contains a bench and a maple leaf. The bench is an enterable supporter.

Rock Garden West is connected with Rock Garden East.

Test me with "get rake / drop rake / sit on bench / get rake".



Suppose a game in which the player is wandering an open landscape with long vistas, allowing him to LOOK in some direction, or even look at an adjacent location.

"A View of Green Hills"

Corinth is a room. Athens is east of Corinth. Epidaurus is southeast of Corinth and east of Mycenae. Mycenae is south of Corinth. Olympia is west of Mycenae. Argos is south of Mycenae. Thebes is northwest of Athens. Pylos is south of Olympia. Sparta is east of Pylos and south of Argos. Delphi is northwest of Thebes.

Understand "look [direction]" as facing.

Facing is an action applying to one visible thing.

Carry out facing:

- let the viewed item be the room noun from the location;
- if the viewed item is not a room, say "You can't see anything promising that way." instead;
- try looking toward the viewed item.

In rules about action handling, "noun" refers to the first object that the player has mentioned in his command, so if the player typed >LOOK WEST, "let the viewed item be the room noun from the location" would be processed as "let the viewed item be the room west from the location", and so on.

We can at need override the default behavior, if it is not going to be appropriate for the player to see the next room over. There is only sky above at any time, so...

Instead of facing up:

- say "Above you is bright sky."

Understand "look toward [any adjacent room]" as looking toward. Understand "examine [any adjacent room]" as looking toward.

Looking toward is an action applying to one visible thing.

Carry out looking toward:

- say "You make out [the noun] that way."

This design allows us to create descriptions for rooms (as seen from the outside) which will work regardless of where we're looking from. For instance:

Instead of looking toward Athens:

- say "Even from here you can make out the silhouette of the Acropolis."

Test me with "look north / look south / look up / look east / east / look west".



Example **Stately Gardens**



An open landscape where the player can see landmarks in nearby areas, with somewhat more complex room descriptions than the previous example, and in which we also account for size differences between things seen at a distance.

This time we're going to assume that the player can see into any room that is on a line of sight within one or two steps of travel.

"Stately Gardens"

Chapter 1 - Laying Out Rooms

A room can be indoors or outdoors.

Use full-length room descriptions.

After deciding the scope of the player:

repeat with the way running through directions:

let first step be the room the way from the location;

if the first step is a room:

place the first step in scope;

let second step be the room the way from the first step;

if the second step is a room, place the second step in scope;

place the obelisk in scope.

The obelisk is so large that it can be seen from every room. If we had a number of such large monuments we might want to write a systematic routine to handle them, but this will do for now.

The room description heading rule is not listed in the carry out looking rules.

Now, we set things up so that the surrounding areas are described automatically as part of the room description:

Building description is a truth state that varies. Building description is false.

After looking when the location is an outdoors room:

now count of sentences is 0;

now building description is true;

repeat with way running through directions:

let space be the room way from the location;

if space is an outdoors room, silently try looking toward space;

if the obelisk is not in the location and the obelisk is unmentioned:

let the way be the best route from location to the Upper Terrace;

if the way is a direction, say "[The obelisk] is proudly visible on [the way]

horizon. [run paragraph on]";

increment the count of sentences;

now building description is false;

unless the count of sentences is 0:

say paragraph break.

But perhaps there are a few rooms where we do not wish that to happen, so we'll build in exceptions for those.

After looking in the rose garden:
say "Otherwise, you are quite cut off."

After looking in the Ha-ha:
do nothing.

And suppose we want to allow the player to look in any direction:

Understand "look [direction]" or "look to/toward [direction]" as facing.

Facing is an action applying to one visible thing.

Carry out facing:
let the viewed item be the room noun from the location;
if the viewed item is not a room:
if the location is indoors, say "Your view is restricted by the lack of doors or windows in that direction." instead;
otherwise say "You can't see anything promising that way." instead;
try looking toward the viewed item.

Instead of facing up:
say "Above you is bright sky."

We also need to tell distant rooms how to describe themselves.

Understand "look toward [any adjacent room]" as looking toward.

Looking toward is an action applying to one visible thing.

Check looking toward a room which does not contain something mentionable:
if building description is false:
say "You can't make out anything of interest that way." instead.

Carry out looking toward:
now every thing is unmentioned;
now the chosen direction is the best route from the location to the noun;
now the second noun is the room the chosen direction from the noun;
if the noun contains something mentionable:
repeat with item running through mentionable things in the noun:
carry out the writing a distant paragraph about activity with the item;
if the noun contains something mentionable:
increment the count of sentences;
choose row count of sentences in the Table of Distance Sentences;
if the second noun is an outdoors room and the second noun contains something mentionable, say "[both entry] [run paragraph on]";
otherwise say "[here entry] [run paragraph on]";
otherwise:
if the second noun is an outdoors room and the second noun contains something mentionable:
increment the count of sentences;
choose row count of sentences in the Table of Distance Sentences;

say "[there entry] [run paragraph on]";
 if building description is false:
 say paragraph break.

And again, some exception needs to be made for seeing what's in the dip in the ground:

Instead of looking toward the Ha-ha:
 now the chosen direction is the best route from the location to the noun;
 now the second noun is the room the chosen direction from the noun;
 if the second noun is an outdoors room and the second noun contains something mentionable:
 increment the count of sentences;
 choose row count of sentences in the Table of Distance Sentences;
 say "[there entry] [run paragraph on]".

The following is to account for cases where the player types "look toward obelisk" or similar, rather than looking toward a room:

Understand "look toward [something]" as examining.

The following is arguably an unnecessary refinement, but the listing of items in the distance gets a bit repetitive unless we vary the sentence structure.

Chosen direction is a direction that varies.

Count of sentences is a number that varies.

Table of Distance Sentences

both	here	there
"From here, you make out [a list of mentionable things in the noun] a little way [chosen direction], and, further on, [a list of mentionable things in the second noun]."	"From here, you make out [a list of mentionable things in the noun] [if the noun is not adjacent to the location]some distance [end if]to [the chosen direction]."	"From here, you make out [a list of mentionable things in the second noun] some distance [chosen direction]."
"To [the chosen direction] there [is-are a list of mentionable things in the noun], partly obscuring your further view of [a list of mentionable things in the second noun]."	"To [the chosen direction] there [is-are a list of mentionable things in the noun]."	"Quite a way [chosen direction] [is-are a list of mentionable things in the second noun]."
"Then [chosen direction] [is-are a list of mentionable things in the noun], and beyond [a list of mentionable things in the second noun]."	"Meanwhile, to [the chosen direction] [is-are a list of mentionable things in the noun]."	"Meanwhile, [chosen direction] in the middle distance [is-are a list of mentionable things in the second noun]."
"When you turn [chosen direction], you see [a list of mentionable things in the noun], and somewhat further on [a list of mentionable things in the second noun]."	"When you turn [chosen direction], you see [a list of mentionable things in the noun]."	"If you turn [chosen direction], you see [a list of mentionable things in the second noun] some way off."
"Somewhere generally [chosen direction] [is-are a list of mentionable things in the noun], beyond which, [a list of mentionable things in the second noun]."	"Roughly [chosen direction] [is-are a list of mentionable things in the noun]."	"Moreover, in the [chosen direction] distance [is-are a list of mentionable things in the second noun]."
"[The chosen direction] shows [a list of mentionable things in the noun] and then [a list of mentionable things in the second noun]."	"And to [the chosen direction] [a list of mentionable things in the noun]."	"Meanwhile, [chosen direction] in the middle distance [is-are a list of mentionable things in the second noun]."
"Then, [chosen direction], [is-are a list of mentionable things in the noun], and beyond [a list of mentionable things in the second noun]."	"Meanwhile, to [the chosen direction] [is-are a list of mentionable things in the noun]."	"Meanwhile, [chosen direction] in the middle distance [is-are a list of mentionable things in the second noun]."

"Finally, [chosen direction], [is-are a list of mentionable things in the noun], somewhat nearer than [a list of mentionable things in the second noun]."

"Finally, to [the chosen direction] [is-are nearer than [a list of mentionable things in the second noun]."

"Finally, [chosen direction] in the middle distance [is-are a list of mentionable things in the second noun]."

Now, our ability to view things at a distance should be determined by the size of the things we're trying to see:

Chapter 2 - Height

A height is a kind of value. 10 feet 11 inches specifies a height. 10 feet 11 specifies a height. The verb to stand means the height property. The verb to measure means the height property. A thing has a height. The height of a thing is usually 3 feet 0.

Definition: a thing is tiny if its height is 0 feet 6 inches or less.

Definition: a thing is short if its height is 3 feet 0 or less.

Definition: a thing is tall if its height is 6 feet 0 or more.

The height of a man is usually 5 feet 10 inches. The height of a woman is usually 5 feet 6 inches.

Definition: a thing is monumental if it is taller than 25 feet 0 inches.

Definition: a thing is mentionable if it stands tall enough to see.

To decide whether (item - a thing) stands tall enough to see:
 if the item is in the Rose Garden and the item is shorter than the roses, no;
 if the item is mentioned, no;
 if the item is in an adjacent room and item is taller than 2 feet 0, yes;
 if the item is taller than 4 feet 0, yes;
 no.

Instead of examining something which is within a room (called the space) which is not the location:

if the location is adjacent to the space:
 if the noun is tiny, say "It is too far from here for you to make out much detail about [the noun]." instead;
 let way be the best route from the location to the space;
 if the way is a direction, say "You gaze off [way] at [the noun]...";
 continue the action;
 otherwise:
 if the noun is short, say "It is too far from here for you to make out much detail about [the noun]." instead;
 let way be the best route from the location to the space;
 if the way is a direction, say "You gaze off [way] into the distance at [the noun]...";
 continue the action.

We might also want to be able to override, manually, the way distant things are described.

Writing a distant paragraph about something is an activity.

Rule for writing a distant paragraph about the lily pond:

if the second noun is a room and something mentionable is in the second noun, say "A [lily pond], [chosen direction], patchily reflects [a list of mentionable things in the second noun] on the far side. [run paragraph on]";

otherwise say "To [the chosen direction], [a lily pond] shimmers in the sunlight. [run paragraph on]"

Rule for writing a distant paragraph about the roses:

if something in the Rose Garden is taller than the roses,

say "Over the tops of [the roses], [chosen direction], you see [a list of mentionable things in the rose garden]. [run paragraph on]";

otherwise say "Immediately [chosen direction] is [the roses]. [run paragraph on]"

Rule for writing a distant paragraph about the obelisk:

if a mentionable thing in the Upper Terrace is shorter than the obelisk,

say "A stupidly grand [obelisk], [chosen direction], towers over [a list of mentionable things in the Upper Terrace]. [run paragraph on]";

otherwise say "To [the chosen direction], you can't help noticing [the obelisk], which is much larger than any object really needs to be. [run paragraph on]".

After writing a distant paragraph about something:

increment the count of sentences.

Moreover, proximate things might have special descriptions too.

Rule for writing a paragraph about something tiny when the location is outdoors:

if the location is the Gravel Circle,

say "Abandoned in the gravel [is-are a list of unmentioned tiny things in the location]. [run paragraph on]";

otherwise say "Half trampled into the grass, and easy to miss, [is-are a list of unmentioned tiny things in the location]. [run paragraph on]"

Before doing something other than examining or approaching to something which is not within the location:

if the player has the noun, continue the action;

say "(first going over to [the noun])[line break]";

try approaching the noun;

if the noun is not within the location, stop the action.

Understand "go toward/to/towards/near [something]" or "approach [something]" as approaching.

Approaching is an action applying to one visible thing.

Check approaching:

if the player is in something, say "You'll have to get up." instead;

if the noun is within the location, say "You're as close to [the noun] as you can get." instead;

let space be the location of the noun;

if the space is not a room, say "You don't quite see how to get there." instead;

let way be the best route from the location to the space;

unless way is a direction,

say "You can't see how to get over there from here." instead.

To head to (space - a room):
let the way be the best route from the location to the space;
if the space is adjacent to the location,
try going way;
otherwise silently try going way.

Carry out approaching:
let space be the location of the noun;
while the space is not the location:
head to space.

This is a bit primitive, since if we had an occasion where going was blocked, we could get stuck in a loop. So we would need to be careful, but for this example it won't arise.

Going state is a truth state that varies. Going state is false.

Check going:
now going state is true.

The description of a room is usually "[if going state is true]You drift [noun] across the open lawn[direction relative to obelisk]. [end if]An absolutely phenomenal quantity of manicured turf stretches from where you stand in almost every direction."

Before reading a command:
now going state is false.

To say direction relative to obelisk:
if obelisk is in the location:
say ", as though drawn magnetically to the foot of the monument";
otherwise:
let way be the best route from the location to the Upper Terrace;
if way is the noun, say ", drawn towards [the obelisk]";
if the way is the opposite of the noun, say ", keeping [the obelisk] more or less at your back".

Chapter 3 - The Grounds

When play begins:
now the left hand status line is "Idyllic";
now the right hand status line is " ".

The Gravel Circle, the Ha-ha, the Sheep Field, the Open Lawn, the Croquet Ground, the Rose Garden, the Upper Terrace, the Middle Terrace, and the Lower Terrace are outdoors.

The Middle Terrace is north of the Lower Terrace and south of the Upper Terrace. The lily pond is fixed in place in the Middle Terrace."You [if going state is true]come to[otherwise]are at[end if] the north edge of a perfectly round lily pond, bordered with stones. Its surface patchily reflects [the marble anteatery] on the south bank." A tent peg and a wilted orchid are in the Middle Terrace. The tent peg measures 0 feet 6. The orchid measures 0 feet 4.

The description of the Lower Terrace is "[if going state is true]You climb [noun] up a small hillock[direction relative to obelisk][otherwise]You stand on a short, round, entirely artificial hillock[end if]."

The marble anteater is a fixed in place thing in the Lower Terrace. The height of marble anteater is 6 feet 2 inches."A marble anteater stands on a pedestal at the top of the hill. In the bright sunlight the white marble makes a striking contrast with [the obelisk] in the distance." The description is "The anteater is very much more than life-size."

The obelisk of black granite is a fixed in place thing in the Upper Terrace."Now that you are at the foot of it, you can properly appreciate the stupid immensity of the obelisk, pointing stonily at heaven." The height of the obelisk is 50 feet 0 inches. The description of the obelisk is "It stands ridiculously tall, and has an inscription on the face."

The inscription is part of the obelisk. The height of the inscription is 0 feet 3 inches. The description of the inscription is "You can't read the squirming, pointed letters, but they make you uneasy."

The Gravel Circle is west of the Upper Terrace, northwest of the Middle Terrace, and north of the Croquet Ground. The description of the Gravel Circle is "[if going state is true]You head [noun] until the lawn thins and[otherwise]Here the lawn[end if] gives way to a circle of raked gravel, which crunches pleasingly beneath you."

Instead of going northwest in the Upper Terrace, try going north.

The Ha-ha is north of the Gravel Circle and northwest of the Upper Terrace. The description of the Ha-ha is "[if going state is true]The land dips here so suddenly that you do not know the dip is there until you're in it; but it prevents livestock from crossing barriers, and that is the important thing[otherwise]You are at the base of a steep-sided depression, so the lawn continues north and south more or less at the level of your head[end if]."

The tip of [the obelisk] is the only thing you can make out from this depression, off to the southeast.". North of the Ha-ha is the Sheep Field. In the Sheep Field is an animal called a black sheep. The black sheep stands 4 feet 3 inches."A black sheep grazes placidly nearby." The description of the black sheep is "It reminds you of your Uncle Tim."

Before going from the Ha-ha:

say "It's a bit of a scramble to get back up the side of the depression, and you keep slipping in the damp grass. But you manage at last."

The Rose Garden is southwest of the Lower Terrace. The thicket of red roses is a fixed in place thing in the Rose Garden. The thicket stands 4 feet 2 inches."Heavy red roses grow over a roughly horseshoe-shaped wall around you. Over this barrier, the head of [the marble anteater] is visible to the northwest, and the tip of [the obelisk] in the distance."

The description of the Rose Garden is "[if going state is true]You slip [noun] into the enclosure of the rose garden. [end if]The rest of the park, and the world, seems muted and quiet."

Instead of smelling the rose garden: try smelling the roses. Instead of smelling the roses, say "The smell tickles the back of your throat and makes you want to cough."

Instead of listening to the rose garden:
say "You can't hear anything at all."

The Open Lawn is north of the Rose Garden, west of the Lower Terrace, and southwest of the Middle Terrace. The Croquet Ground is north of the Open Lawn, west of the Middle Terrace, southwest of the Upper Terrace, and northwest of the Lower Terrace.

A discarded champagne cork is in the Open Lawn. It stands 0 feet 2 inches.

A stone bench is an enterable supporter in the Croquet Ground. It stands 3 feet 8 inches. "There is a stone bench here -- a sort of stone sofa, really, with nymphs disporting themselves on the arms and back." The description of the bench is "It used to be a Roman sarcophagus -- hence the nymphs -- but someone has thoughtfully recarved it as lawn furniture."

The half-size Bentley is a vehicle in the Gravel Circle. "A sort of child's-toy version of a Bentley is parked [if something parkable is in the location]beside [the tallest parkable thing in the location][otherwise]close at hand[end if]." The description of the half-size Bentley is "Of beautiful and unambiguously luxurious lines, but sized down to hold only one or (at a stretch) two people, and powered by electricity." The half-size Bentley stands 3 feet 6 inches.

Definition: a thing is parkable if it is not a person and it is not the Bentley.

Instead of touching the obelisk, say "Though it is black stone in sunlight, the obelisk is very cold to the touch."

Test me with "look east / look toward obelisk / s / s / e / sw / ne / n / n / w / n / n / examine obelisk / touch obelisk / read inscription".

45

★ Example Something Narsty

WI

A staircase always open and never openable.

In a game with many staircases, we might want:

A staircase is a kind of door. A staircase is always open. A staircase is never openable.

Defining the staircase this way means that we will never be able to get away with (for instance) a folding ladder into the attic which is sometimes closed up. So alternatively we might do

"Something Narsty"

A staircase is a kind of door. A staircase is usually open. A staircase is seldom openable.

We could then write a rule so that whenever the player types CLIMB [any staircase], the command is diverted to an enter command, while all other doors refuse to respond to such treatment. Still, this kind is now usable, as we see in this trivial example:

The ladder is a staircase. It is above the Woodshed and below the Scary Loft.

Test me with "up".

But these are refinements for a later chapter.

63

★ **Example When?**

WI

A door whose description says "...leads east" in one place and "...leads west" in the other.

Very simple, but quite frequently useful:

"When?"

The temporal vortex is an open door. It is west of Yesterday and east of Today.
"A whirling temporal vortex leads [if the player is in
Yesterday]west[otherwise]east[end if]."

89

★ **Example Hayseed**

WI

A refinement of our staircase kind which can be climbed.

Presumably all staircase-type connections between rooms should respond when the player says CLIMB STAIRS (or the equivalent). So

"Hayseed"

A staircase is a kind of door. A staircase is usually open. A staircase is seldom openable.

The ladder is a staircase. It is above the Barn and below the Hayloft.

Instead of climbing a staircase:
try entering the noun.

Test me with "climb ladder / g".

Attempts to climb other types of door will still be treated as useless.

★ Example Higher Calling

All doors in the game automatically attempt to open if the player approaches them when they are closed.

"Called" is quite useful in the context of rules about going, since go rules often refer to things other than the noun the player typed. For instance, suppose we want to have doors which politely open when the player tries to pass through them:

"Higher Calling"

Before going through a closed door (called the blocking door):

say "(first opening [the blocking door])[line break]";
silently try opening the blocking door;
if the blocking door is closed, stop the action.

Dome is a room. North of Dome is North Chapel. South of the Dome is South Chapel. West of the Dome is Western End. Quiet Corner is northwest of the Dome, north of Western End, and west of North Chapel. Loud Corner is east of North Chapel, northeast of Dome, and north of Eastern End. Eastern End is north of Dim Corner and east of Dome. Dim Corner is southeast of Dome and east of South Chapel. Ruined Corner is southwest of Dome, west of South Chapel, and south of Western End.

The church door is east of Eastern End and west of the Courtyard. The church door is a door.

Test me with "e / e".

A fuller set of such rules is included in the Locksmith extension.

★ Example Wainwright Acts

A technical note about checking the location of door objects when characters other than the player are interacting with them.

Suppose we wanted to write rules for a character who will interact with doors in other locations even when the player is not present. This poses a little challenge: doors are actually single objects, and -- with the same shuffling of stage properties that applies to backdrops -- they are moved as needed to represent the door object in whatever room contains the player.

That means that it isn't safe to rely on a phrase like

if an open door is in the location of Bernard

because, even if Bernard's location is connected by doors to other places, the actual representation of that door may not be "in" Bernard's location, from the model's point

of view, at this exact moment.

This does not, of course, mean that we can't ask this question; just that we have to be a little cleverer about how we phrase it. Every door has properties that correspond to the two locations

linked:

the front side of the blue door (a room, which is arbitrarily one side of the door)
the back side of the blue door (arbitrarily the other side)

We can make this information easier to check with a conditional relation, like so:

Liminality relates a door (called X) to a room (called Y) when the front side of X is Y or the back side of X is Y. The verb to be a threshold of means the liminality relation.

And this allows us to write rules that have characters interacting with doors even in the player's absence:

"Wainwright Acts"

The Waiting Room is a room. The waiting room door is west of the Waiting Room and east of the Gents' Loo. The Waiting Room door is an open door. "The waiting room door [if open]stands open[otherwise]is shut firmly[end if]."

Sir Humphrey is a man in the Gents' Loo.

Liminality relates a door (called X) to a room (called Y) when the front side of X is Y or the back side of X is Y. The verb to be a threshold of means the liminality relation.

Definition: a person is other if he is not the player.

Every turn:

repeat with indiscreet one running through other people:
repeat with port running through open doors that are a threshold of the location of the indiscreet one:
if the port is a threshold of the location and the indiscreet one is not in the location:

say "Through [the port], you overhear [the indiscreet one] discussing [one of]his hopes for your imminent resignation[or]your wife's infidelity[or]your financially straitened circumstances[or]ways to avoid attending your birthday party[or]your halitosis[as decreasingly likely outcomes]."

Test me with "z / z / z / w / z / e / close door / z".

Here we expand on the simple examples *When?* and *Whence?*; this time we want the player to be able to refer to doors by their directions, as in "the west door" when the door in question does in fact lead west.

"Whither?"

The temporal vortex is an open door. It is west of Yesterday and east of Today.

The initial appearance of a door is usually "Nearby [an item described] leads [if the other side of the item described is visited][direction of the item described from the location] to [the other side][otherwise][direction of the item described from the location][end if]."

Direction-relevance relates a door (called X) to a direction (called Y) when the direction of X from the location is Y. The verb to be directionally-relevant to means the direction-relevance relation.

Understand "[something related by direction-relevance] door" as a door.

As an added touch, we respond also to the case where the player postulates a door in some direction when there is no such thing at the moment:

Rule for printing a parser error when the player's command includes "[non-door direction] door":

say "There is no door in that direction." instead.

Definition: a direction (called direction D) is non-door:

let the target be the room-or-door direction D from the location;
if the target is a door:
no;
yes;

Test me with "examine west door / x east door / w / x w door / x e door / tie me to the west door / tie the west door to me / push the west door east / push the east door west".

24



Example Neighborhood Watch

WI

A locked door that can be locked or unlocked without a key from one side, but not from the other.

Suppose we want a locked door that can be opened with a key, but is also openable by hand without a key from one side only. We start by defining an ordinary lockable door and the key that controls it:

"Neighborhood Watch"

The shabby door is a door. It is outside from the Studio Apartment and inside from the Rickety Stairwell. The shabby door is locked.

The brass key is carried by the player. It unlocks the shabby door.

The next part is going to require that we modify the normal operation of the "lock" command. "Lock" ordinarily requires that the player supply two objects: a thing he wants to unlock, and the key he wants to use on it. The full command is LOCK DOOR WITH THE KEY, and Inform will not accept simply LOCK DOOR as locking.

Therefore, we're going to need to create our own new variant on the lock verb (and the unlock verb, while we're at it). The full procedure for this is laid out in the chapters on Action and Understanding, but here is an example:

Understand "lock [something]" as locking keylessly. Locking keylessly is an action applying to one thing.

Here we've created a new action -- locking something without a key -- and we've told Inform to understand LOCK DOOR as this action, rather than an incomplete command to LOCK DOOR WITH SOMETHING.

Now we add some instructions so that the game will not let us use this keyless unlocking command unless we're in the right place or are properly equipped:

Check locking keylessly:
if the noun is not a door, say "[The noun] is not something you can lock."
instead;
if the noun is locked, say "[The noun] is already locked." instead;
if the player carries the brass key and the player is in the Stairwell, try locking
the noun with the brass key instead;
if the player is in the Stairwell, say "You can't lock the door from this side
without the key." instead.

This check rule is performed before the keyless locking action succeeds. The first thing we do is try to use the key if the player is outside and has the key: this way, LOCK DOOR will turn automatically into LOCK DOOR WITH THE KEY, under circumstances where that is both possible and necessary.

The second thing is to check whether the player is outside but keyless, and, if so stop the action from being performed successfully. Here we print a failure message followed by the word "instead", which tells Inform that we've substituted some other outcome for the usual performance of the action.

Now we're reasonably sure that the player is only locking keylessly in the case that he is inside the Studio. (We might have to do a more thorough check for this if there were more than two rooms, but as it is, the player can only be in the Stairwell or in the Studio, so if we have ruled out the Stairwell, we are safe.) So now we want to add what happens when locking-without-a-key command succeeds:

Carry out locking keylessly:
now the noun is locked.

That's it. We've just told Inform to make the door be locked. "Now..." syntax will be explained more thoroughly in the chapter on change. But we still haven't described to

the player what just happened, so let's provide a description of that, too:

Report locking keylessly:
say "You flip over the deadbolt to lock [the noun]."

And now we have to do a similar set of things for unlocking:

Understand "unlock [something]" as unlocking keylessly. Unlocking keylessly is an action applying to one thing.

Check unlocking keylessly:
if the noun is not a door, say "[The noun] is not something you can lock."
instead;
if the noun is unlocked, say "[The noun] is already unlocked." instead;
if the player carries the brass key and the player is in the Stairwell, try
unlocking the noun with the brass key instead;
if the player is in the Stairwell, say "You can't unlock the door from this side
without the key." instead.

Carry out unlocking keylessly:
now the noun is unlocked.

Report unlocking keylessly:
say "You flip over the deadbolt to unlock [the noun]."

Test me with "unlock door / drop key / open door / out / close door / lock door /
open door / in / get key / out / close door / lock door / unlock door".

Some (but not all) of this work is done for you if you like by the Locksmith extension. If you prefer, you can include that extension, then follow the documentation in order to implement the remainder of the scenario. Locksmith takes care of implementing the additional locking and unlocking actions, and provides some other conveniences.



Example One Short Plank

A plank bridge which breaks if the player is carrying something when he goes across it. Pushing anything over the bridge is forbidden outright.

"One Short Plank"

The East Jungle is a room. The plank bridge is west of the East Jungle and east of the West Jungle. The plank is an open unopenable door. "A precarious plank bridge extends [if the location is West Jungle]east[otherwise]west[end if] across the chasm." The description of the plank is "Extremely fragile and precarious."

Instead of going through the plank when the player is carrying something:
say "You step gingerly across the plank, which bows under your weight. But your meagre possessions are the straw which breaks the camel's back!";
end the story.

After going through the plank:

say "You step gingerly across the plank, grateful that you're not burdened.";
continue the action.

There is a feather in the East Jungle.

But indeed, why stop there?

The gigantic stone ball is a thing in the West Jungle. It is pushable between rooms.

Before going through the plank with something:

say "Surely you jest." instead.

Test me with "w / e / w / push ball e / e / get feather / w".

151



Example Elsie

WI

A door that closes automatically one turn after the player opens it.

Suppose we want to create an automated door of the sort that closes when it isn't in use. A convenient way is to write a rule that fires "every turn when the sliding door was open". This will be true only if the door was open at the beginning of the turn: if the player just opened it this turn, it stays open, and if it was already closed, it stays closed. Thus:

"Elsie"

The axis-ward is a direction. The opposite of axis-ward is hub-ward. Understand "aw" or "axisward" as axis-ward.

The hub-ward is a direction. The opposite of hub-ward is axis-ward. Understand "hw" or "hubward" as hub-ward.

The Ship's Bridge is a room.

The sliding door is a door. It is axis-ward from Bridge and hub-ward from C Deck. The initial appearance is "There is a door in this wall[if closed] -- or at least, the potential for a door, since currently it is sealed, distinguishable from the rest of the wall only by the warning stripes on its surface[end if]."

Every turn when the sliding door was open:

now the sliding door is closed;

if the player can see the sliding door:

say "The sliding door slips back into place, sealing the wall as though there had never been a breach."

After opening the sliding door:

say "You press the appropriate buttons, and a section of wall slides away."

Test me with "open door / look / enter door / z".

**Example Garibaldi 1**

Providing a security readout device by which the player can check on the status of all doors in the game.

Suppose we would like to allow the player to view the status of all the doors functioning in the game; and we want to identify those doors by mentioning which two rooms they connect. The following uses some techniques that will be covered in later chapters, but the basic idea may be obvious:

"Garibaldi"

The security readout is a device. The description of the readout is "The screen is blank."

Instead of examining the switched on security readout:

```
say "The screen reads: [fixed letter spacing]";
say line break;
repeat with item running through doors:
  say line break;
  say " [item] ([front side of the item]/[back side of the item]): [if the item is
locked]LOCKED[otherwise]UNLOCKED[end if]";
say variable letter spacing;
say paragraph break.
```

It is more or less arbitrary which room winds up as the "front side" and which as the "back", but in this case it hardly matters.

The player carries the security readout.

The Docking Bay is a room. The inner airlock is a door. It is north of the Docking Bay and south of the Zocalo. The inner airlock is lockable and unlocked. The outer airlock is lockable and locked. It is a door. It is south of the Docking Bay and north of Space.

The quarantine seal is a door. It is west of the Zocalo and east of Medlab. Quarantine seal is locked.

The security pass unlocks the inner airlock. The player carries the security pass.

Test me with "x readout / turn on readout / x readout / lock inner airlock with security pass / x readout".

**Example Whence?**

A kind of door that always automatically describes the direction it opens and what lies on the far side (if that other room has been visited).

It would be fairly tedious reading to have a large game full of doors that describe themselves this way. Nonetheless, if we insisted we could use our knowledge of the map as leverage to make every door in the game describe itself automatically.

To do this, we make use of the phrase "direction of (the door) from (a room)" -- in this case, the direction of the door we're looking at when viewed from the player's location. Thus:

"Whence?"

The temporal vortex is an open door. It is west of Yesterday and east of Today.

The initial appearance of a door is usually "Nearby [an item described] leads [if the other side of the item described is visited][direction of the item described from the location] to [the other side][otherwise][direction of the item described from the location][end if]."

Test me with "w / e".

Yet a further variation on this, which can automatically understand "the east door" and "the west door" when appropriate, may be found in the example "Whither?".

116

★ Example Vitrine

WI

An electrochromic window that becomes transparent or opaque depending on whether it is currently turned on.

"Vitrine"

Plaza View is a room. "Your uncle's apartment, on loan to you for viewing the parade and celebrations today. This would be more of a sacrifice on his part if he weren't currently yachting around Corfu."

The smart window is a device in Plaza View. It is fixed in place. "A vast smart window [if transparent]overlooks the park[otherwise]has turned to a sheet of hazy blue[end if]." The smart window can be transparent. The smart window is transparent. The description is "An electrochromic device which changes shade and transparency in response to the application of current."

Curtains are so last year."

Note the "can be transparent" line. Devices ordinarily are not allowed to have transparency or opaqueness, but we can make an exception in this case. Without that line, attempts to change the transparency of the window will fail.

Carry out switching off the window: now the window is transparent.

Carry out switching on the window: now the window is opaque.

Instead of searching a transparent window: say "Isn't it lovely out there?"

Instead of searching an opaque window: say "The window is currently darkened."

Test me with "look through window / switch window / look through window / look".

21



Example Escape

WI

Window that can be climbed through or looked through.

Suppose we want to offer the player a window he can climb through, instead of a boring ordinary door. Our window will be like a door in that it connects two rooms, appears in both places, and impedes movement when it is shut. But we also want to add that we can look through it and see what lies on the other side; and we further want to understand "climb through window" or "jump through window" as attempts to pass through it.

We'll start by defining a couple of rooms and making the window a door between them.

"Escape"

Your Bedroom is a room. The bedroom window is a door. It is west of Your Bedroom and east of the Grassy Slope.

Now we have a "bedroom window" object which can be entered. Now, to catch the case where the player types "LOOK THROUGH WINDOW":

Instead of searching the window:

say "Through the window, you make out [the other side of the window]."

The other side of a door is always defined to be the room that we are not currently in when doing the check. When we are in the bedroom, the other side will be the grassy slope, and vice versa. "Searching" is the action that occurs when the player attempts to LOOK THROUGH something. (To review what grammar gives rise to what actions, we can always consult the Actions portion of the Index.)

Next we want to cover the case where we climb through the window:

Instead of climbing the window:

try entering the window.

And because "climb window" is understood but "climb THROUGH window" is not, we will have to borrow from the chapter on Understanding to add some new vocabulary to the game (and we'll add Jump too, while we're at it):

Understand "climb through [something]" as climbing. Understand "jump through [something]" as climbing.

Now the final piece: Inform will already keep the player from going through a closed window, but it will say "You can't, since the bedroom window is in the way." This is probably not ideal, so we can replace the instruction thus:

```
Instead of going through the closed window:  
  say "The window is shut: you'd break the glass."
```

```
Test me with "look through window / climb through window / open window / climb  
through window / look through window / close window / e / open window / e".
```

217

★★ Example Dinner is Served

WI

A window between two locations. When the window is open, the player can reach through into the other location; when it isn't, access is barred.

"Dinner is Served"

Street in Kolonaki is a room. "There is a single round table out on the street here, and a window more or less at knee level looks down into the Olive Tree Gyro Shop, which is partly basement."

The Street contains a round table. The table is scenery. On the round table is a plate. On the plate are a gyro and a mound of fresh potates. The plate is portable. The potates and the gyro are edible. The description of potates is "They'd be called french fries, at home, but these are steak-cut and fried in olive oil." The description of the gyro is "Dripping garlic-yogurt sauce."

Olive Tree Gyro Shop is inside from Street in Kolonaki. Kostis is a man in the Gyro Shop. In the Shop is a stand. On the stand is a rotating column of cooking lamb flesh. In the shop is a closed, openable container called a drinks refrigerator. The refrigerator contains a can of Mythos beer and a can of Coke Light.

Here's the part that allows reaching through the window.

We replace the usual rule that says the player can never reach into a room with one that more specifically checks whether we are trying to reach through the window. If we aren't, we return the usual refusal. If we are, we return a custom refusal if the window is closed ("You can't reach through the closed window"), but allow access if the window is open.

The can't reach through closed window rule is listed instead of the can't reach inside rooms rule in the reaching inside rules.

```
This is the can't reach through closed window rule:  
  let reaching through the window be false;  
  if the container in question is a room and the container in question is not the  
location:  
  if the container in question is the Street and the location is the Olive Tree  
Gyro Shop:  
    now reaching through the window is true;
```

```
if the container in question is the Gyro Shop and the location is the Street:
    now reaching through the window is true;
if reaching through the window is true:
    if the window is closed:
        say "You can't reach through the closed window.";
        deny access;
    otherwise:
        allow access;
otherwise:
    say "You can't reach into [the container in question] from here.";
    deny access.
```

And the rest is window-dressing.

After looking when a room (called the next room) is adjacent:
try examining the next room.

Instead of examining a supporter, say "On [the noun] [is-are a list of things on the noun]." Instead of examining an open container, say "In [the noun] [is-are a list of things in the noun]."

The window is a backdrop. It is in the Street and the Shop. The window can be openable. The window can be open. The window is openable and closed. Instead of searching the window in the Street: try examining the shop. Instead of searching the window in the Shop: try examining the street.

Understand "examine [any adjacent room]" as examining.

Instead of examining a room:
say "Over in [the noun], you can see [a list of visible things in the noun]."

After deciding the scope of the player:
if the player is in the Street, place the Shop in scope;
if the player is in the Shop, place the Street in scope.

Test me with "examine shop / open refrigerator / open window / examine shop / open refrigerator / get beer / in / examine street / out / get gyro / close window / put gyro in refrigerator / open window / put gyro in refrigerator".

272



Example Port Royal 4

WI

A cell window through which the player can see people who were in Port Royal in the current year of game-time.

Our protagonist is imprisoned in Port Royal, waiting out his years, and sometimes through the window of his cell he is able to see someone.

We are, however, obsessive about historical accuracy, so we provide a table of people who really lived in the city, together with the year in which their existence is attested. We want these people to appear in the description only in the year when they are known to have been present. (After all, mortality was high in Port Royal and new

people were constantly arriving, so someone's presence one year is no guarantee of their continued existence the next.)

"Waiting for Godot, Chyrurgeon"

New New Prison is a room. "You have a not very interesting room. Through the window you see passing [current denizen]."

Instead of waiting:

increment the current year;
say "It is now the year [the current year].";
try looking.

When play begins: now the right hand status line is "[current year]".

Every turn:

if the current year is 1692:
say "It turns out you have remained imprisoned until the great earthquake of 1692! Oops.";
end the story.

Current year is a number that varies. The current year is 1664.

To say current denizen:

repeat through the Table of Occupations and People:
if the date attested entry is the current year:
say "[nickname entry] [family entry], [trade entry]";
blank out the whole row;
rule succeeds;
say "absolutely no one".

It is possible to look up a row corresponding to, say, a specific year value using "listed in", but repeat through is convenient here because we know that we will never wind up trying to print entries when no row can be successfully selected.

Table of Occupations and People

Trade	nickname	family	Date attested
"architect"	"Robert"	"Snead"	1684
"baker"	"William"	"Wingar"	1683
"barber"	"William"	"Alcocke"	1676
"blacksmith"	"William"	"Davidson"	1679
"bricklayer"	"Samuel"	"Richardson"	1683
"butcher"	"John"	"Dennis"	1676
"carpenter"	"John"	"Albert"	1675
"cabinet-maker"	"Robert"	"Avis"	1666
"joiner"	"Peter"	"Bartaboa"	1666
"chandler"	"William"	"Bates"	1674
"chyrurgeon"	"William"	"Axtell"	1674
"chyrurgeon"	"Thomas"	"Trapham"	1678
"combmaker"	"Paul"	"Bennett"	1673
"cooper"	"James"	"Hall"	1676
"cooper"	"Henry"	"Pullein"	1675
"cordwainer"	"George"	"Barnard"	1675
"cordwainer"	"Edward"	"Skannon"	1680
"cordwainer"	"John"	"Wilmott"	1675
"drugster"	"William"	"Mathews"	1682
"fisherman"	"Richard"	"Collingwood"	1674
"glazier"	"Thomas"	"Hudson"	1684

"goldsmith"	"Richard"	"Lord"	1677
"gunsmith"	"Stephen"	"Massey"	1664
"hatmaker"	"John"	"Rosewell"	1683
"ivory turner"	"William"	"Clifton"	1691
"labourer"	"John"	"Dennis"	1674
"limeburner"	"John"	"Hardwick"	1675
"mariner"	"Alexander"	"Bailing"	1680
"mariner"	"Thomas"	"Bowtell"	1675
"mariner"	"Peter"	"Claiton"	1675
"mariner"	"Joseph"	"Cupid"	1672
"mariner"	"Michael"	"Dunn"	1675
"mason"	"John"	"Stone"	1673
"merchant"	"John"	"Agard"	1680
"merchant"	"David Lopez"	"Narbona"	1674
"merchant"	"Abraham"	"Langford"	1675
"merchant"	"John"	"Sweeting"	1675
"merchant"	"Charles"	"Knight"	1680
"merchant"	"Cornelius"	"Vandananker"	1670
"merchant"	"Moses Jesurum"	"Cordova"	1675
"pewterer"	"Simon"	"Benning"	1667
"pipemaker"	"John"	"Pope"	1680
"porter"	"George"	"Paul"	1670
"poulterer"	"Richard"	"Jeffreys"	1677
"sailmaker"	"Adam"	"Brewer"	1671
"schoolmaster"	"Peter"	"Bird"	1677
"shipwright"	"William"	"Cavell"	1676
"tailor"	"William"	"Case"	1676
"tailor"	"Pewter"	"Ebden"	1683
"waterman"	"William"	"Brocke"	1674
"waterman"	"Joel"	"Clements"	1668
"wherryman"	"John"	"Grant"	1669
"victualler"	"Barnaby"	"Adams"	1675
"vintner"	"Gabriel"	"Adkins"	1668
"tavern-keeper"	"John"	"Baldwin"	1670
"tavern-keeper"	"Mary"	"Dayton"	1664
"tavern-keeper"	"James"	"Turpin"	1679
"tavern-keeper"	"Christopher"	"Mayham"	1664

Test me with "wait / wait / wait".



Example A Haughty Spirit

Windows overlooking lower spaces which will prevent the player from climbing through if the lower space is too far below.

Suppose we have a game in which the player can climb through windows which overlook rooms below. We want him to be allowed to climb out windows to reach a room on the same level or at most one level lower than the one he's on; otherwise, he should get a refusal, saying that he would break his neck.

To figure out the height distance between the start room and the destination room, we might have a repeat loop look at all the directions one has to follow along the "best route" path between the two rooms, and record any ups and downs; then subtract the number of "up" steps from the number of "down" steps, and report what remains.

"A Haughty Spirit"

To decide what number is the distance (first place - a room) rises above (second place - a room):

let the total distance be the number of moves from the first place to the

```

second place;
  if the total distance is less than 1, decide on 0;
  let count of down moves be 0;
  let count of up moves be 0;
  let next place be the first place;
  repeat with counter running from 1 to the total distance:
    let the way be the best route from the next place to the second place;
    if the way is down, let count of down moves be the count of down moves
plus 1;
    if the way is up, let the count of up moves be the count of up moves plus 1;
    let next place be the room the way from next place;
  let the decision be the count of down moves minus the count of up moves;
  decide on the decision.

```

Now we just have to create windows and some action rules for interacting with them...

A window is a kind of thing. A window is always fixed in place. A window can be open or closed. A window is usually closed. A window can be openable or unopenable. A window is usually openable.

Understand "climb through [something]" as entering. Understand "jump through/out [something]" as entering.

Before entering a closed window:
 say "[The noun] would have to be opened first." instead.

Instead of entering a window:
 if the noun overlooks a room (called the far side):
 let fall be the distance the location rises above the far side;
 if fall is greater than 1, say "You'd break your neck." instead;
 say "You tumble into [the far side].";
 move the player to the far side;
 otherwise:
 say "There's nowhere to go."

Instead of examining a window:
 say "[The noun] [if the noun is open]opens over[otherwise]gives a view of[end if] [the list of rooms overlooked by the noun]."

Here we must anticipate a little from the chapter on Relations, and provide ourselves with a way of keeping track of how windows and rooms relate to one another:

Overlooking relates various windows to various rooms. The verb to overlook means the overlooking relation. The initial appearance of a window is usually "
 [The item described] overlooks [the list of rooms overlooked by the item described]."

The Square Keep is above the Winding Staircase. The Winding Staircase is above the Motte. A crown and a broken sword are in the Motte. The Bailey is west of the Motte.

The long window is in the Keep. The long window overlooks the Bailey and the Motte. The narrow window is in the Winding Staircase. The narrow window overlooks the Bailey.

Test me with "jump through window / open window / jump through window / d / x narrow window / open window / climb through window / e / up / down".

We could then add rules to allow the player to look through windows and see things in the rooms below, but that would require more material from later chapters.

194

★ Example The Dark Ages Revisited

WI

An electric light kind of device which becomes lit when switched on and dark when switched off.

This will be explored more in subsequent examples, but one of the things we can do with carry out rules is extend the function of existing commands so that they do more, or have special effects in specific situations. For instance, suppose we want to have a class of electric light:

"The Dark Ages Revisited"

An electric light is a kind of device. Carry out switching on an electric light: now the noun is lit. Carry out switching off an electric light: now the noun is unlit.

This will not affect the behavior of any other devices when switched; it will also not change the way in which switching lights on and off is reported. The player will still see "You switch the sodium lamp on." or the like. In this case that is probably what we want. If we wanted a special way of describing turning on electric lights as opposed to all other devices, we could also add an after rule for the electric light class. Adding this rule to the carry out train does guarantee, though, that in no case will we manage to make the lamp lit without actually making it switched on (or vice versa).

The Stopped Corridor is a room. "A low, square-cut corridor, running north to south, stooping you over."

The sodium lamp is an electric light in the Stopped Corridor. "[if switched on]The sodium lamp squats on the ground, burning away.[otherwise]The sodium lamp squats heavily on the ground.[end if]". The description is "It is a heavy-duty archaeologist's lamp, [if switched off]currently off.[otherwise]blazing with brilliant yellow light.[end if]"

Instead of burning the sodium lamp, try switching on the lamp.

So far so easy. Since we've built the description of its light or darkness into the lamp's description, though, we may want to get rid of the "...is switched on" line that automatically follows when we look at something. For this we do need to borrow from a later chapter:

The examine devices rule is not listed in the carry out examining rules.

Test me with "x lamp / switch lamp on / look / x lamp".

Example Hymenaeus

Understanding "flaming torch" and "extinguished torch" to refer to torches when lit and unlit.

"Hymenaeus"

A torch is kind of thing. Understand the lit property as describing a torch. Understand "lighted" or "flaming" or "burning" as lit. Understand "extinguished" as unlit. A torch is usually lit.

Before printing the name of a lit torch, say "flaming ".
Before printing the name of an unlit torch, say "extinguished ".

The Wedding Procession is a room.

Orpheus is a man in the Wedding Procession. Orpheus carries a torch.
Eurydice is a woman in the Wedding Procession. Eurydice carries a torch.

Rule for writing a paragraph about someone (called target):
say "[The target] carries [a list of things carried by the target]."

Every turn:

if a random chance of 1 in 2 succeeds and a torch is lit:
let target torch be a random lit torch;
now the target torch is unlit;
say "Aquilo blows down from the north, extinguishing the torch carried by [the holder of the target torch]."

Instead of examining a lit torch:

say "It casts a bright glow over [the holder of the noun]."

Instead of examining an unlit torch:

say "[The holder of the noun] is looking at it disconsolately, obviously worried about the omens."

Test me with "z / z / z / look / x flaming torch / x extinguished torch".

Example Reflections

Emphasizing the reflective quality of shiny objects whenever they are described in the presence of the torch.

"Reflections"

Behind the Waterfall is a room. "Though one wall of the cave is open to the waterfall, the quantity of water is so great that barely any light comes through from the outside." Behind the Waterfall is dark.

Surface is a kind of value. The surfaces are shiny and dull. A thing has a surface. A thing is usually dull.

The player carries a reflecting ball, a canopic jar, an abacus, a plumbline, a piece of chalk, and a torch. The reflecting ball is shiny.

Aladdin's lamp is a shiny thing in Behind the Waterfall.

Brightness is a kind of value. The brightnesses are guttering, weak, radiant and blazing. The torch has a brightness. The torch is blazing. The torch is lit.

Understand "blow out [something]" or "blow [something]" or "extinguish [something]" as blowing out. Blowing out is an action applying to one thing.

Carry out blowing out: say "Futile."

Instead of blowing out the torch:

now brightness of torch is the brightness before the brightness of the torch;
say "The light of the torch dies to [brightness of torch]."

Instead of blowing out the guttering torch:

say "Fool! Do you want to put it out entirely?"

Rule for writing a paragraph about a shiny thing:

say "The [brightness of the torch] light of [the torch] reflects in the surface[if the number of shiny things in the location > 1][end if] of [the list of shiny things in the location]."

Before printing the name of the torch while writing a paragraph about something:
if the torch is in the location, say "fallen ".

Test me with "drop ball / look / blow torch / look / drop torch / look".

363

★ Example Peeled

WI

Two different approaches to adjusting what the player can interact with, compared.

Suppose we have a situation where the player is in darkness, but is allowed to feel and interact with (except for examining) any large objects. In that case, we write a scope rule that puts those large objects into scope all the time, and trust the "requires light" aspect of verbs like examining to prevent the player from doing any actions that he shouldn't:

"Peeled"

A thing can be large or small.

Before touching a large thing when in darkness: say "You grope for [the noun]..."

After deciding the scope of the player:
repeat with item running through large things in the location:
place item in scope.

Some generic surroundings are backdrop. They are everywhere. Understand "walls" or "wall" or "ceiling" or "ground" or "floor" or "area" or "room" or "here" as the generic surroundings. Instead of touching the generic surroundings: say "You encounter nothing extraordinary." Instead of touching the generic surroundings when in darkness: say "You try feeling your way around and reach [a list of large things in the location]." After deciding the scope of the player when in darkness: place the surroundings in scope.

The Room of Mystery is a dark room. The bearskin rug is a large thing in the Room of Mystery. Instead of touching the rug: say "It feels furry!"

The peeled grape is a small thing in the Room of Mystery. Instead of touching the peeled grape: say "Gosh, is that an eyeball?"

Test me with "feel floor / feel rug / eat rug / examine rug / get grape".

Sadly, because the grape is small, the player will never encounter this horror.

Alternatively, suppose we have a situation in which the player can use one command to interact with a kind of thing that isn't normally in scope. It's usually most convenient to write the "understand" rule appropriately rather than use the scope activity.

(Note that we define "inquiring about" as applying to one **visible** thing; otherwise we would be required to be able to touch the catsuit in order to inquire about it. More on this restriction may be found in the Advanced Actions chapter on the topic of visible, touchable, and carried things.)

"Peeled"

Mr Steed's Flat is a room.

Understand "ask about [any subject]" as inquiring about. A subject is a kind of thing. The skintight catsuit is a subject. Inquiring about is an action applying to one visible thing.

Carry out inquiring about something:
say "'What can you tell me about [the noun]?' you demand. Mr Steed raises his eyebrows, but does not reply."

Test me with "ask about catsuit / x catsuit".

All this said, there do arise certain complex situations when we want an activity-specific scoping.



Suppose we want to have a room with a light switch. Turning the switch off makes the room go dark; turning it on restores the light. This kind of switch is an obvious candidate as a device.

"Down Below"

Terrifying Basement is a room. The light switch is a switched on device in the Terrifying Basement. It is fixed in place.

Here we define our light switch, and we also make it start out as "switched on". The Terrifying Basement will also start out lit (as all rooms do, by default, unless we specifically say that they are dark). We further say that it is fixed in place to avoid the ludicrous possibility of the player picking it up and carrying it away.

Next we add some instructions to control how turning the light switch on and off affects the room light. These borrow from later chapters on actions, but the gist may be obvious anyway:

Carry out switching off the light switch: now the Terrifying Basement is dark.

Carry out switching on the light switch: now the Terrifying Basement is lighted.

Inform already has the idea of light and darkness built in; we will see more about this later, and the Phrasebook (in the Index tab) also contains a list of all the adjectives (lighted, dark, etc) which are important to use here.

Speaking of the Index, the Actions tab contains a list of all the grammar that can be used to activate a given command: for instance, the switching action responds to "switch [something]" or "turn on [something]". In this case, we may want to give the player an extra option or two. It would be pretty natural for a player to try >FLIP SWITCH, so let's add that in:

Understand "flip [something switched off]" as switching on. Understand "flip [something switched on]" as switching off. Understand "flip [something]" as switching on.

The nuances of this will be explored in the chapter on Understanding. What is useful to know here is that we have taught Inform to understand that >FLIP LIGHT SWITCH means to turn it on when the switch is already off; if the switch is already on, FLIP SWITCH means to turn the switch off. Depending on the kind of device we are modeling (button? lever? dial?), we might want to write similar lines for commands such as PUSH, PRESS, PULL, TURN, and so on.

Finally, we need to deal with a special case. In general, the player cannot interact with other things in a dark room because he can't see them, but if we adhered strictly to this it would be impossible for him to find the light switch to turn it back on. So we need something from the chapter on Activities to change this:

After deciding the scope of the player when the location is the Terrifying Basement:
place the light switch in scope.

Upstairs is above the Terrifying Basement.

Test me with "turn off light / look / flip light switch".

51



Example The Undertomb 2

WI

Flickering lantern-light effects added to the Undertomb.

"Undertomb 2"

Section 1 - Procedure

Brightness is a kind of value. The brightnesses are guttering, weak, radiant and blazing.

A brightness can be adequate or inadequate. A brightness is usually adequate. Guttering is inadequate.

Temperature is a kind of value. 100C specifies a temperature.

A brightness has a temperature. The temperature of a brightness is usually 700C. The temperature of blazing is 1400C. The temperature of radiant is 1100C.

A dead end is a kind of room with printed name "Dead End" and description "This is a dead end, where crags in the uneven rock are caught by the [brightness of the lantern] flame you hold aloft. Despite [river sound] there is no sign of the stream." A dead end is usually dark.

A dead end has some text called river sound. The river sound of a dead end is usually "a faint whispering of running water".

Section 2 - Scenario

The Undertomb is a dark room. "From this dim cross-groined room, exits depart east, south, and northwest."

East is a dead end. South is a dead end with printed name "Collapsed Dead End".

Northwest is a dead end called the Tortuous Alcove. The Tortuous Alcove has river sound "a gurgle of running water".

The player carries a book. The description of the book is "[if the brightness of the lantern is adequate]Many secrets are now yours.[otherwise]No, the print's too tiny by this awful light."

In the Undertomb is a lantern. It is lit. The lantern has a brightness. The lantern is blazing. The description of the lantern is "The lantern shines with a flame at [temperature of the brightness of the lantern]."

After waiting in the Tortuous Alcove when the brightness of the lantern is not guttering:

now the lantern is the brightness before the brightness of the lantern;
say "You wait so long that your lantern dims a bit."

Test me with "east / west / get lantern / east / west / south / north / northwest / read book / x lantern / z / x lantern / read book / look / z / x lantern / read book / look / z / x lantern / read book / look".

348



Example Hohmann Transfer

WI

Changing the way dark rooms are described to avoid the standard Inform phrasing.

Inform automatically keeps track of light and darkness, handling such questions as whether a room is lit, whether the player can see any light sources, etc., and then managing the descriptions accordingly. When the room is dark and no light sources are visible, the player is said to be "in darkness".

If we don't specify otherwise, Inform will describe our surroundings in a dark room thus:

Darkness

It is pitch dark, and you can't see a thing.

This is fine in many situations, but we may sometimes want to replace this phrase with something else.

"Hohmann Transfer"

The Western Hemisphere is a dark room. "The cloud mass covers much of the land on this side of the planet, and a particularly nasty storm is brewing off to the south."

The Eastern Hemisphere is west of the Western Hemisphere. The Eastern Hemisphere is east of the Western Hemisphere. The Eastern Hemisphere is north of the Western Hemisphere. The Eastern Hemisphere is south of the Western Hemisphere. "This side of the planet is more ocean than land, with only two continents worthy of the name, and a volcanic archipelago in the north seas."

Use full-length room descriptions.

Rule for printing the description of a dark room:

say "It's night on this side of the planet, so you can make out only the glow of urbanized areas along the seacoasts." instead.

Rule for printing the name of a dark room:

say "Dark Side" instead.

And now a few minor refinements so that we can see what happens when one room becomes dark and the other light:

Carry out going:

say "You fire the thrusters and loop around to the other side of the planet before settling into a new geosynchronous orbit. Six months and one minute later..."

The time of day is 4:55 PM.

At 5 PM:

now the Eastern Hemisphere is dark;
now the Western Hemisphere is lit.

Rule for printing the announcement of darkness:

say "The planet abruptly spins itself over, exposing its cool underbelly to the sun."

Test me with "e / z / z / w / z / z / e".

81



Example Unblinking

WI

Finding a best route through light-filled rooms only, leaving aside any that might be dark.

Suppose we're simulating a situation where the player needs to travel through lit areas only, but we want to give him some hints about which way might be safe. Here we'll find our best route through light-filled rooms.

The slightly tricky part is that it's not necessarily easy to tell whether a room has a lamp in it. We may say "if the Crypt is lighted", but that only tells us whether it has been declared to be inherently lighted or dark, not whether it happens to contain a light source that the player would be able to see if he went in.

The easiest way to get around this is to create an object -- the light-meter; place it in the target location; and check whether it "can see" a lit object. This preserves all the usual rules about open and closed containers, transparency, etc.

"Unblinking"

Section 1 - Procedure

The light-meter is a privately-named scenery thing.

Definition: a room (called the target room) is light-filled:

if the target room is lighted:
yes;
move the light-meter to the target room;
let the answer be false;
if the light-meter can see a lit thing:
now the answer is true;

now the light-meter is nowhere;
decide on the answer.

That done, we're free to use our best-route phrases to choose a particular route.

Section 2 - Scenario

The Tomb of Angels is a room. "This ancient underground tomb is lightless but for a few shafts from the surface. Everywhere in the shadows are carved angels, their faces worn away by water and pollution, their wings little more than nubs."

The Upward Path is above the Tomb of Angels. It is dark. "The staircase switches back on itself many times as it ascends towards the Crash Site."

A container called the sarcophagus is in the Upward Path. It is closed and openable. "A sarcophagus rests in the niche here, [if open]the lid pushed aside[otherwise]the lid firmly in place[end if]."

The Crash Site is above the Upward Path. "The ceiling has wholly caved in here, and the belly of the spaceship above you is visible -- including the escape hatch."

A candle is a kind of thing. A candle is usually lit. The player carries four candles.

After looking when the location is not the Crash Site:

if the best route from the location to the Crash Site through light-filled rooms is a direction (called next way):

say "It looks like there's a safe, lit path [if the number of moves from the location to the Crash Site through light-filled rooms is 1]straight[otherwise]if you go[end if] [next way].";

otherwise:

say "It looks like there is no fully lit path from here to the Crash Site."

Test me with "up / drop candle / down / up / take the candle / open sarcophagus / put candle in sarcophagus / down / up / close sarcophagus / d".

An important word of caution: this method would give false negatives if there were a backdrop lightsource, such as the moon, providing light to the Upward Path. This is because backdrops are actually moved around the map by Inform during play, following the player around. So if the moon backdrop is in the Crash Site with the player, it will not be in the Upward Path as well -- even if it's scheduled to move there as soon as the player does.

347



Example Zorn of Zorna

Light levels vary depending on the number of candles the player has lit, and this determines whether or not he is able to examine detailed objects successfully.

WI

"Zorn of Zorna"

Visibility rule:

if examining:
if the detail of the noun is fine and the number of visible lit candles is less than 5, there is insufficient light;
if the detail of the noun is ordinary and the number of visible lit candles is less than 3, there is insufficient light;
there is sufficient light.

Detail is a kind of value. The details are fine, ordinary, and gross. A thing has detail.

A candle is a kind of thing. Before printing the name of a candle while not burning or blowing out: say "[if lit]lit [otherwise]unlit [end if]". A candle is usually lit. Before printing the plural name of a candle while not burning or blowing out: say "[if lit]lit [otherwise]unlit [end if]". A candle is usually lit. Understand the lit property as describing a candle. A candle is usually gross.

Instead of burning a candle: now the noun is lit; say "You light [the noun]."

Understand "blow out [something]" or "extinguish [something]" or "put out [something]" as blowing out. Understand the command "snuff" as "extinguish". Blowing out is an action applying to one thing.

Understand "burn [unlit candle]" as burning.

Instead of blowing out a candle:
now the noun is unlit;
say "You put out [the noun]."

Rule for printing a refusal to act in the dark:

if we are examining something, say "The details of [the noun] are too fine to make out in the light of only [the number of visible lit candles in words] candle[s]." instead.

Every turn when the Todal is visible:

if the number of visible lit candles is greater than 1:
say "The brightness of the room wakens the Todal from slumber, and with you unarmed...";
end the story;
otherwise:
say "Todal sleeps fitfully, troubled by even that faint light."

A room is usually dark.

The Palace is a room. "The Duke is out; the way is clear. East is Saralinda's Chamber; north, a hallway zigs and zags down to the gate that leads out." A finely-written placard is in the Palace. "A finely-written placard is on the wall next to this exit." The placard is fine. The description of the placard is "You read: 'Beware the Todal: its bite is worse than its gleep."

No more than one candle!"

The candle-stand is a supporter in the Palace. Understand "stand" as the candle-stand. The description of the candle-stand is "The candle-stand is a tall metal branch for holding lights, but someone has quite practically added casters

to the bottom." It is pushable between rooms. Three candles are on the candle-stand. Instead of removing something from the candle-stand: say "[The noun] is fixed quite firmly in place." Instead of taking something which is on the candle-stand: say "[The noun] won't come out of the holder." Instead of putting something on the candle-stand: say "[The candle-stand] is full."

Saralinda's Chamber is east of the Palace. "Now that Saralinda herself is gone, there is no real radiance in this place." Two unlit candles are in Saralinda's Chamber.

A large-print romantic novel is in Saralinda's Chamber. The novel is ordinary. The description of the novel is "'She Was Only The Chimney-Sweep's Daughter', by Marie Swelldon."

The Zig-Zag Hallway is north of the Palace. "The Hallway goes left, then right, then left again..." Two unlit candles are in the Hallway.

Todal is an animal in the Zig-Zag Hallway.

Rule for printing the description of a dark room when the Todal is in the location:
try listening.

Instead of listening when in darkness and the Todal is in the location:
say "In the darkness something softly gleeps."

Instead of going north from the Hallway when in darkness:
say "You stumble and cannot find your way."

North of the Hallway is Freedom. Instead of going to Freedom: say "You make it out into the cool night air at last!"; end the story finally.

Test me with "examine placard / get placard / n / listen / n / s / examine candle-stand / push candle-stand east / examine novel / get unlit candle / light it / light unlit candle / examine placard / push candle-stand west / e / examine novel / w / n / n".

349



Example Four Stars 1

WI

An elaboration of the idea that when light is absent, the player should be given a description of what he can smell and hear, instead.

"Four Stars"

A thing has some text called sound. The sound of a thing is usually "silence".

The report listening rule is not listed in the report listening to rules.

Carry out listening to something:
say "From [the noun] you hear [the sound of the noun]."

Instead of listening to a room:

if an audible thing can be touched by the player, say "You hear [the list of audible things which can be touched by the player].";
otherwise say "A merciful peace prevails."

Definition: a thing is audible if the sound of it is not "silence".

Before printing the name of something audible while listening to a room:
say "[sound] from the "

A thing has some text called scent. The scent of a thing is usually "nothing".

The report smelling rule is not listed in the report smelling rulebook.

Carry out smelling something:

say "From [the noun] you smell [scent of the noun]."

Instead of smelling a room:

if a scented thing can be touched by the player, say "You smell [the list of scented things which can be touched by the player].";
otherwise say "The place is blissfully odorless."

Definition: a thing is scented if the scent of it is not "nothing".

Before printing the name of something scented while smelling a room: say "[scent] from the "

The Waning Moon Resort is a dark room. "A spacious room with a flagstone floor, and a dreamcatcher hung over the king-size bed." The dreamcatcher is scenery in the Resort. The description is "The usual web of threads and crystals, feathers and beads." Instead of taking the dreamcatcher, say "Ah, ah -- you might be tempted to take it as a souvenir, except that the price list in the minibar clearly states they charge \$65 apiece if you walk off with one. Cheaper than stealing the Frette bathrobes, but still probably not a good idea."

The king-size bed is an enterable supporter in the Resort. The description is "200-thread-count Egyptian cotton sheets, according to the website. You would make fun, only they really are extraordinarily comfortable." The player is on the bed. A Lindt chocolate is on the bed. It is edible. The scent of the chocolate is "chocolate-hazelnut smell".

Instead of exiting: say "You are too weary to move."

The suitcase is an openable closed container in the Resort.

An electric light is a kind of device. Carry out switching on an electric light: now the noun is lit. Carry out switching off an electric light: now the noun is unlit. Understand "light" as an electric light.

The solar lamp is an electric light in Waning Moon Resort. The description is "Specially designed to give light in a spectrum resembling sunlight, to improve the mood and make a person energetic." The lamp is switched on and lit.

An electric noisemaker is a kind of device. An electric noisemaker has some text called usual sound. The usual sound of an electric noisemaker is usually "beepbeepbeep". Carry out switching on an electric noisemaker: now the sound of the noun is the usual sound of the noun. Report switching on an electric noisemaker: say "[The noun] goes [usual sound of the noun]!" instead. Report switching off an electric noisemaker: say "You switch off [the noun], silencing the [usual sound of the noun]." instead.

Carry out switching off an electric noisemaker: now the sound of the noun is "silence".

The bedside table is in the Resort. The table supports a potted plant and a Bose speaker. The scent of the potted plant is "rosemary"

The Bose speaker is an electric noisemaker. The usual sound of the speaker is "soothing whalesong". The sound of the speaker is "soothing whalesong". The speaker is switched on.

Instead of touching a device: say "You feel the surface of [the noun] and discover the switch."

Instead of touching a scented thing: say "The brush of your fingers stirs loose a fresh cloud of [scent of the noun] smell."

Rule for printing the description of a dark room: try listening; try smelling; rule succeeds.

Instead of examining an audible thing while in darkness: try listening to the noun. Instead of examining something while in darkness: try touching the noun.

Before touching something when in darkness:
say "You grope about..."

After inserting the plant into something:
say "You unceremoniously dump [the noun] into [the second noun], hoping it sustains no important damage thereby."

Before printing the name of a dark room: if the player can touch an audible thing, say "Noisy "; if the player can touch a scented thing, say "Perfumed ".

Visibility rule when in darkness:
if examining something, there is sufficient light;
there is insufficient light.

Rule for printing the announcement of darkness: say "It is now pleasantly lightless in here." instead.

Rule for deciding the scope of the player while in darkness: place the location in scope.

To decide whether in daylight:
if in darkness, no;
yes.

Instead of sleeping when in daylight:
say "You've never been able to sleep with the light on."

Instead of sleeping when the player can touch an audible thing (called the irritant):
say "The steady [sound of the irritant] from [the irritant] prevents your slumber."

Instead of sleeping when the player can touch a scented thing (called the irritant):
if the irritant is chocolate, say "The smell of chocolate continues to tantalize you, keeping you from sleep."
otherwise say "You sniffle. [The irritant] is probably acting on your allergies."

Instead of sleeping:
say "You slip easily into the arms of Morpheus."
end the story finally saying "At last..."

When play begins:
say "You have at last escaped from the airport and gotten through customs; survived an unnerving taxi ride over icy highways; stared down the impertinent concierge; endured the bellhop's catalog of features in your room; and achieved, finally, a moment of peace. Time for a good night's slumber!"

Test me with "x dreamcatcher / switch lamp off / look / sleep / eat chocolate / sleep / get plant / examine plant / open suitcase / put plant in suitcase / close suitcase / sleep / look / examine base / switch base off / sleep".

291



Example Cloak of Darkness

WI

Implementation of "Cloak of Darkness", a simple example game that for years has been used to demonstrate the features of IF languages.

"Cloak of Darkness" is a brief example game that has been implemented in nearly every IF system currently used. It hasn't got much claim to complexity or richness, but it does exemplify many of the standard things one might want an IF language to be able to do: define descriptions and the results of actions, assign synonyms to nouns, create new verbs, handle darkness, track repeated acts, and so on.

Here is what the game looks like in Inform:

"Cloak of Darkness"

The story headline is "A basic IF demonstration."

Use scoring.

The maximum score is 2.

Whatever room we define first becomes the starting room of the game, in the absence of other instructions:

Foyer of the Opera House is a room. "You are standing in a spacious hall, splendidly decorated in red and gold, with glittering chandeliers overhead. The entrance from the street is to the north, and there are doorways south and west."

Instead of going north in the Foyer, say "You've only just arrived, and besides, the weather outside seems to be getting worse."

We can add more rooms by specifying their relation to the first room. Unless we say otherwise, the connection will automatically be bidirectional, so "The Cloakroom is west of the Foyer" will also mean "The Foyer is east of the Cloakroom":

The Cloakroom is west of the Foyer. "The walls of this small room were clearly once lined with hooks, though now only one remains. The exit is a door to the east."

In the Cloakroom is a supporter called the small brass hook. The hook is scenery. Understand "peg" as the hook.

Inform will automatically understand any words in the object definition ("small", "brass", and "hook", in this case), but we can add extra synonyms with this sort of Understand command.

The description of the hook is "It's just a small brass hook, [if something is on the hook]with [a list of things on the hook] hanging on it[otherwise]screwed to the wall[end if]."

This description is general enough that, if we were to add other hangable items to the game, they would automatically be described correctly as well.

The Bar is south of the Foyer. The printed name of the bar is "Foyer Bar". The Bar is dark. "The bar, much rougher than you'd have guessed after the opulence of the foyer to the north, is completely empty. There seems to be some sort of message scrawled in the sawdust on the floor."

The scrawled message is scenery in the Bar. Understand "floor" or "sawdust" as the message.

Neatness is a kind of value. The neatnesses are neat, scuffed, and trampled. The message has a neatness. The message is neat.

We could if we wished use a number to indicate how many times the player has stepped on the message, but Inform also makes it easy to add descriptive properties of this sort, so that the code remains readable even when the reader does not know what "the number of the message" might mean.

Instead of examining the message:
increment score;
say "The message, neatly marked in the sawdust, reads...";
end the story finally.

This second rule takes precedence over the first one whenever the message is trampled. Inform automatically applies whichever rule is most specific:

Instead of examining the trampled message:
say "The message has been carelessly trampled, making it difficult to read.
You can just distinguish the words...";
end the story saying "You have lost".

This command advances the state of the message from neat to scuffed and from scuffed to trampled. We can define any kinds of value we like and advance or decrease them in this way:

Instead of doing something other than going in the bar when in darkness:
if the message is not trampled, now the neatness of the message is the
neatness after the neatness of the message;
say "In the dark? You could easily disturb something."

Instead of going nowhere from the bar when in darkness:
now the message is trampled;
say "Blundering around in the dark isn't a good idea!"

This defines an object which is worn at the start of play. Because we have said the player is wearing the item, Inform infers that it is clothing and can be taken off and put on again at will.

The player wears a velvet cloak. The cloak can be hung or unhung. Understand "dark" or "black" or "satin" as the cloak. The description of the cloak is "A handsome cloak, of velvet trimmed with satin, and slightly splattered with raindrops. Its blackness is so deep that it almost seems to suck light from the room."

Carry out taking the cloak:
now the bar is dark.

Carry out putting the unhung cloak on something in the cloakroom:
now the cloak is hung;
increment score.

Carry out putting the cloak on something in the cloakroom:
now the bar is lit.

Carry out dropping the cloak in the cloakroom:
now the bar is lit.

Instead of dropping or putting the cloak on when the player is not in the cloakroom:
say "This isn't the best place to leave a smart cloak lying around."

When play begins:
say "[paragraph break]Hurrying through the rainswept November night, you're glad to see the bright lights of the Opera House. It's surprising that there aren't more people about but, hey, what do you expect in a cheap demo game...?"

Understand "hang [something preferably held] on [something]" as putting it on.

Test me with "s / n / w / inventory / hang cloak on hook / e / s / read message".

And that's all. As always, type TEST ME to watch the scenario play itself out.

50

★ Example The Undertomb 1

WI

A small map of dead ends, in which the sound of an underground river has different strengths in different caves.

This example creates a kind of room, "dead end", and gives each a textual description called its "river sound". Note the use of a text substitution to fill in the appropriate river sound text in each dead end.

"The Undertomb"

A dead end is a kind of room with printed name "Dead End" and description "This is a dead end. You'll have to go back the way you came, consoled only by [river sound]." A dead end is usually dark.

The Undertomb is a dark room. East is a dead end. South is a dead end with printed name "Collapsed Dead End". Northwest is a dead end called the Tortuous Alcove. In the Undertomb is the lantern. It is lit.

A dead end has some text called river sound. The river sound of a dead end is usually "a faint whispering of running water". The Tortuous Alcove has river sound "a gurgle of running water".

Test me with "get lantern / e / w / s / n / nw / se".

364

★ Example Four Stars 2

WI

Using "deciding the scope" to change the content of lists such as "the list of audible things which can be touched by the player".

As we have seen, a well-written understand rule will often solve the problem of allowing the player to apply specific actions to objects not normally in scope. When we need to adjust scope for some other reason than reading the player's command, though, "deciding the scope of..." may come in handy.

For instance, suppose we wanted to extend Four Stars 1 to add a tomcat on the balcony that will be heard whenever the player listens from the next room, as in:

>listen

You hear the soothing whalesong from the Bose speaker and the yowling from the tomcat.

To do this, we need to make sure that in the rule that assembles our listening description,

Instead of listening to a room:

if an audible thing can be touched by the player, say "You hear [the list of audible things which can be touched by the player].";
otherwise say "A merciful peace prevails."

now includes the tomcat in the "list of audible things which can be touched by the player".

To this end, we're going to change the way we assess scope, but only during the listening action. Otherwise the tomcat remains in the other room and off-limits. The new source text is marked out below:

"Four Stars"

Section 1 - Procedure

A thing has some text called sound. The sound of a thing is usually "silence".

The report listening rule is not listed in the report listening to rules.

Carry out listening to something:

say "From [the noun] you hear [the sound of the noun]."

Instead of listening to a room:

if an audible thing can be touched by the player, say "You hear [the list of audible things which can be touched by the player].";
otherwise say "A merciful peace prevails."

Definition: a thing is audible if the sound of it is not "silence".

Before printing the name of something audible while listening to a room:

say "[sound] from the "

A thing has some text called scent. The scent of a thing is usually "nothing".

The report smelling rule is not listed in the report smelling rulebook.

Carry out smelling something:

say "From [the noun] you smell [scent of the noun]."

Instead of smelling a room:

if a scented thing can be touched by the player, say "You smell [the list of scented things which can be touched by the player].";
otherwise say "The place is blissfully odorless."

Definition: a thing is scented if the scent of it is not "nothing".

Before printing the name of something scented while smelling a room: say "[scent] from the "

Here is our addition:

After deciding the scope of the player while listening or sleeping or looking:
if in darkness:
repeat with locale running through adjacent rooms:
place locale in scope.

A reaching inside rule while listening or sleeping or looking:
rule succeeds.

Section 2 - Scenario

The Waning Moon Resort is a dark room. "A spacious room with a flagstone floor, and a dreamcatcher hung over the king-size bed." The dreamcatcher is scenery in the Resort. The description is "The usual web of threads and crystals, feathers and beads." Instead of taking the dreamcatcher, say "Ah, ah -- you might be tempted to take it as a souvenir, except that the price list in the minibar clearly states they charge \$65 apiece if you walk off with one. Cheaper than stealing the Frette bathrobes, but still probably not a good idea."

And now our threat to the player's peace:

The Balcony is outside from the Resort. In the Balcony is a tomcat. The sound of the tomcat is "yowling". After printing the name of the tomcat when the tomcat is not visible: say " outside on the balcony".

From here we continue with the same scenario as before:

The king-size bed is an enterable supporter in the Resort. The description is "200-thread-count Egyptian cotton sheets, according to the website. You would make fun, only they really are extraordinarily comfortable." The player is on the bed. A Lindt chocolate is on the bed. It is edible. The scent of the chocolate is "chocolate-hazelnut smell".

Instead of exiting: say "You are too weary to move."

The suitcase is an openable closed container in the Resort.

An electric light is a kind of device. Carry out switching on an electric light: now the noun is lit. Carry out switching off an electric light: now the noun is unlit. Understand "light" as an electric light.

The solar lamp is an electric light in Waning Moon Resort. The description is "Specially designed to give light in a spectrum resembling sunlight, to improve the mood and make a person energetic." The lamp is switched on and lit.

An electric noisemaker is a kind of device. An electric noisemaker has some text called usual sound. The usual sound of an electric noisemaker is usually "beepbeepbeep". Carry out switching on an electric noisemaker: now the sound of the noun is the usual sound of the noun. Report switching on an electric noisemaker: say "[The noun] goes [usual sound of the noun]!" instead. Report switching off an electric noisemaker: say "You switch off [the noun], silencing the [usual sound of the noun]." instead.

Carry out switching off an electric noisemaker: now the sound of the noun is "silence".

The bedside table is in the Resort. The table supports a potted plant and a Bose speaker. The scent of the potted plant is "rosemary"

The Bose speaker is an electric noisemaker. The usual sound of the speaker is "soothing whalesong". The sound of the speaker is "soothing whalesong". The speaker is switched on.

Instead of touching a device: say "You feel the surface of [the noun] and discover the switch."

Instead of touching a scented thing: say "The brush of your fingers stirs loose a fresh cloud of [scent of the noun] smell."

Rule for printing the description of a dark room: try listening; try smelling; rule succeeds.

Instead of examining an audible thing while in darkness: try listening to the noun.
Instead of examining something while in darkness: try touching the noun.

Before touching something when in darkness:
say "You grope about..."

After inserting the plant into something:
say "You unceremoniously dump [the noun] into [the second noun], hoping it sustains no important damage thereby."

Before printing the name of a dark room: if the player can touch an audible thing, say "Noisy "; if the player can touch a scented thing, say "Perfumed ".

Visibility rule when in darkness:
if examining something, there is sufficient light;
there is insufficient light.

Rule for printing the announcement of darkness: say "It is now pleasantly lightless in here." instead.

Rule for deciding the scope of the player while in darkness: place the location in scope.

To decide whether in daylight:
if in darkness, no;
yes.

Instead of sleeping when in daylight:
say "You've never been able to sleep with the light on."

Instead of sleeping when the player can touch an audible thing (called the irritant):
say "The steady [sound of the irritant] from [the irritant] prevents your slumber."

Instead of sleeping when the player can touch a scented thing (called the irritant):
if the irritant is chocolate, say "The smell of chocolate continues to tantalize

you, keeping you from sleep.";
otherwise say "You sniffle. [The irritant] is probably acting on your allergies."

Instead of sleeping:
say "You slip easily into the arms of Morpheus.";
end the story finally saying "At last..."

When play begins:
say "You have at last escaped from the airport and gotten through customs;
survived an unnerving taxi ride over icy highways; stared down the impertinent
concierge; endured the bellhop's catalog of features in your room; and achieved,
finally, a moment of peace. Time for a good night's slumber!"

Test me with "listen / x dreamcatcher / switch lamp off / look / sleep / eat
chocolate / sleep / get plant / examine plant / open suitcase / put plant in
suitcase / close suitcase / sleep / look / examine base / switch base off / sleep".

Of course, this new version is less happy for the player, as we haven't included any
way to silence the cat.

95



Example The Art of Noise

WI

Things are all assigned their own noise (or silence). Listening to the
room in general reports on all the things that are currently audible.

This example involves redesigning the LISTEN command, removing its built-in
function and replacing that with something more ambitious. We will learn more
about how to do this later on.

"The Art of Noise"

A thing has some text called sound. The sound of a thing is usually "silence".

The report listening rule is not listed in the report listening to rules.

Carry out listening to something:
say "From [the noun] you hear [the sound of the noun]."

Instead of listening to a room:
if an audible thing can be touched by the player, say "You hear [the list of
audible things which can be touched by the player].";
otherwise say "Nothing of note."

Definition: a thing is audible if the sound of it is not "silence".

Before printing the name of something audible while listening to a room:
say "[sound] from the "

The Sharper Image is a room. The pet rock is a thing in the Sharper Image. The
toy car is a thing in the Sharper Image. The sound of the car is "whirring and
zooming". The plastic widget is a thing in the Sharper Image. The sound of the

plastic widget is "bleeps and bloops". The pointless gadget is a thing in Sharper Image. The sound of the pointless gadget is "buzzbuzzbuzz".

The soundproof case is a transparent openable container in the Sharper Image. It is closed and fixed in place.

Test me with "listen / listen to rock / listen to car / get all / open case / put all in case / listen / close case / listen / listen to car".

131

Example Weathering

WI

The automatic weather station atop Mt. Pisgah shows randomly fluctuating temperature, pressure and cloud cover.

"Weathering"

A cloud pattern is a kind of value. The cloud patterns are cumulus, altocumulus, cumulonimbus, stratus, cirrus, nimbus, nimbostratus.

The Mount Pisgah Station is a room. "The rocky peak of Mt. Pisgah (altitude 872m) is graced only by an automatic weather station. The clouds, close enough almost to touch, are [a random cloud pattern]. Temperature: [a random number from 7 to 17] degrees, barometric pressure: [950 + a random number from 0 to 15] millibars."

Test me with "look / look / look".

157

Example Full Moon

WI

Random atmospheric events which last the duration of a scene.

"Full Moon"

Wolf Pursuit is a scene. Wolf Pursuit begins when play begins.

Every turn during Wolf Pursuit, say "[one of]A twig snaps behind you![or]The wind howls in your ears.[or]You feel chilly.[at random]".

When play begins:

say "You have lost your spectacles, and the lamp, and can see barely further than the next tree. Roots keep trying to trip you, too..."

The Dark Forest is a room. "You are mostly aware that you are not as alone here as you would like, and that the ground is uneven."

Test me with "z / z / z / z".

Example Night and Day

Cycling through a sequence of scenes to represent day and night following one another during a game.

Suppose we want to have a sequence of nights and days in our game, with one scene to govern each daylight condition.

"Night and Day"

The sun is a backdrop. It is everywhere. The description is "Currently out of sight."

Night is a recurring scene. Night begins when play begins. Night begins when Dusk ends. Night ends when the time since Night began is 3 minutes.

Notice that our two conditions for the beginning of Night are not in conflict: it will be night-time when the game begins, and then night will also recur every time the Dusk scene ends.

When Night begins:

say "The sun falls below the horizon and the temperature drops abruptly to well below zero.";

now the description of the sun is "Currently out of sight."

Dawn is a recurring scene. Dawn begins when Night ends. Dawn ends when the time since Dawn began is 1 minute.

When Dawn begins:

say "The sun appears on the horizon.";

now the description of the sun is "It is tiny and weak."

Day is a recurring scene. Day begins when Dawn ends. Day ends when the time since Day began is 3 minutes.

When Day begins:

say "The sun is now properly up."

Dusk is a recurring scene. Dusk begins when Day ends. Dusk ends when the time since Dusk began is 1 minute.

When Dusk begins:

say "The sun has passed across the sky and is on the verge of setting."

Cratered Landscape is a room. "The ground here is [if Night is happening]dim silver, with the craters visible as darker splotches[otherwise]the color of dried blood; here and there it is also rippled by impact craters[end if]. The horizon curves visibly."

Test me with "z / z / z / look / x sun / z / z / z / z / z / z / z / z".

If we run this example and then have a look at the scenes index, we'll see that the cycle is listed through thus:

Night (recurring)
Dawn (recurring)
Day (recurring)
Dusk (recurring)
Night

with the second "Night" in italics, to indicate that it is a repetition of the same scene that has already been listed above.

144



Example Totality

WI

To schedule an eclipse of the sun, which involves a number of related events.

First we define the events, and then we create a phrase to schedule them:

"Totality"

At the time when the penumbra appears:
say "The sunlight dies away to an eerie, brownish penumbra."

At the time when the eclipse begins:
say "The moon passes across the disc of the sun, plunging the world into darkness.";
now the Chamber is dark.

At the time when the eclipse ends:
say "The moon passes away from the sun, and a gloomy penumbral light returns.";
now the Chamber is lighted.

At the time when normal sunlight returns:
say "The sun shines once more bright in the sky, not to be eclipsed again on this spot for another thirty-seven years."

To schedule an eclipse for (totality - a time):
the penumbra appears at two minutes before totality;
the eclipse begins at totality;
the eclipse ends at three minutes after totality;
normal sunlight returns at five minutes after totality.

Now we make use of the new phrase:

When play begins, schedule an eclipse for 3:27 PM.

The Chamber is a room.

The time of day is 3:25 PM.

Test me with "z / z / z / z / z / z / z / z / z / z".

We shall see much more about creating phrases later. Their advantage is that they enable a complicated sequence of operations to be given a meaningful name, and that they can be re-used many times as needed.

120



Example Orange Cones

WI

Creating a traffic backdrop that appears in all road rooms except the one in which the player has laid down orange cones.

Because we can invent our own adjectives (see the chapter on Phrases), we can make the conditions for a backdrop as simple or as complicated as we like.

In this scenario, we want the player to be able to take, move, and drop orange traffic cones to seal off one street or another. So we create our own "accessible" adjective as follows:

Definition: a road is accessible if the orange cones are not in it.

...and now

move the traffic backdrop to all accessible roads.

will tell the traffic backdrop where to appear.

"Orange Cones"

A road is a kind of room.

The traffic is a backdrop. It is not scenery. The initial appearance is "Dense traffic snarls the streets, making it difficult to cross even with the lights. Men on motorbikes edge between the cars, and sometimes pull up onto the sidewalks to go around." The description is "It is more or less as usual for this time of day. It's a wonder it ever dissipates, really."

When play begins:

move the traffic backdrop to all accessible roads.

A line of orange cones are a thing.

Definition: a road is accessible if the orange cones are not in it.

After dropping the orange cones in a road:

say "With steely determination you begin to lay out the orange cones, blocking access to this segment of street. This produces honking and swearing -- but you persevere.";

update backdrop positions.

After taking the orange cones:

say "You go around taking up the orange cones, and within moments the traffic begins to flow into the street again.";

update backdrop positions.

That accomplishes everything we set out to do, but let's add a very simple puzzle to test it out with:

The Town Square is a road. North of the Town Square is Candle Street. Candle Street is a road. East of Town Square is Mortar Street. Mortar Street is a road.

The line of orange cones are in Mortar Street.

The player wears a reflective vest and a hard hat.

The manhole cover is a door. "[if location is accessible]Under the cars in the middle of the street, you can just make out the cover of the manhole you need to get into.[otherwise]There's a promising manhole in the middle of the street.[end if]". It is closed and openable. It is below the Town Square and above the Access Tunnel.

Instead of opening the manhole cover when the location is an accessible road:
say "You can't get anywhere near the manhole cover with all these cars above it."

Instead of taking the orange cones in Town Square:
if the manhole cover is closed or Town Square is accessible, continue the action;
otherwise say "You'd better not let the traffic back in until you've closed the manhole. There'll be accidents otherwise."

After going to the Access Tunnel:
say "With a stealthy glance left and right, you lower yourself into the access tunnel, thus accomplishing your mission for Chapter 2. To continue your adventure, see Chapter 3: The Vault of Peaquod.";
end the story finally.

And finally, a couple of features from the Activities chapter to make the output more elegant:

Rule for writing a paragraph about the orange cones: say "A line of orange cones holds back the traffic from entering here."

Rule for printing the name of the orange cones when the cones are carried by the player:
say "stack of orange cones".

Test me with "x traffic / open manhole / n / x traffic / s / e / x traffic / get cones / look / x traffic / w / drop cones / look / open manhole / take cones / d".



A stream of random pedestrians who go by the player.

Suppose we have an urban space we want to populate with random passers-by. These should have a range of characteristics and not always be described in the same way;

and once the player has noticed one, he should be able to look at her further, until another pedestrian crosses his path.

"Uptown Girls"

Riverside Drive is a room. "There's a pleasant late-afternoon view of the Hudson, and a snap in the air, and you would rather be here than anywhere."

Instead of going a direction, say "Oh, you know where you're going; no need to deviate from the usual path."

Instead of waiting, say "You stroll along enjoying the November crispness."

It gets a little annoying to have a random event occurring every single turn of play, so let's introduce some randomness to determine how often the message appears:

Every turn when a random chance of 1 in 3 succeeds:
reset passerby;
choose a random row in the Table of Atmospheric Events;
say "[event entry][paragraph break]"

Table of Atmospheric Events

event

"Slowly [a passerby] strolls by, turning to look at you as she passes."

"Some [passerby] nearly bumps into you."

"You dodge to avoid [a passerby]."

"You weave around [a passerby], who has stalled to look into a window."

"There's a ruckus as one of the ubiquitous taxis nearly collides with [a passerby] crossing the street."

"[The passerby] beside you waves to a friend across the street."

"To your left, [a passerby] drops her purse, and swears as she retrieves it."

Hair color is a kind of value. A person has hair color. the hair colors are red-headed, brunette, blonde.

Height is a kind of value. A person has height. The heights are tall, medium-height, short.

Grooming is a kind of value. A person has grooming. The groomings are messy and tidy.

To reset passerby:

now the hair color of the passerby is a random hair color;

now the height of the passerby is a random height;

now the grooming of the passerby is a random grooming.

The passerby is a woman in Riverside Drive. The passerby is scenery. Understand "woman" or "lady" as the passerby. The printed name of the passerby is "[one of]woman[or]lady[purely at random]".

At this point we borrow some techniques from later to describe the woman with a random combination of characteristics, and to let the player refer to her by those traits:

Before printing the name of the passerby:

- if a random chance of 1 in 4 succeeds, say "[height] ";
- if a random chance of 1 in 4 succeeds, say "[grooming] ";
- if a random chance of 1 in 4 succeeds, say "[hair color] ".

Understand the hair color property as describing the passerby. Understand the height property as describing the passerby. Understand the grooming property as describing the passerby.

If we also wanted each of those combinations to mean some more specifically-described woman:

Instead of examining a passerby:

repeat through Table of Passerby Descriptions:

if hair entry is the hair color of the passerby and height entry is the height of the passerby and grooming entry is the grooming of the passerby, say "[description entry][paragraph break]".

Table of Passerby Descriptions

hair	height	grooming	description
red-headed	tall	messy	"An older woman with long red hippie-hair poking out of a ponytail in straggles, and bent to hide how tall she is."
red-headed	medium-height	messy	"A shaggy red-head with shingled hair."
red-headed	short	messy	"Almost an urchin, and very young, with ginger hair and a smudged nose and far too many freckles."
red-headed	tall	tidy	"A precise career woman with henna-red hair."
red-headed	medium-height	tidy	"Her hair is red in the way that lollipops and fire trucks are red: not by nature but by art. The rest of her clothing is pretty ordinary, though."
red-headed	short	tidy	"Thin and small in every sense, with chin-length red-hair. Even high heels do not bring her head much above your shoulder."
brunette	tall	messy	"A Juno-esque woman with dark hair, wearing something resembling a tent."
brunette	medium-height	messy	"An unremarkable woman with dark brown hair and the aura of needing a wash."
brunette	short	messy	"There are mustard stains on the t-shirt of this short brown-haired woman. Estimated age ca. 40. Possibly homeless."
brunette	tall	tidy	"A leggy brunette in business attire."
brunette	medium-height	tidy	"Medium-height, brown-haired, generally nondescript."
brunette	short	tidy	"A neat little dark-haired girl."
blonde	tall	messy	"A tall blonde of about thirteen who looks as though she has not yet figured out how to get her wardrobe to catch up with her rate of growth. Her t-shirt and her pants are too short."
blonde	medium-height	messy	"Black leather pants and the wall-o-hair look."
blonde	short	messy	"One of those shocking platinum blonde types, with a tiger-patterned skirt. Reeeally trashy."
blonde	tall	tidy	"Elfin and severe, with perfectly straight hair falling to the middle of the back."
blonde	medium-height	tidy	"A rounded, Marilyn-esque blonde."
blonde	short	tidy	"Pin-precise in a blue-and-white striped suit and a boyish haircut."

Test me with "z / z / x passerby / z / z / x passerby".

Chapter 4: Time and Plot

§4.1. *The Passage Of Time*; §4.2. *Scripted Scenes*; §4.3. *Event Scheduling*;
§4.4. *Scene Changes*; §4.5. *Flashbacks*; §4.6. *Plot Management*

-  Contents of *The Inform Recipe Book*
-  Chapter 3: Place
-  Chapter 5: The Viewpoint Character
-  Indexes of the examples

§4.1. The Passage Of Time

A story that makes heavy use of time may want to give the player a hint that time is important - and an easy way to keep track of how it's going - by adding the current time to the status line, instead of the score. To do this, we would write

When play begins: change the right hand status line to "[time of day]".

All else being equal, time passes at a rate of one minute per turn. But this need not be so: we can imagine a story where turns take much less time, or much more; or a story in which the passage of time was sometimes suspended, or one in which different actions required different amounts of time to perform.

Situation Room provides a way to print 24-hour time, while **Zqlran Era 8** implements a completely new measurement of time, for a story set on an alien world.

Uptempo and **The Hang of Thursdays** speed up time's passage: turns take fifteen minutes in the former, or a quarter day in the latter.

Timeless makes certain actions instant, so that they don't count against the clock; this is sometimes useful in timed situations where the player needs to review the situation before going on with a tricky puzzle. **Endurance** systematically extends this idea to allow us to assign different durations to any action in the story. **The Big Sainsbury's** goes the opposite direction, and meticulously adds a minute to the clock for all implicit take actions, just so that the player isn't allowed to economize on moves.

An alternative approach to time is not to tell the player specifically what hour of the day it is at all, but to move from one general time period to another as it becomes appropriate - when the player has solved enough puzzles, or worked his way through enough of the plot. To this end we might use scenes representing, say, Thursday afternoon and then Thursday evening; then our scene rules, rather than the clock, would determine when Thursday afternoon stopped and Thursday evening began:

Thursday afternoon is a scene. Thursday evening is a scene.

Thursday afternoon ends when the player carries the portfolio.

Thursday evening begins when Thursday afternoon ends.
When Thursday evening begins:
say "The great clock over St. Margaret's begins to chime 6.";

Though this gives time a loose relation to the number of turns played, it feels surprisingly realistic: players tend to think of time in a story in terms of the number of *significant* moves they made, while the random wandering, taking inventory, and looking at room descriptions while stuck don't make as big an impression. So advancing the story clock alongside the player's puzzle solutions or plot progress can work just as well as any stricter calculation.

★ See **Passers-By, Weather and Astronomical Events** for cycles of day and night scenes

★ See **Waiting, Sleeping** for commands to let the player wait until a specific time or for a specific number of minutes

★ See **Clocks and Scientific Instruments** for clocks that can be set to times and that have analog or digital read-outs

★ See **Timed Input** for discussion of extensions allowing real-time input

-
- ↑ Start of Chapter 4: Time and Plot
 - ← Back to Chapter 3: Place: §3.9. Passers-By, Weather and Astronomical Events
 - Onward to §4.2. Scripted Scenes
 - ↓ Example 142: ★ **Situation Room** Printing the time of day in 24-hour time, as in military situations.
 - ↓ Example 377: ★ **The Big Sainsbury's** Making implicit takes add a minute to the clock, just as though the player had typed TAKE THING explicitly.
 - ↓ Example 394: ★ **Uptempo** Adjust time advancement so the game clock moves fifteen minutes each turn.
 - ↓ Example 409: ★ **Timeless** A set of actions which do not take any game time at all.
 - ↓ Example 410: ★★ **Endurance** Giving different actions a range of durations using a time allotment rulebook.
 - ↓ Example 183: ★★★ **The Hang of Thursdays** Turns take a quarter day each, and the game rotates through the days of the week.
 - ↓ Example 260: ★★★ **Zqlran Era 8** Creating an alternative system of time for our game, using new units.
-

§4.2. Scripted Scenes

Sometimes we want to arrange a scene in which something goes on in the background (as though it were a movie playing) while the player goes about his business; or where a series of things has to happen before the player gets to the end.

The simplest way to arrange background events for a scene is to write the sequence of events into a table and work our way through it, printing one line per turn, until the scene runs out. **Day One** does exactly this.

At other times, we want a scene to last as long as it takes the *player* to do something. **Entrapment** lets the player poke around and explore as much as he likes, but ends as soon as he has accomplished the scene's goal - which, unfortunately for him, is to get into an embarrassing situation so that another character can walk in and make fun of him. **The Prague Job** has a scene that requires the player to do a more specific set of tasks, but nags him and hurries him along until he's done.

Bowler Hats and Baby Geese assumes that our story is going to be assembled with a number of scenes, some of which will need to prevent the player from leaving the location until the scene is complete: it thus defines a "restricted" property for scenes, so that all such elements of the plot will work in the same way.

For more complex sorts of scripts and schedules, it may be worth consulting the extensions.

★ See **Characters Following a Script** for a character whose conversation with the player is scripted to follow a pattern and then conclude

-
- ↑ Start of Chapter 4: Time and Plot
 - ← Back to §4.1. The Passage Of Time
 - Onward to §4.3. Event Scheduling
 - ↓ Example 162: ★ **The Prague Job** Scenes used to provide pacing while the player goes through his possessions.
 - ↓ Example 155: ★★ **Entrapment** A scene in which the player is allowed to explore as much as he likes, but another character strolls in as soon as he has gotten himself into an awkward or embarrassing situation.
 - ↓ Example 159: ★★★ **Bowler Hats and Baby Geese** Creating a category of scenes that restrict the player's behavior.
 - ↓ Example 160: ★★★ **Day One** A scene which plays through a series of events in order, then ends when the list of events is exhausted.
-

§4.3. Event Scheduling

We can use a schedule of events to give some life to our environment: if we have a town setting, for instance, it makes sense for shops and libraries to open and close at set times; this is just what we find in **IPA**.

Air Conditioning Is Standard has characters who follow a timed schedule of events to interact with each other, while the player mostly wanders around missing out on the action. (Sometimes life is like that.) The same effects could have been achieved with scenes instead of clock times, but there are occasions when we do want to plan our characters' behavior to the minute rather than waiting for the player to be in the right place to observe it: in a murder mystery or a time-travel story, the exact timings might be quite significant.

We may also want to add events to the schedule during play, as in

Instead of pushing the egg-timer: say "It begins to mark time."; the egg-timer clucks in four turns from now.

At the time when the egg-timer clucks: say "Cluck! Cluck! Cluck! says the egg-timer."

Similarly, we can schedule things during play to happen at a specific time of day, as shown in **Hour of the Wren**.

★ See **Scene Changes** for more things that arrive at pre-determined times

★ See **Ships, Trains and Elevators** for a train that follows a schedule, carrying the player along if he is aboard

↑ Start of Chapter 4: Time and Plot

← Back to §4.2. Scripted Scenes

→ Onward to §4.4. Scene Changes

↓ Example 141: **☆☆ IPA** Shops which each have opening and closing hours, so that it is impossible to go in at the wrong times, and the player is kicked out if he overstays his welcome.

↓ Example 146: **☆☆☆ Hour of the Wren** Allowing the player to make an appointment, which is then kept.

↓ Example 354: **☆☆☆☆ Air Conditioning is Standard** Uses "writing a paragraph about" to make person and object descriptions that vary considerably depending on what else is going on in the room, including some randomized NPC interactions with objects or with each other.

§4.4. Scene Changes

In a plot that takes place over multiple locations or has several distinct scenes, we may want to move the player or change the scenery around him. **Age of Steam** brings a train on and off-stage as the plot requires. **Meteoric** similarly brings a meteor into view at a certain time of day, showing off several implementations depending on whether or not the player is supposed to be able to refer to the meteor after it has gone.

Entrevaux constructs an organized system such that all scenes have their own lists of props and associated locations, and props are moved on and off automatically. Scene changes are also announced with a pause and a new title, such as "Chapter 2: Abduction".

Space Patrol - Stranded on Jupiter inserts an interlude in which the player's possessions and clothes are switched for new ones and the player moved to a new location - and then put back where he started from.

★ See **Flashbacks** for more ways to move the player from one level of reality to another

- ⬆ Start of Chapter 4: Time and Plot
 - ⬅ Back to §4.3. Event Scheduling
 - ➡ Onward to §4.5. Flashbacks
 - ⬇ Example 156: **★ Age of Steam** The railway-station examples so far put together into a short game called "Age of Steam".
 - ⬇ Example 119: **★★ Meteoric I and II** A meteor in the night sky which is visible from many rooms, so needs to be a backdrop, but which does not appear until 11:31 PM.
 - ⬇ Example 158: **★★★ Space Patrol - Stranded on Jupiter!** We'll be back in just a moment, with more exciting adventures of the... Space Patrol!
 - ⬇ Example 163: **★★★★ Entrevaux** Organizing the game by scenes, where each scene has a location and prop lists so that it can be set up automatically.
-

§4.5. Flashbacks

The viewpoint character may often need to remember events long past. The easiest way to do this is with a cut-scene, in which at some relevant point we pause the story and print a long passage of text describing the memory. Because large amounts of text can be hard for the player to take in, we may want to include some pauses in the presentation of this material; this facility is provided by the Basic Screen Effects extension by Emily Short, and might work something like this:

[Include Basic Screen Effects by Emily Short.](#)

Instead of examining the photograph for the first time:
say "This reminds you of the summer of '69...";
wait for any key;
say "... flashback content...";
wait for any key.

The "pause the game" phrase in the same extension offers a more dramatic pause that also clears the screen before printing new text.

Cut-scenes are easy to implement but should be used sparingly, since players often get impatient with long uninteractive passages. A slightly more deluxe implementation might insert an interactive scene that simply happens to be set in the past, before going on with another scene set "now"; and, indeed, some IF abandons the idea of "now" entirely, presenting pieces in a non-chronological order and letting the player work out how the sequence works together.

The most challenging case to implement (though still not very hard) is the one where we remove the player from one scenario, let him play through a flashback with past possessions and clothing, and then restore him to the same situation he left, with all of the same possessions and clothing. **Pine 3** shows how to do this: the code to change the player's status is isolated at the end of the example, and might fruitfully be reused. **Pine 4** expands on the same idea by adding another flashback scene, demonstrating one that can be visited repeatedly and one that can be seen only once.

★ *See **Scene Changes** for more uses of stripping and restoring the player*

★ See **Background** for other ways of introducing information that the player character already knows

★ See **Alternate Default Messages** for comments on how to change the tense of an interactive scene

⬆ Start of Chapter 4: Time and Plot

⬅ Back to §4.4. Scene Changes

➡ Onward to §4.6. Plot Management

⬇ Example 165: **★★★★ Pine 3** Pine: Allowing the player to visit aspects of the past in memory and describe these events to the princess, as a break from the marriage-proposal scene.

⬇ Example 167: **★★★★ Pine 4** Pine: Adding a flashback scene that, instead of repeating endlessly, repeats only until the Princess has understood the point.

§4.6. Plot Management

A plot manager (sometimes called a drama manager) is a piece of the program whose job it is to plan out events so that, whatever the player does, the story advances and an interesting narrative results. The plot manager might, for instance, decide that the player has wandered around for too many scenes without making any progress, and might compensate by making something happen that gives him a new hint on his current problem. It might trigger characters to act when it thinks the story should be reaching a crisis point. It might introduce new complications when it determines that the player is running out of problems to solve.

This is a theoretically challenging field. Sophisticated plot management requires that the story make difficult guesses, such as whether the player is "stuck" and what the player is working on right now. The advantage of using such a system is that (done very well) it makes the story extremely responsive to the player's behavior, which means that he is a real agent in the unwinding of the plot. It also contributes to the replayability, since trying the story a second or third time will produce quite different outcomes. But it is procedurally difficult to design a good plot management system and it requires a huge amount of content, as well: in order for the plot manager to give the player hints, change the course of events to suit his focus, and so on, the story has to have available many, many more scenes than will ever occur in any single playing.

Fate Steps In is only a *very* brief sketch in this direction, one in which the "fate" entity is trying to accomplish an end goal and, every turn, looks for ways to push the story towards that conclusion, whatever the player does.

★ See **Goal-Seeking Characters** for alternate ways to make characters act on their own

-  Start of Chapter 4: Time and Plot
 -  Back to §4.5. Flashbacks
 -  Onward to Chapter 5: The Viewpoint Character: §5.1. The Human Body
 -  Example 209:  **Fate Steps In** Fate entity which attempts to make things happen, by hook or by crook, including taking preliminary actions to set the player up a bit.
-

Examples from Chapter 4: Time and Plot

-  Start of this chapter
-  Chapter 5: The Viewpoint Character
-  Indexes of the examples

142

Example **Situation Room**

WI

Printing the time of day in 24-hour time, as in military situations.

Though Inform normally prints times in AM/PM terms, it stores the hours and minutes as 24-hour time; so, if we like, we can easily extract that information again thus:

"Situation Room"

The Situation Room is a room.

To say (relevant time - a time) as 24h time:
let H be the hours part of relevant time;
let M be the minutes part of relevant time;
say "[if H is less than 10]0[end if][H][if M is less than 10]0[end if][M]".

When play begins:
now the time of day is 6:09 PM;
now the right hand status line is "[time of day as 24h time]".

Test me with "z".

377

Example **The Big Sainsbury's**

WI

Making implicit takes add a minute to the clock, just as though the player had typed TAKE THING explicitly.

Implicit takes are a convenience to players; in general, we would like to avoid asking players to type any more obvious commands than strictly necessary, while allowing the computer to guess as much as it safely can.

Occasionally, though, we have designed a timed puzzle in which the player has a limited number of moves in which to accomplish his objectives. In that case, the

implicit take complicates matters, because it means that a player who types

>EAT GATEAU
(first taking the gateau...)

gets away with a spare move compared to the precise but naïf dupe who types

>TAKE GATEAU
>EAT GATEAU

...and really, that doesn't seem quite fair. The way to fix this problem is to fill in the extra minute on the clock during the implicit take; and that is indeed what we do in the following example.

"The Big Sainsbury's"

Sainsbury's is a room.

The crispy duck and the Guinness steak pie are edible things in Sainsbury's.

Rule for implicitly taking something:
follow the advance time rule;
continue the activity.

When play begins:
now the right hand status line is "[time of day]".

Test me with "take crispy duck / eat crispy duck / eat steak pie".

394

Example Uptempo

WI

Adjust time advancement so the game clock moves fifteen minutes each turn.

Suppose a game in which all actions take a very long time. Here's a simple implementation:

"Uptempo"

The fast time rule is listed instead of the advance time rule in the turn sequence rules.

This is the fast time rule:
increment the turn count;
increase the time of day by 15 minutes.

When play begins: now the right hand status line is "[time of day]".

The Temporal Hot Spot is a room.

Test me with "z / z".

This works fine as it stands, but we may run into some difficulty with it if we add scheduled events:

At 9:30 AM:
say "Two turtles run by, almost too fast to see."

At 9:37 AM:
say "A snail blitzes past."

At 9:42 AM:
say "The grass grows."

At 9:50 AM:
say "Several flowers burst open."

Time is counted forward after the schedule has already been consulted, so that only the 9:30 AM event happens between 9:30 and 9:45; the next two appear to occur between 9:45 and 10:00 AM, and the 9:50 AM event is not reported until the 10:00 AM to 10:15 wait. To get around this, we might schedule events only on the fifteen-minute mark when we want them to occur. Alternatively, we might try instead

"Uptempo"

The fast time rule is listed before the timed events rule in the turn sequence rules.

The advance time rule is not listed in the turn sequence rules.

This is the fast time rule:
increment the turn count;
increase the time of day by 15 minutes.

When play begins: now the right hand status line is "[time of day]".

The Temporal Hot Spot is a room.

At 9:30 AM:
say "Two turtles run by, almost too fast to see."

At 9:37 AM:
say "A snail blitzes past."

At 9:42 AM:
say "The grass grows."

At 9:50 AM:
say "Several flowers burst open."

Test me with "z / z / z / z".

This time our revised time-advancing rule is listed just before the event scheduler, not just afterwards.

409

Example Timeless

WI

A set of actions which do not take any game time at all.

In a game with tight timing, it is sometimes friendliest to the player to let him LOOK and EXAMINE as much as necessary without being penalized.

"Timeless"

Examining something is acting fast. Looking is acting fast.

Now we need a rule which, just at the right moment, stops the turn sequence rulebook in the cast of our new fast-acting actions:

The take visual actions out of world rule is listed before the every turn stage rule in the turn sequence rules.

This is the take visual actions out of world rule: if acting fast, rule succeeds.

Thus the rest of the turn sequence rulebook is omitted for looking or examining: in effect, they become out-of-world actions like "saving the game". If we wanted to add, say, taking inventory to the list of instant activities, we would just need to define it as acting fast, too.

Now the scenario for testing:

When play begins:

say "You are cornered by a pack of zombie wolves, armed only with a torch and a pair of pinking shears. This may be your last moment on earth, unless you can think fast!"

Cleft is a room. "You're backed into a cleft in the granite: behind you are only steep, high faces of stone, and before you a narrow passage."

The plural of zombie wolf is zombie wolves. A zombie wolf is a kind of animal. Four zombie wolves are in Cleft.

Rule for writing a paragraph about zombie wolves:

say "The good news is that there isn't much space in which for the zombie wolves to attack.";

now every zombie wolf is mentioned.

A steep high face of stone is scenery in Cleft. Understand "rock" as the stone. The description is "Now that you look more closely, there appear to be pitons driven into the rock."

Some pitons are part of the stone. The description of the pitons is "It looks as though someone else has made this ascent before."

Instead of climbing the stone, try going up. Instead of climbing the pitons, try going up.

Above the Cleft is Clifftop.

Every turn when the location is Cleft:
say "Alas, your time has run out. The alpha wolf springs--";
end the story.

Every turn when the location is Clifftop:
say "After a breathless climb, you emerge at last onto the open clifftop.";
end the story finally.

Test me with "x me / x stone / x pitons / climb pitons".

410



Example Endurance

WI

Giving different actions a range of durations using a time allotment rulebook.

Here we move to a systematic way of giving different durations to different actions, including even variations on the same act -- so that for instance climbing a steep hill might take several minutes more than other going actions. We do this by setting a number, "work duration", to represent the number of minutes consumed by a given action, and then consulting a rulebook to find out how long the past turn's action should take. By default, an action will take 1 minute.

We'll start by emulating the behavior of "Uptempo": each turn we'll set the clock forward most of the way, then check to see what has changed since the last turn, print any relevant events, and only then set the clock forward the final minute. The exception is when an action is set to take no time at all; in that case, we'll skip the rest of the turn sequence rules entirely.

"Endurance"

Work duration is a number that varies.

Every turn:
now work duration is 0;
increment the turn count;
follow the time allotment rules;
if work duration is 0, rule succeeds;
increase the time of day by (work duration minutes - 1 minute).

The time allotment rules are a rulebook.

A time allotment rule for examining or looking:
now work duration is 0;
rule succeeds.

A time allotment rule for going:
now work duration is 2;
rule succeeds.

A time allotment rule for going up:
now work duration is 5;
rule succeeds.

A time allotment rule for waiting:
now work duration is 10;
rule succeeds.

The last time allotment rule:
now work duration is 1.

When play begins: now the right hand status line is "[time of day]".

The Quai is a room. "An attractive park at the edge of the river Aude: here you can wander among palm trees, and watch cyclists go by on the bike path; in the water there are ducks. In the cafe to your north, patrons sip their pastis; and above you is the medieval walled city and its castle."

The Cafe is north of the Quai. "A charming collection of umbrella-shaded tables, from which one can watch the river and the walls of the city beyond. The noise of traffic is only a minor distraction."

The City is above the Quai.

After going to the City:
say "You struggle uphill for some distance...";
continue the action.

At 9:15 AM:
say "The bells ring out from Place Carnot."

Test me with "z / n / s / u".

183



Example The Hang of Thursdays

WI

Turns take a quarter day each, and the game rotates through the days of the week.

"The Hang of Thursdays"

The Stage is a room. Rule for printing the name of the stage: say "[current weekday] [current time period]" instead.

A weekday is a kind of value. The weekdays are Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday. The current weekday is a weekday that varies. The current weekday is Saturday.

A time period is a kind of value. The time periods are morning, afternoon, evening, night. The current time period is a time period that varies. The current time period is afternoon.

This is the new advance time rule:

if the current time period is less than night:

now the current time period is the time period after the current time period;

otherwise:

now the current time period is morning;

now the current weekday is the weekday after the current weekday.

Now we need to borrow from a later chapter to make these instructions apply to the passage of time:

The new advance time rule is listed instead of the advance time rule in the turn sequence rules.

Test me with "z / z / z / z / z".

260



Example Zqlran Era 8

WI

Creating an alternative system of time for our game, using new units.

Suppose that our game takes place on an alien planet that does not follow Earth time. On this planet, we want to track time with different units. We also want time to advance in those units, and we want to be able to set a schedule of timed events.

"Zqlran Era 8"

The Barren Lavender Surface of Zql is a room. "It is late twilight on Zql. Overhead, two crescent moons, both green, mark the sluggish passage of time. A cold wind is blowing over the pale purplish ground cover, but it does not penetrate your airtight suit."

A Zqlran date is a kind of value. 14-88 specifies a Zqlran date with parts zqls and frbs. Current zqlran date is a zqlran date that varies. The current zqlran date is 8-22. Previous zqlran date is a zqlran date that varies. The previous zqlran date is 8-20.

When play begins:

now left hand status line is "[current zqlran date], or [current zqlran date in words]".

To say (Zqlra - a Zqlran date) in words:

say "[zqls part of Zqlra] Z, [frbs part of Zqlra] f."

Inform automatically supplies a way to say a new unit, which will look similar to the format in which we defined that unit in the first place. But we can (as shown here) create our own alternative say phrases to express the units in other ways as well.

Next, we need to meddle with time advancement so that time is tracked in Zqlran date rather than in minutes. This requires borrowing a trick from a later chapter, to replace Inform's built-in time handling with an alternative time handling rule of our own:

The Zqlran time rule is listed instead of the advance time rule in the turn sequence rules.

This is the Zqlran time rule:
increment turn count;
now the previous zqlran date is current zqlran date;
increase the current zqlran date by 0-02;
repeat through the Table of Zql Schedule:
if era entry is greater than previous zqlran date and era entry is not greater than current zqlran date:
say event entry;
say paragraph break;
blank out the whole row.

Table of Zql Schedule

```
era event
8-24 "A wisp-thin cloud blows rapidly across the face of Nepenthe, the lesser of the two green moons."
8-28 "The cloud across Nepenthe clears."
```

Note that we could if we wished use a different device for scheduling events: this one simply prints text at scheduled eras, but we might also (for instance) make the event entry be a rule for Inform to follow, and tell Inform to carry out that rule at the scheduled time.

162

★ Example The Prague Job

WI

Scenes used to provide pacing while the player goes through his possessions.

Suppose we want to remind the player that he doesn't have all the time in the world, by starting to nag him when he's nearly, but not entirely, done going over his inventory in preparation for a job.

"The Prague Job"

A thing can be seen or unseen. A thing is usually unseen. Carry out examining:
now the noun is seen.

The player carries a lockpick, a smoke bomb, a grappling hook, and a pair of gloves. The description of the lockpick is "Effective on most kinds of key locks, it is a gift from your mentor in the discipline, old Wheezy." The description of the smoke bomb is "Your last of these, so you should rely on it only when other modes of escape have vanished. It takes effect when dropped, producing a cloud of purple haze sufficient to fill a medium-sized room." The description of the grappling hook is "Good for shooting at balconies and other sorts of

overhang." The gloves are wearable. The description of the gloves is "Black and shiny, with gripping material on the palms. Batman would be jealous."

The Toilet is a room. "The walls are painted an unattractive green; the fixtures are a bit old. But it is the only place in the hostel with any privacy." The Long Hallway is outside from the Toilet.

Reviewing Possessions is a scene. Reviewing Possessions begins when play begins.

Escalating Danger is a scene. Reviewing Possessions ends when Escalating Danger begins. Escalating Danger begins when preparations near completion.

To decide whether preparations near completion:

if at least two of the things which are carried by the player are seen, yes;
no.

When Escalating Danger begins: say "Someone pounds on the door of your hideout and yells at you in Czech."

Instead of going from the Toilet during Reviewing Possessions: say "You need to go over your equipment first, and make sure you're ready here."

Instead of going from the Toilet during Escalating Danger: say "You're not done checking over your materials."

Instead of waiting during Escalating Danger: say "There's no time to waste."

Every turn during Escalating Danger: if the time since Escalating Danger began is greater than 1 minute, say "Impatient footsteps pass your door again."

Escalating Danger ends when every thing which is carried by the player is seen. When Escalating Danger ends, say "There -- nothing damaged or torn. You're ready to go."

Mission is a scene. Mission begins when the player is in the Long Hallway. When Mission begins: end the story saying "The game is afoot"

Test me with "i / x lockpick / out / x bomb / out / x hook / x gloves / out".

155

★★ Example Entrapment

WI

A scene in which the player is allowed to explore as much as he likes, but another character strolls in as soon as he has gotten himself into an awkward or embarrassing situation.

The power of scenes lies in their ability to watch for general conditions and move the narrative along whenever these are fulfilled. Instead of waiting for the player to do one specific thing, the game waits for the world to be in a certain condition, before moving to the next stage of the plot.

For instance, suppose we have a story in which the player has been captured for doing something inappropriate at court and is brought in to await a meeting with a palace official. We want to give the player a few minutes to stew, and we want the scene to end with him doing something mildly peculiar or embarrassing, and the official catching him in the act. So we tempt him into trying any of a number of different kooky activities, and just wait until he falls into the trap...

"Entrapment"

Waiting Suite is a room. "You find yourself in a narrow room, more cozy than is really comfortable, with dark paneling on all the walls. Underfoot is a thick carpet the color of dried blood. The head of a dragon kit is mounted on the wall."

The wood paneling is scenery in the Waiting Suite. The description is "Just the sort of ornate panels that might conceal a carved switch. You've heard all sorts of rumors about secret rooms and passages in the palace, some of which have not been opened in centuries because no one remembers how to get at them." Understand "panels" or "panel" or "panelling" as the paneling.

Instead of switching on the paneling, say "First you'll have to locate any switches or catches with a careful search."

The thick carpet is scenery in the Waiting Suite. Understand "red" or "blood" or "rug" as the carpet. The description is "A dull, unwelcoming weave, only a touch redder than the wood around you. You discern that it does not lie perfectly flat."

Instead of touching the paneling for the first time: say "You run your hands over the paneling with a methodical touch, knowing exactly what you're looking for but never quite feeling anything that gives or twists; then thump lightly, looking for hollow spaces."

Instead of touching the paneling for the second time: say "With increased vigor, you run your fingers along the borders between panels, then smack each panel sharply at the center. No luck yet, but if you keep at it, you're bound to turn up anything that's there to find."

Instead of attacking the paneling: try touching the paneling. Instead of searching the paneling: try touching the paneling. Understand "knock on [something]" or "tap [something]" or "tap on [something]" as attacking.

After touching the paneling when the player is not confident:

say "Having polished off all the panels within easy reach, you now have to contort yourself around furniture here and crawl along the floorboards there, hitting each panel three times quite solidly before moving on."; now the player is embarrassed.

Instead of looking under the carpet for the first time:

say "You take a corner of the carpet and tug. The floor is sticky, so it doesn't come up on the first try."

A small table is an enterable supporter in the Waiting Suite. On the table is a copy of Dragon Pursuit Today. The description of Dragon Pursuit Today is "Full of glossy illustrations of dragons in various stages of capture, captivity, and destruction. The back of the magazine contains small black-and-white

advertisements for hunting kits and the like." Some advertisements and some illustrations are part of Dragon Pursuit Today. The description of the illustrations is "You have the misfortune to look first at the photographs accompanying 'Cleaning Dragon Splanchna', and feel quite unwell." The description of the advertisements is "Mostly terse ads and phone numbers."

After looking under the table:

say "It's quite a low table and you have to get down on your knees and poke your head underneath in order to get a good look."

After looking under the carpet:

say "You pull again at the carpet. There is a tug, then a tearing, as the ancient fabric struggles against the fabric glue. Some of the carpet winds up in your hand and some of it remains in patchy threads adhering to the floor."

After entering the table:

say "You climb onto the small table, noticing belatedly that you are leaving muddy footprints on its polished surface. Oh well: you can wipe them away again when you get down."

The dragon head is scenery in the Waiting Suite. Understand "kit" or "mouth" as the dragon head. The description is "Its eyes are wide with bewildered surprise; its mouth gapes, its forked tongue protrudes indignantly. From down here it looks as though there's something shiny stuck in its mouth, though you can't tell for sure." The head contains a shiny thing. The description of the shiny thing is "Intriguing but impossible to see clearly." Instead of taking the shiny thing, try searching the dragon head.

Before searching the dragon head:

if the player is not on the table, try entering the table;
if the player is not on the table, stop the action.

After searching the dragon head: say "You have a good look inside the dragon's mouth. There's a ball of lucite inside, propping the jaw in display position."

A person can be confident, nervous, or embarrassed. The player is confident.

Touching the paneling is embarrassing behavior. Looking under the carpet is embarrassing behavior. Entering the table is embarrassing behavior. Looking under the table is embarrassing behavior.

Instead of embarrassing behavior:

if the player is nervous, now the player is embarrassed;
if the player is confident:
say "Before you can act, you hear movement from the inner office. You freeze, not quite ready to be discovered in this situation. But no one comes out, and you begin to breathe more easily.";
now the player is nervous;
otherwise:
continue the action.

Causing trouble is a scene. Causing trouble begins when play begins. Causing trouble ends when the player is embarrassed. When Causing trouble ends: say "Just at this inopportune moment, you hear a throat being cleared behind you.

"We can see you now within," says a dry voice."; end the story saying "To be continued..."

Test me with "switch paneling / touch paneling / g / g / g".

Test more with "x dragon / x shiny / search head / g".

...and this scene might lead to another, and so on.

The purpose of an open-ended scene like this might be puzzly or narrative: we might be waiting for the player to get a puzzle solved, or we might be waiting for him to fulfil some plot condition that must be met before we can go on.

159



Example **Bowler Hats and Baby Geese**

WI

Creating a category of scenes that restrict the player's behavior.

Scenes can have properties -- a fact that is very useful when it comes to writing a series of scenes that all need to act alike in some respect.

Suppose we have a plot that features a number of scripted scenes, where we need the player to stand still and wait while the events of the scene play out. One way to set this up is to create a property for such scenes -- let's call them "restricted" -- and then write a rule that keeps the player in place while the scene happens:

"Bowler Hats and Baby Geese"

Section 1 - The Procedure

A scene can be restricted or free.

Instead of going somewhere during a restricted scene:

say "Better to stay here for the moment and find out what is going to happen next."

And now let's set up our restricted scene. In it, a clown is going to turn up wherever the player is (it doesn't matter where on the map he's gotten to at this point) and do a performance; the player will not be able to leave the area until the performance completes. We'll start with the setting:

Section 2 - The Stage and Props

The Broad Lawn is a room. "A sort of fun fair has been set up on this broad lawn, with the House as a backdrop: it's an attempt to give local children something to do during the bank holiday. In typical fashion, everyone is doing a very good job of ignoring the House itself, despite its swarthy roofline and dozens of blacked-out windows."

The House is scenery in the Broad Lawn. The description is "A cautious vagueness about the nature of the inhabitants is generally considered a good

idea. They might be gods, or minor demons, or they might be aliens from space, or possibly they are embodiments of physical principles, or expressions of universal human experience, or... at any rate they can run time backward and forward so it warbles like an old cassette. And they're always about when somebody dies. Other than that, they're very good neighbors and no one has a word to say against."

Instead of entering the House:

say "You can't go in, of course. It's not a house for people."

The Gazebo is north of the Broad Lawn. "The gazebo is sometimes used for bands, but at the moment has been appropriated for the distribution of lemonade."

The clown is a man. "A clown wearing [a list of things worn by the clown] stands nearby." The description is "He winks back at you."

The clown wears a purple polka-dot bowler hat. He carries a supply of baby geese. The description of the supply of baby geese is "Three or four. Or five. It's hard to count." Understand "goose" or "gosling" or "goslings" as the supply of baby geese.

There are some eggs. The description of the eggs is "A blur, really."

There is a Spanish omelet. The description of the Spanish omelet is "Exquisitely prepared."

...And now the scene itself:

Section 3 - The Scenes

The Clown Performance is a restricted scene. Clown Performance begins when the turn count is 3.

When Clown Performance begins:
move the clown to the location.

Every turn during Clown Performance:
repeat through the Table of Clowning:
say "[event description entry][paragraph break]";
blank out the whole row;
stop.

When Clown Performance ends:
now the eggs are nowhere;
now the clown carries the omelet.

Clown Performance ends when the number of filled rows in the Table of Clowning is 0.

Table of Clowning
event description

"A clown with a purple polka-dot bowler hat strides into the vicinity and begins to juggle baby geese."

"While the clown juggles, the baby geese visibly grow older and larger. The clown becomes unnerved."

"In an attempt to resolve the problem, the clown reverses the direction of his juggling. The geese revert to goslings."

"The goslings become smaller and smaller until the clown is juggling goose eggs[replace eggs]."

"The clown throws all the eggs into the air at once and catches them in the bowler hat. He takes a bow; the audience applauds. As a final gesture, he upends his hat to release a perfectly cooked omelet."

To say replace eggs:

now the supply of baby geese is nowhere;
now the clown carries the eggs.

Free Time is a scene. Free Time begins when Clown Performance Ends.

Test me with "scenes / n / z / z / look / x geese / s / x geese / x eggs / z / s".

160



Example Day One

WI

A scene which plays through a series of events in order, then ends when the list of events is exhausted.

"Day One"

Lecture is a scene. Lecture begins when play begins.

Every turn during Lecture:

repeat through Table of Lecture Events:
say "[event entry][paragraph break]";
blank out the whole row;
rule succeeds.

Here we use a table (see subsequent chapters) to keep track of all the events we wish to have occur during the course of the scene.

Table of Lecture Events

event

"'Welcome to Precolumbian Archaeology 101,' thunders Dr Freitag from the front of the class. 'Miss-- yes, you in the back. If you can't find a free seat, how are you going to find Atlantis? Sit down or leave. Now. Thank you.'"

"Freitag stands behinds his desk and lines up the pile of books there more neatly. 'It has come to my attention over previous years that there are two sorts of person who enroll in my class,' he says.

'Some of you will be members of the swim team or women's lacrosse players who have a distribution requirement to fulfill and are under the mistaken impression that archaeology must be easier than psychology. If that description applies to you, I advise you to drop the class now rather than at the midterm break. Under absolutely no circumstances will I ever sign a withdrawal form for someone who is crying at the time. Make a note of that, please.'"

"'The second sort of person,' Dr Freitag says, getting another wind. 'Yes, the

second sort of person takes this class because she imagines that it is going to lead to adventure or possibly to new age encounters with dolphins.'

His eye moves over the class, lingering an especially long time on a girl in a patchwork skirt.

'You should also leave now, but since you are probably lying to yourself about the reasons you're here, you will probably not heed my warning and we will be doomed to a semester of one another's company nonetheless.'

'''Whatever you may tell yourself, you are not here to gain a deeper understanding of the world or get in touch with yourself or experience another culture.'

He paces before the first row of desks, hammering on them one at a time. 'I know you probably wrote an admissions statement saying that that is what you hoped to do. Well, too bad. It is not inconceivable that some of you, somehow, will muddle towards a deeper understanding of something thanks to this class, but I am not holding my breath, and neither should you.'

'Freitag takes a breath. 'No, my dear freshmen, what you are here to do is learn facts. FACTS. Facts are unpopular in this university and, I am unhappily aware, at most of the institutions of inferior preparation from which you have come. Nonetheless, facts it will be. I will expect you to learn names. I will expect you to learn dates. I will expect you to study maps and I will expect you to produce evidence of exacting geographical knowledge on the exams. I will expect you to learn shapes of pottery and memorize masonry designs. There are no principles you can learn which are more important or more useful than a truly colossal bank of facts right there in your own head.'

'''I do not ever want to hear that you do not need to learn things because you will be able to look them up. This is the greatest fallacy of your computer-semi-literate generation, that you can get anything out of Google if you need it. Not only is this demonstrably false, but it overlooks something phenomenally important: you only know to look for something if you already know it EXISTS. In short there is no way to fake knowledge, and I am not going to pretend there is.' He smiles in lupine fashion.

'This class is likely to be the most miserable experience of your four years in university. Clear?'

'Everyone is silent.'

'The lecture is interrupted by the shrill of a bell.'

And then we define the scene so that it ends when the table runs out.

Lecture ends when the number of filled rows in the Table of Lecture Events is 0.

One advantage of this is that we can then edit the events in the scene by changing just the table; the scene will always run the right length and end on the turn when the last event occurs.

And to add a few additional details:

Instead of doing something other than waiting, looking, listening or examining during Lecture:

say "Dr Freitag glares at you so fiercely that you are frozen into inaction."

Notice the careful phrasing of "doing something other than..." so that we do not mention the objects; if we had written "something other than listening to something...", the instead rule would match only action patterns which involved a noun. We state the rule more generally so that it will also match nounless commands such as JUMP and SING, since Freitag will probably take a dim view of those as well.

When Lecture ends:

now Freitag is nowhere;
say "There is a flurry of movement as your fellow students begin to put away their books. Dr Freitag makes his way to the door and is gone before anyone can ask him anything."

The Classroom is a room. Dr Freitag is a man in the Classroom. "Dr Freitag paces before the blackboard."

Test me with "listen / x dr / x me / jump / z / z / z / z / z / x dr".

141



Example IPA

WI

Shops which each have opening and closing hours, so that it is impossible to go in at the wrong times, and the player is kicked out if he overstays his welcome.

Suppose we wanted a game set in a living town, with locations opening and closing at different times of day, and business carrying on as usual. The point might be to force the player to plan his itinerary carefully to hit the right spots at the right times; or we might be writing a more contemplative piece, where part of the enjoyment came from just watching the characters wander around doing their daily business...

"IPA"

When play begins: now the right hand status line is "[time of day]".

The time of day is 9:50 AM.

A shop is a kind of room. A shop has a time called the opening hour. The opening hour of the shop is usually 8 AM. A shop has a time called the closing hour. A shop usually has closing hour 6 PM.

Check going to a shop (called the target):

if the time of day is before the opening hour of the target,
say "[The target] is locked up until [the opening hour of the target]." instead.

Check going to a shop (called the target):

if the time of day is after the closing hour of the target,
say "[The target] has been closed since [the closing hour of the target]." instead.

Every turn when the location is a shop:
let deadline be the closing hour of the location;
if the deadline is before the time of day:
let target be a random adjacent room which is not a shop;
say "You are gently but firmly ushered out, since [the location] is closing.";
move the player to the target.

The Strip-mall Parking Lot is a room. "Dead Christmas trees are heaped outside the bagel shop. Strips of dirty ice survive along the curb, and in the shadows of the lamp-posts. A wet, almost illegible sheet of algebra homework is plastered to the asphalt.

Pinewood Brewing Supply is at the east end of the lot."

Pinewood Brewing Supply is a shop. It is east of Parking Lot. The opening hour of Pinewood Brewing Supply is 10:00 AM. The closing hour of Pinewood Brewing Supply is 3:30 PM. "Shelves and shelves of malt and hops; large glass carboys, and plastic tubing; empty bottles; bottle-caps; bottle-labeling kits; starters for vinegar, sourdough, root beer.

A sweet malty smell hangs in the air."

Instead of going to Brewing Supply when the time of day is before the opening hour of Brewing Supply for the second time:
say "You rattle at the door again. 'Hold your horses, for crying out loud,' yells a voice from within."

Noah's Bagels is a shop. It is north of the Parking Lot. The opening hour of Noah's Bagels is 6:00 AM. The closing hour of Noah's Bagels is 11:00 AM. "The selection has been somewhat picked over, leaving you with your choice of Pumpnickel, Asiago, or Everything."

Test me with "e / e / n / z / s / e / z / e / z / z / e".



Allowing the player to make an appointment, which is then kept.

Here we allow the player to set the time at which some future event is going to happen, rather than letting the game decide. We'll need to borrow the syntax for defining new actions from a later chapter:

"Hour of the Wren"

When play begins:

say "You more or less stumble across them in Central Park: a disparate group of people, all of different ages, sitting in a circle. They aren't talking to one another -- in fact, they seem to be trying very hard to ignore one another, like people in the waiting room of an especially embarrassing kind of doctor. You are about to go around when a woman in a grey pressed suit comes up to you. Her suit-skirt is trimmed in lavender cord, and she looks as though she might have been extremely sharp-dressed in 1944. She hands you a card."

The Circle in the Grass is a room. "No one is looking at you, except for the secretary, if that is what she is."

The player carries a card. The description of the card is "Typed: 'Active astrology - dislike your fortunes? change your stars! - make an appointment now - hour of the wren STILL AVAILABLE.'".

The time of day is 1:55 PM.

Understand "pick [time]" or "choose [time]" or "make appointment for [time]" or "make an appointment for [time]" as making an appointment for. Making an appointment for is an action applying to one time.

Carry out making an appointment for:
say "Fate cannot be commanded more than once."

Instead of making an appointment for the time understood for the first time:
say "You settle on [the time understood] for your appointment. The woman makes a note of it in an appointment book, which she carries in a brown paper bag. 'Excellent choice, ma'am,' she says in a low, urgent voice. 'You'll be very satisfied.'";
stars shift at the time understood.

Understand "hour of the wren" as 2:00 PM.

At the time when stars shift:
end the story saying "insert cataclysm here".

Test me with "x card / make appointment for hour of the wren / z / z / z / z".

354



Example Air Conditioning is Standard

WI

Uses "writing a paragraph about" to make person and object descriptions that vary considerably depending on what else is going on in the room, including some randomized NPC interactions with objects or with each other.

"Air Conditioning is Standard"

Section 1 - The Garage

A person has some text called current occupation. The current occupation of a person is usually "None".

Mood is a kind of value. The moods are bemused, bored, attentive, rapt, and blushing. A person has a mood. A person is usually attentive.

Instead of examining a person:
now every thing is unmentioned;
carry out the writing a paragraph about activity with the noun.

Rule for writing a paragraph about a person (called X):
 let the subsequent mention be "Name";
 if the current occupation of X is not "None":
 say "[current occupation of X]. ";
 let the subsequent mention be "He";
 if X is female, let the subsequent mention be "She";
 if X wears something unmentioned:
 if the subsequent mention is "Name", say "[The X] ";
 otherwise say "[subsequent mention] ";
 say "is wearing [a list of unmentioned things worn by X]";
 if X carries something unmentioned, say " and carrying [a list of
 unmentioned things carried by X]";
 say ".";
 otherwise:
 if X carries something unmentioned:
 if the subsequent mention is "Name", say "[The X] ";
 otherwise say "[subsequent mention] ";
 say " is carrying [a list of unmentioned things carried by X]."

The Garage is a room. "Above the street door is a spectacular art nouveau fanlight, wherein a stained-glass Spirit of Progress bestows the gift of Transportation on mankind.

The sun, gleaming through the hair of Progress, throws amber curls on the macadam floor."

The fanlight is scenery in the Garage. The description is "A semi-circle of stained glass as wide as the garage door, designed by Louis Comfort Tiffany himself. No expense has been spared."

The gift of Transportation is part of the fanlight. The description is "The gift of Transportation is envisioned as a cornucopia disgorging a steam locomotive. And that blue bit of glass might be the Montgolfier balloon."

The Spirit of Progress is part of the fanlight. The description is "It is part of her character to have bare shoulders like that."

The machinist is a bored man. He is in the Garage. He is wearing a grimy pair of overalls. He carries a wrench and a screwdriver. The current occupation of the machinist is "[The machinist] is making some adjustments to [the random thing which is part of the Victorian Car] with his [random thing carried by the machinist]"

The Victorian Car is a device in the Garage. A cast-iron steering wheel, a leather bucket seat, a horn, and a combustion engine are part of the Victorian Car. The seat is an enterable supporter.

Rule for writing a paragraph about a device (called X):
 let the subsequent mention be "Name";
 if the X is unmentioned:
 say "[The X] is here. ";
 let the subsequent mention be "It";
 if something is part of X:
 if the subsequent mention is "Name", say "[The X] ";
 otherwise say "[subsequent mention] ";
 say "[if a mentioned thing is part of X]also [end if]features[if a mentioned

thing is part of X], in addition to [the list of mentioned things which are part of X],
[end if] [a list of unmentioned things which are part of X]";
say ".".

Rule for printing the name of the steering wheel while writing a paragraph about
a person:

say "steering wheel".

A supporter has some text called position. The position of a supporter is usually
"None".

The Office is west of the Garage. The Office contains a desk. The desk has the
position "A [desk] with several dozen drawers stands in the center of the room".
On the desk are some papers.

After printing the name of a supporter (called X) which supports an unmentioned
thing:

now X is unmentioned.

Rule for writing a paragraph about a supporter (called X):

let the subsequent mention be "Name";

if the position of X is not "None":

say "[position of X]. ";

let the subsequent mention be "It";

if a mentioned thing is on X:

say "Besides [the list of mentioned things which are on X], ";

let the subsequent mention be "it";

if the subsequent mention is "Name", say "[The X] ";

otherwise say "[subsequent mention] ";

say "holds [a list of unmentioned things which are on X]."

Section 2 - Schedule

The time of day is 4:38 PM.

At 4:42 PM:

move the machinist to the Office;

say "The machinist wanders into the Office to get some paperwork.";

now every thing carried by the machinist is on the desk;

now the current occupation of the machinist is "[The machinist] rifles through
[the papers] on [the desk]".

At 4:43 PM:

move the young lady to the Garage;

if the young lady can be seen by the player,

say "An attractive young lady walks in from the street, and glances around
as though she has never been here before."

At 4:45 PM:

if the young lady can be seen by the player,

say "With a not-quite-convincing air of innocence, [the young lady] happens
to lean upon [the horn], which bleats loudly.";

otherwise say "There is a honk from the Garage[if the machinist can be seen
by the player]. The machinist looks up with a frown[end if].";

now the horn is mentioned.

At 4:46 PM:

move the machinist to the Garage;
say "The machinist strolls from the Office into the Garage to find out what is going on.";
now the current occupation of the machinist is "[The machinist] is chatting with [the young lady]. He seems to be demonstrating the various features of [the car], including [the random thing which is part of the car]";
now the current occupation of the young lady is "[The young lady] is asking [the machinist] a number of questions about [the car]".

At 4:49 PM:

if the young lady can be seen by the player, say "[The machinist] gives [the young lady] his arm to climb into [the seat].";
move the young lady to the seat;
now the young lady is rapt;
now the current occupation of the young lady is "[The young lady] is turning [the steering wheel] from side to side";
now the current occupation of the machinist is "[The machinist] is leaning on the door of [the car], pointing out features to [the young lady]";
move the besotted expression to the machinist;
now the machinist is wearing the besotted expression.

At 4:52 PM:

now the sober grey gown is unbuttoned at the neck;
if the young lady can be seen by the player, say "[The young lady] murmurs something about the wilting heat, and undoes a button or two of her gown. The machinist's expression is comical, or would be, if you weren't annoyed."

Every turn when the player is in the Garage and young lady is on the seat:
say "You are beginning to feel a little unnecessary in this scene."

Every turn when the player is in the Office and the young lady is on the seat:
say "There's no sound at all from the other room, not even conversation."

Before going to the Garage when the young lady is on the seat:
now the sober grey gown is tellingly dishevelled;
move the young lady to the Garage;
now the young lady is blushing;
say "There is a flurry of movement as you enter the room.";
now the current occupation of the young lady is "[The young lady] stands near the door, tapping her foot nervously";
now the besotted expression is nowhere;
now the current occupation of the machinist is "[The machinist] is leaning against [the car], looking smug".

Section 3 - Initially Out of Play

The besotted expression is a wearable thing. The description is "It looks foolish, doesn't it?"

The young lady is a bemused woman. She is wearing a sober grey gown and a pair of black boots. The current occupation of the young lady is "[The young lady] is running a gloved finger along the chassis of [the victorian car]"

Before printing the name of the young lady while writing a paragraph about a person:

say "[mood of the young lady] "

The description of the grey gown is "Something about the perfect row of tiny buttons has the wrong effect -- at any rate, it is natural to wonder how long they take to undo." The gown can be buttoned almost to the chin, unbuttoned at the neck, or tellingly dishevelled.

Rule for printing the name of the gown when writing a paragraph about a person:

say "sober grey gown ([sober grey gown condition])"

Test me with "z / look / look / z / look / west / east / z / look / z / look / z / look / west / east".

156

★ Example Age of Steam

WI

The railway-station examples so far put together into a short game called "Age of Steam".

The following source is very short and simple, yet it already feels surprisingly interesting in play, because something is going on which the player does not control but must observe. The single scene both starts and finishes.

"Age of Steam"

The Station is a room. "Eynforme Halt is a raised platform fringed with cowslip: a whistle-stop with no more than a signal and a water-tank."

The Flying Scotsman is fixed in place. "The Flying Scotsman, fastest train in the world, is now at a dead standstill."

Train Stop is a scene. Train Stop begins when the player is in the Station for the third turn. Train Stop ends when the time since Train Stop began is 3 minutes.

When Train Stop begins:

now the Flying Scotsman is in the Station;

say "The Flying Scotsman pulls up at the platform, to a billow of steam and hammering."

When Train Stop ends:

now the Flying Scotsman is nowhere;

say "The Flying Scotsman inches away, with a squeal of released brakes, gathering speed invincibly until it disappears around the hill. All is abruptly still once more."

Instead of entering the Flying Scotsman, say "Alas, the [time when Train Stop began] arrival is only to take on water, not to set down or pick up."

Test me with "z / z / z / enter flying scotsman / z / z".



Example Meteoric I and II

A meteor in the night sky which is visible from many rooms, so needs to be a backdrop, but which does not appear until 11:31 PM.

The game below begins at half past eleven, and one turn later, it's meteor time:

"Meteoric I"

The time of day is 11:30 PM.

At 11:31 PM:

now the meteor is in the great outdoors;
say "A meteor streaks across the sky."

The great outdoors is a region. The Spanish Balcony is east of the Inner Court. The Court and Balcony are in the great outdoors. Inside from the Court is the Swimming Pool.

The meteor is a backdrop. Instead of doing something to the meteor, say "The meteor is no longer visible, now nothing more than a memory."

Test me with "wait / wait / examine meteor / west / examine meteor / in / examine meteor".

Or for something a little slower-moving and with no after-image:

"Meteoric II"

The time of day is 4:30 PM.

At 4:31 PM:

now Phobos is in the great outdoors;
say "Phobos rises from the western horizon."

At 10:06 PM:

now Phobos is nowhere;
say "Phobos sets over the eastern horizon."

The great outdoors is a region. The Martian Balcony is east of the Inner Court. The Court and Balcony are in the great outdoors. Inside from the Court is the Heavy Water Swimming Pool.

Phobos is a backdrop. Instead of doing something to Phobos, say "Phobos orbits a mere 6000km above you, which is practically touching range for astronomy. On the other hand, astronomy isn't all that practical."

Test me with "wait / wait / examine phobos / west / examine phobos / in / examine phobos".

Though we should not really use Earthly time-keeping, since the Martian day is about half an hour longer than ours.



Example Space Patrol - Stranded on Jupiter!

We'll be back in just a moment, with more exciting adventures of the...
Space Patrol!

American radio adventure series of the 1950s were unobtrusively sponsored by breakfast cereals, as the following modest example demonstrates. Note that the scene-changing for the commercial break needs to know nothing about the actual programme it breaks into: if Part I were replaced with a different Space Patrol episode, Part II need not be changed at all.

"Space Patrol #57 - 1953-10-31 - Stranded on Jupiter!"

Use scoring.

Part I - Serial

Red Spot is a room. "You are in the middle of a vast red oval plain. Overhead, the thick Jovian clouds swirl menacingly, and a fine acrid dust falls instead of rain." Some acrid dust is scenery in the Red Spot. The description of the dust is "The rust-colored dust coats every surface. You've no idea how deep it goes."

Instead of going in Red Spot, say "As you once told Cadet Lucky, Jupiter's a mighty big planet, maybe bigger than Iowa. Why, the Red Spot alone stretches out almost to the horizon."

The player wears a silver uniform and rubber boots. The player carries a shovel and an Analscope. The description of the Analscope is "As you recall from Space Patrol #9 - 1952-11-29 - The Electronic Burglar, the Analscope is a device for locating buried metals. That's what guided you all the way from the orbit of Uranus. (Oh, all right, Neptune.) If only you hadn't crashed!"

The metal plate is a fixed in place container. It is openable and closed. In the metal plate is some water. The description of the metal plate is "Stamped with the distinctive logo of the previous mission."

Instead of examining the player, say "Your hair clumps together stickily, thanks to the dustfall."

Digging is an action applying to one thing. Understand "dig [something]" or "dig in [something]" as digging.

Instead of digging the dust, try looking under the dust. Instead of looking under the dust when the metal plate is not visible: move the metal plate to the location; say "You brush aside the dust underfoot and -- what were the odds? -- it turns out that you landed just where the previous landing party did, thirteen ill-fated years ago. Here is the metal plate that covers their original well.

But wait! Called by the clanging of your shovel on the plate, a band of Jovian pterodactyls swoop down to attack! You're totally defenceless! You don't have a hope! You're absolutely finished!"; increase the score by 10; move K-Klak to Red Spot.

K-Klak the Pterodactyl is an animal. "K-Klak, leader of the Jupiter Pterodactyls, menaces you. A terrifying creature of scaly wings, with a dragon's tail, K-Klak stands... about 1/8th of an inch tall." Instead of doing something to K-Klak, say "K-Klak makes a frankly panicky noise and leaps backwards, out of your way."

After opening the metal plate: increase the score by 10; say "You have found water! You're saved! K-Klak makes a (very cautiously) pleased noise. Now to find the stolen Brainograph, and track down the crook with the thick Jewish accent and his henchmen with their thick Polish accents..."; end the story finally.

The maximum score is 20.

Part II - Cereal

When play begins, say "Instant Ralstons and Regular Ralstons, the hot whole-wheat cereals in the red and white checkerboard packages present... SPACE PATROL... High adventure in the wild vast reaches of space... Missions of daring in the name of interplanetary justice... Travel into the future as Buzz Corey, Commander-in-Chief of the..."

Last score is a number that varies. Every turn: now the last score is the score.

Ralstons Ad is a scene. Ralstons Ad begins when score is not the last score. Ralstons Ad ends when the Ricechex is consumed.

Include Basic Screen Effects by Emily Short.

When Ralstons Ad begins:

- center "*** We'll be back in just a moment! ***";
- pause the game;
- strip the player;
- move the player to the Kitchen.

When Ralstons Ad ends:

- center "*** And now, back to today's exciting adventure ***";
- pause the game;
- restore the player.

Saved location is a room that varies. Locker is a container. Wardrobe is a container.

To strip the player:

- now every thing carried by the player is in the locker;
- now every thing worn by the player is in the wardrobe;
- now saved location is location.

To restore the player:

- now every thing carried by the player is in the Kitchen;
- now every thing in the locker is carried by the player;
- now every thing in the wardrobe is worn by the player;
- move the player to saved location.

The Space Patrol Kitchen is a room. "The nerve center of the Space Patrol! This is where cadets fill up with their SUPER-FUEL. North leads to the astro control room, while back south is the cargo hold." A breakfast bowl is in the Kitchen. In

the bowl is Ricechex. Ricechex is edible. The Ricechex can be consumed or uneaten. The Ricechex is uneaten.

Instead of going north in Kitchen: say "[refusal to leave]". Instead of going south in Kitchen: say "[refusal to leave]".

Instead of examining the player when Ralstons Ad is happening: say "You are currently being played by a generically attractive person of about 30, with very good teeth and well-kept nails."

After eating the Ricechex: say "That's right folks, always start your day the SPACE PATROL way with a tasty bowl of Ricechex, Wheatchex or good hot Ralstons. Mmmm Mmmm. You just can't get enough of the sugary goodness in Ricechex, Wheatchex and good hot Ralstons."; now the ricechex is consumed.

Instead of tasting the Ricechex:
say "Wow! *wolf-whistle* Man oh man oh man! Yumm-y!"

To say refusal to leave:
repeat through Table of Refusals:
say "[nope entry][paragraph break]";
blank out the whole row;
rule succeeds;
say "You can't. Eat your Ricechex."

Table of Refusals

nope

"You can't go that way in the limited universe of this sponsored message."

"Or that way."

"You've already tried that!"

"Why would you want to walk away when you have an alluring bowl of Ricechex right here?"

Test me with "n / i / x me / x dust / dig dust".

Test ad with "n / s / n / s / n / i / x me / get bowl / taste ricechex / eat ricechex".

Test ending with "x plate / x k-klak / open plate".

Episode 57 of "Space Patrol" was actually called "Iron Eaters Of Planet X", just in case the reader feels that any of the foregoing unfairly traduces a work of thoughtful science fiction.



Organizing the game by scenes, where each scene has a location and prop lists so that it can be set up automatically.

For some games, it makes sense to organize the entire game around scenes rather than around locations, moving the player when a new scene begins and laying out new props.

To this end, we might extend Inform's default handling of scenes so that each scene has properties to indicate prop lists and locations, and move objects in and out of play automatically as the scenes change. For instance:

"Entrevaux"

Part 1 - Procedure

A scene has a room called the starting location.

A scene has a list of objects called the scenery props.

A scene has a list of objects called the inventory props.

The starting location is the room to which the player should be moved; scenery props are things that need to be put there when the scene begins; inventory props, things that are given to the player when the scene begins; and the description some printed text to introduce the new scene. We may still occasionally need to have recourse to special "When the Dancing-Lesson begins..." rules for individual scenes, but for the most part this allows us to set scenes up in a consistent and predictable way.

Another point that might be slightly less obvious: sometimes we want to announce a change of location to the player when the scene starts, and sometimes we don't. In particular any scene that starts "when play begins" should probably not explicitly describe the entered room, since that would duplicate the description automatically produced on the first turn of play. So we add a property to track whether any given scene should be announcing its location:

A scene can be location-silent or location-loud.

And let's say that we also want to announce each new scene as another "chapter" of the game in play, with a pause before the scene begins.

Here we include "Basic Screen Effects" because it will allow us to pause the game for a keypress, then clear the screen before each new chapter:

Include Basic Screen Effects by Emily Short.

The chapter counter is a number that varies.

First when a scene (called the current scene) which is not the Entire Game begins:

```
if chapter counter is greater than 0:  
    pause the game;  
    increment chapter counter;  
    say "[bold type]Chapter [chapter counter]: [current scene][roman type]";
```

Last when a scene (called the current scene) which is not the Entire Game begins:

```
repeat with item running through the scenery props of the current scene:  
    move the item to the starting location of the current scene;  
repeat with item running through the inventory props of the current scene:
```

move the item to the player;
if the location is not the starting location of the current scene:
if the current scene is location-loud:
move the player to the starting location of the current scene;
otherwise:
move the player to the starting location of the current scene, without
printing a room description.

At the end of each scene, we strike the set and remove all the loose objects from play.

When a scene (called the current scene) ends:
repeat with item running through things which are not fixed in place:
if the item is not the player:
now the item is nowhere.

Part 2 - Scenario

Entrevaux Station is a room. "The station building consists of a waiting room and a ticket-selling office so small that only one person can buy a ticket at a time. On the outside wall is a clock that runs twelve minutes late; but since the trains also run twelve minutes, give or take, behind their published schedule, this clock is helpful in establishing reasonable expectations. [paragraph break]Painted on the door is the logo of the Chemin de Fer de Provence, the only railway in France that is not part of the SNCF."

The Hillside Tower is a room. "It's very dark in here, lacking artificial lighting, but from the rough rectangular window you can see a slice of hillside and a little of the river Var."

The window is scenery in the Hillside Tower. The description is "Through it can be seen a slice of wooded hillside and exposed grey-brown cliff. You are in the southern French foothills of the Alps, and the territory is dry. The only respite is the river Var, a milky blue at this time of year, running shallowly over mud and large stones far below your window." Understand "view" or "slice of hillside" or "hillside" or "hill" or "river" or "var" or "mud" or "stones" or "large stones" as the window.

A used ticket is a thing. The description is "A piece of receipt paper indicating that you have paid the one-way fare of 9 euros from Nice. There is a hole punched through it."

A one-euro coin is a thing. The description is "It's a bimetal coin, brassy around the rim and silvery in the center. One side shows western Europe, with unusual prominence given to the UK, and the other side Leonardo da Vinci's four-armed, four-legged man having a nice stretch. It's dated 2002."

Some re-enactors are a person. "Milling about one end of the station is a crowd of medieval re-enactors." The description is "They're dressed in a somewhat aimless range of styles roughly honoring the period of 900-1500 AD. One gentleman is wearing a knobby leather cap; which is a good thing, because there is a rooster standing on his head." Understand "men" or "man" or "gentleman" or "rooster" or "reenactors" or "crowd" or "medieval" or "woman" or "women" as the re-enactors.

A kidnapper is a person. "Your kidnapper is watching you from the corner with his arms folded. You have the impression he's just marking time until someone more important arrives." The description is "He does not look at all like the kidnapping sort, but more like a sommelier at a superior restaurant: he wears a black pinstriped suit and has nicely-manicured hands."

The trolley is an enterable fixed in place container. "The 'train' on which you arrived is really just a single car, more like a trolley than a proper train." Understand "car" or "train" as the trolley. The description is "It has a glass front, so you can see ahead while riding: an innovation among trains."

Arrival is a location-silent scene. "After many days['] journey, you have arrived at last in Entrevaux, a walled medieval town now chiefly of interest to tourists and crusade re-enactors."

The starting location of Arrival is the Entrevaux Station.

The scenery props of Arrival are { re-enactors, trolley }.

The inventory props of Arrival are { the used ticket, one-euro coin }.

Arrival begins when play begins. Arrival ends when the time since Arrival began is 2 minutes.

Abduction is a location-loud scene. "You check into the Hotel Vauban and sleep deeply enough; it was a long and sticky trip to get here."

Then in the middle of the night something confusing happens. You have the impression of strangers in your room, and then a searing pain, and you don't come back to yourself until midmorning of the following day..."

The starting location of Abduction is the Hillside Tower.

The scenery props of Abduction are { kidnapper }.

Abduction begins when Arrival ends.

Test me with "i / x re-enactors / z / z / i / x him".

165



Example Pine 3

WI

Pine: Allowing the player to visit aspects of the past in memory and describe these events to the princess, as a break from the marriage-proposal scene.

"Pine"

Part 1 - The Set-up

This is mostly a repeat of what we have already seen, but for the sake of producing a playable scenario, we include it. The new material appears at Part 2.

A person can be asleep or awake. A person can be active or passive.

The Spinning Tower is a room. "A remote corner of the old castle, reserved for spinning and weaving tasks."

Sleeping Beauty is an asleep woman in the Spinning Tower. "[if asleep]Sleeping Beauty lies here, oblivious to your presence[otherwise]Sleeping Beauty stands beside you, looking [attitude][end if]." The description is "She is even more magnificent than the rumors suggested." Understand "woman" or "girl" or "princess" or "lady" as Sleeping Beauty.

Discovery is a scene. Discovery begins when play begins. Discovery ends when Sleeping Beauty is awake. Marriage Proposal is a scene. Marriage Proposal begins when Discovery ends.

When Discovery ends: say "Throughout the palace you can hear the other sounds of stirring and movement as the spell of centuries is broken."; now Beauty is passive.

Instead of waking an awake person: say "Redundant."

Instead of waking an asleep person: say "Yes, but how?"

Instead of attacking an asleep person:

now the noun is awake;

say "[The noun] sits bolt upright. 'Hey! Ow!' So much for that true love's kiss nonsense."

Instead of kissing an asleep person:

now the noun is awake;

say "[The noun] slowly stirs to wakefulness!"

Instead of throwing water at an asleep person:

now the second noun is awake;

now the noun is nowhere;

say "You pour out [the noun] on [the second noun]."

[The second noun] wakes, shuddering. 'Agh! I had a terrible dream about drowning and then-- Hey!'"

The player carries a jug of water. Understand "pour [something] on [something]" or "splash [something] at/on [something]" as throwing it at.

Table of Conversation

topic	reply	quip
"dream/dreams/nightmare/nightmares/sleep"	"Sleep well?"	you ask solicitously.
'Not really,' she replies, edging away from you. So much for that angle."	"Ghastly nightmares,' she remarks. You nod politely."	
"marriage/love/wedding/boyfriend/beau/lover"	"So,' you say. 'This is a little weird since we just met, but, um. Would you like to get married?'"	
She looks at you nervously. 'Do I have to?'"	"I, er,' she says. 'I hope I'm not supposed to marry you or something.'" "	
"marriage/love/wedding/boyfriend/beau/lover"	"I was told I was going to marry you and inherit the kingdom,' you say, apologetically. 'Would that be very bad?' This could be awkward, considering your family circumstances -- you did promise your mother that everything would be better, after this --	

'Oh, it's not you -- I'm seeing someone,' she says, smiling quickly.

You try to think how to point out that it's been a hundred years since she last saw her boyfriend. "Do you think I could go look for someone? I'm seeing him, you see, and I think I've been... sick... for a while, so he might be worried."

You try to think how to point out that it's been a hundred years since she last saw her boyfriend. And try not to think how awkward things would be in your family if she refuses to marry you."

"marriage/love/wedding/boyfriend/beau/lover" "You've been up here for a hundred years,' you say. An unpleasant thought occurs to you. 'Was your young man in the castle somewhere?'"

She shakes her head mutely." "She goes to the window and looks out at the now-fading thicket of briar. 'That took a while to grow,' she observes. 'I've been up here longer than I thought.'"

You shrug, uncomfortable."

Instead of asking an awake beauty about a topic listed in the Table of Conversation:

now Beauty is passive;
say "[reply entry][paragraph break]";
blank out the whole row.

Instead of telling an awake beauty about something: try asking the noun about it.

Instead of asking an asleep person about something:
say "[The noun] snores."

Marriage Proposal ends when the number of filled rows in the Table of Conversation is 0.

Every turn during Marriage Proposal:
if Beauty is active and Beauty is visible:
repeat through Table of Conversation:
say "[quip entry][paragraph break]";
blank out the whole row;
make no decision.

Every turn: now Beauty is active.

When Marriage Proposal ends: end the story saying "This is going to take some explaining."

So far we haven't much of a chance to affect matters and make them better. Suppose we'd like to add an element to the conversation where we're allowed to tell Beauty about past events -- and explore them a bit; and if the first retelling doesn't go quite as planned, we're allowed to revisit these scenes to hit them with a bit more emphasis.

Part 2 - Flashbacks

Instead of asking an awake beauty about a topic listed in the Table of Flashback Material:

now Beauty is passive;
say "[reply entry][paragraph break]".

A fact is a kind of thing. The family circumstances is a fact. A fact can be known or unknown. A fact can be current or past.

Once known, a fact remains known permanently -- this could be useful if we wanted to make some rules about how Beauty acts when she knows different information. By contrast, a fact is only "current" if it is the last thing discussed. Since a player can mention a fact over and over, he can make it "current" again and again, and thus reactivate the flashback.

Table of Flashback Material

topic	reply
"poor/poverty/family/money/mother/circumstances" or "family circumstances" or "my family/mother"	"[if family circumstances is unknown]I wish you'd give some thought to marrying me. You see,' you say, your jaw tensing. 'I wouldn't ask if it weren't for my [family circumstances]...' [otherwise]I don't think you fully understand the [family circumstances],' you say.[end if]"

After printing the name of a fact (called target): now the target is current; now the target is known.

This "After printing the name..." rule will be explained later in the chapter on activities; for now, it is enough to know that whenever family circumstances is mentioned in the table of flashback material, this rule will automatically be called. Now the terms under which the flashback happens:

Poverty flashback is a recurring scene. Poverty flashback begins when family circumstances is current. When poverty flashback begins: strip the player; move the player to the Woodcutter's Shack.

Note the "recurring" here: we want the player to be able to revisit this scene as needed.

The Woodcutter's Shack is a room. "Your family lives in a shack in the forest. There are holes in the roof, and in the winter the snow comes in -- rain, too, for that matter. The walls aren't very well-boarded, and don't keep out the wind, and even though you live in the middle of dense woods, you can never gather enough fuel to keep this place fully heated. And then there's the stench. Pigs wander freely in and out, and your three youngest brothers play on the floor."

Pigs are an animal in the shack. The pigs are scenery. The description is "They really are very grubby, dirty animals, but what's worse than that, the value of pigs has declined a lot over the last few decades. This is hard to explain to someone who has been out of touch with the world for a while, but keeping pigs for meat is a dubious prospect when there's less and less for them to forage on." Instead of smelling the pigs: say "They smell the way animals do, when they live among their own refuse."; increase the pity of Beauty by 2.

The brothers are a man. The description of brothers is "Hans, Franz, and Lukas. Twins and then the baby. So young, and growing up fatherless; and soon to be orphaned entirely, if your mother's health does not improve." Understand "brother" or "twin" or "twins" or "baby" or "franz" or "hans" or "lukas" as the brothers.

The untidy bed is scenery in the Shack. Mother is a woman on the untidy bed. The description of mother is "She is wasting away of a slow disease, her skin stretched tautly over bone. She hasn't been the same since your father left." On the bed is a folded letter.

The description of the letter is "Many times read over and creased, the letter explains how your father has gone away with a wealthy countess and will not return. Your mother was not able to read it herself, of course, and had to have it explained to her by the parish priest. Now she keeps it by the bed and crumples it in her fits of delirium."

Instead of kissing or touching Mother for the first time:

say "You place a gentle kiss on her feverish brow. She looks up at you, her oldest -- yes, never mind that bit -- with a look of sincere trust and admiration.

'You'll find a way through this for us,' she says, squeezing your fingers. 'I know you will.'"; increase the pity of Beauty by 3.

Instead of kissing Mother: say "You have no more heart-rending memories of affection to recount; that one incident will have to serve, for rhetorical purposes."

Instead of waiting in the Shack: say "The wind blows sharply through the walls."

Instead of attacking someone in the Shack:

say "Though sometimes the conditions of your life make you grouchy and impatient, you would never dream of striking a member of your own family. But from time to time you do feel the temptation."

Beauty has a number called pity. After examining something in the Woodcutter's Shack, increment the pity of Beauty. After examining mother, increase the pity of Beauty by 2. After examining the letter, increase the pity of Beauty by 3.

Poverty flashback ends when waiting or the time since poverty flashback began is five minutes.

When Poverty flashback ends:

now family circumstances is past;
say "...you finish describing the miserable circumstances of your home life, and allow your attention to return to the present.";
restore the player;
now Beauty is passive;
if Beauty is sympathetic, say "'Oh dear!' she says. 'What a dreadful life!' She wrings her hands. 'No wonder you are eager to improve your lot...! But --' Her brow clears, a new thought occurring. 'You needn't marry me, you know! We could arrange it differently! I am certain that my father would give you a large reward, instead, and then I would not be separated from my current boyfriend!'" ;
otherwise say "She makes a disgusted face, but she doesn't seem nearly so heart-wrung as you had hoped to make her. Tough audience, these modern princesses."

Definition: Beauty is sympathetic if the pity of Beauty is greater than 4.

To say attitude:

if Beauty is sympathetic, say "distressed on your behalf";
otherwise say "a little confused".

And the following is the same as in the Space Patrol example as well: we need a systematic way to remove the player's possessions, then put everything back when the flashback is over:

Saved location is a room that varies. Locker is a container. Wardrobe is a container.

To strip the player:

now every thing carried by the player is in the locker;
now every thing worn by the player is in the wardrobe;
now saved location is location.

To restore the player:

now every thing carried by the player is in the location;
now every thing in the locker is carried by the player;
now every thing in the wardrobe is worn by the player;
move the player to saved location.

Test me with "x beauty / wake beauty / pour water on beauty / ask beauty about sleep / tell beauty about poverty / smell pigs / x mother / x letter / kiss mother / ask beauty about marriage / z / z".

Because we haven't changed the endings of the "Marriage Proposal" scene, there is still only one way for this scenario to work out; but at least now the player has the opportunity to alter Beauty's attitude a bit (or not) before the game is done.

167



Example Pine 4

WI

Pine: Adding a flashback scene that, instead of repeating endlessly, repeats only until the Princess has understood the point.

Suppose in addition to our pathetic little family history, we have another secret to convey to the Princess, this one a little more peculiar. She either gets it or she doesn't; once she gets it, we do not revisit that flashback, though it is still possible to keep visiting the poverty flashback.

"Pine"

Part 1 - The Set-up

A person can be asleep or awake. A person can be active or passive.

The Spinning Tower is a room. "A remote corner of the old castle, reserved for spinning and weaving tasks."

Sleeping Beauty is an asleep woman in the Spinning Tower. "[if asleep]Sleeping Beauty lies here, oblivious to your presence[otherwise]Sleeping Beauty stands beside you, looking [attitude][end if]." The description is "She is even more magnificent than the rumors suggested." Understand "woman" or "girl" or "princess" or "lady" as Sleeping Beauty.

Discovery is a scene. Discovery begins when play begins. Discovery ends when Sleeping Beauty is awake. Marriage Proposal is a scene. Marriage Proposal begins when Discovery ends.

When Discovery ends: say "Throughout the palace you can hear the other sounds of stirring and movement as the spell of centuries is broken."; now Beauty is passive.

Instead of waking an awake person: say "Redundant."

Instead of waking an asleep person: say "Yes, but how?"

Instead of waiting in the presence of an asleep person (called snorer): say "You are alone with the sound of [the snorer] snoring sonorously."

Instead of attacking an asleep person:
now the noun is awake;
say "[The noun] sits bolt upright. 'Hey! Ow!' So much for that true love's kiss nonsense."

Instead of kissing an asleep person:
now the noun is awake;
say "[The noun] slowly stirs to wakefulness!"

Instead of throwing water at an asleep person:
now the second noun is awake;
now the noun is nowhere;
say "You pour out [the noun] on [the second noun]."

[The second noun] wakes, shuddering. 'Agh! I had a terrible dream about drowning and then-- Hey!'"

The player carries a jug of water. Understand "pour [something] on [something]" or "splash [something] at/on [something]" as throwing it at.

Table of Conversation

topic	reply	quip
"dream/dreams/nightmare/nightmares/sleep"	"Sleep well?"	you ask solicitously.
'Not really,' she replies, edging away from you. So much for that angle."		"'Ghastly nightmares,' she remarks. You nod politely."
"marriage/love/wedding/boyfriend/beau/lover"		"'So,' you say. 'This is a little weird since we just met, but, um. Would you like to get married?'"

She looks at you nervously. 'Do I have to? I mean, I'd rather not.'

Well, this could get prickly fast."		"'I, er,' she says. 'I hope I'm not supposed to marry you or something.' Uh oh."
"marriage/love/wedding/boyfriend/beau/lover"		"'I was told I was going to marry you and inherit the kingdom,' you say, apologetically. 'Would that be very bad?' This could be awkward, considering your family circumstances -- you did promise your mother that everything would be better, after this --"

'Oh, it's not you -- I'm seeing someone,' she says, smiling quickly.

You try to think how to point out that it's been a "'Do you think I could go look for someone? I'm seeing him, you

hundred years since she last saw her boyfriend."

see, and I think I've been... sick... for a while, so he might be worried.'

You try to think how to point out that it's been a hundred years since she last saw her boyfriend. And try not to think how awkward things would be in your family if she refuses to marry you."

"marriage/love/wedding/boyfriend/beau/lover" ""Do you think you could consider alternatives if he's no longer interested in you?' you suggest.

She gives you the look of a wounded squirrel. 'My father might not approve of my love for the kitchen boy, but his heart is faithful and true!' she exclaims.

'Right; supposing that he's still around, I'm sure that his love won't have faded,' you say, considering your fingernails. Maybe you'd better come clean with her about your identity, after all: she might be more favorably inclined if she understood that you won't interfere in her base-born romances."

"I don't expect you to understand,' she says in a low whisper. 'I know it is not considered proper for a princess to love -- and such a one as my William, who works in the kitchen --' Her glance dares you to laugh. '-- but I cannot marry you without telling you this truth.'

Right then. Perhaps you'd better tell her your secret, in exchange?"

"marriage/love/wedding/boyfriend/beau/lover" ""You've been up here for a hundred years,' you say. An unpleasant thought occurs to you. 'Was your young man in the castle somewhere?'

She shakes her head mutely." "She goes to the window and looks out at the now-fading thicket of briar. 'That took a while to grow,' she observes. 'I've been up here longer than I thought.'

You shrug, uncomfortable."

Instead of asking an awake beauty about a topic listed in the Table of Conversation when Marriage Proposal is happening:

now Beauty is passive;
say "[reply entry][paragraph break]";
blank out the whole row.

Instead of telling an awake beauty about something: try asking the noun about it.

Instead of asking an asleep person about something:

say "[The noun] snores."

Marriage Proposal ends in failure when the number of filled rows in the Table of Conversation is 0.

Every turn during Marriage Proposal:

if Beauty is active and Beauty is visible:
repeat through Table of Conversation:
say "[quip entry][paragraph break]";
blank out the whole row;
make no decision.

When Marriage Proposal ends in failure: end the story saying "This is going to take some explaining."

Part 2 - Flashbacks

Instead of asking an awake beauty about a topic listed in the Table of Flashback Material:

now Beauty is passive;
say "[reply entry][paragraph break]".

A fact is a kind of thing. A fact can be known or unknown. A fact can be current or past.

The family circumstances is a fact. The secret identity is a fact. The printed name of secret identity is "secret".

Table of Flashback Material

topic	reply
"poor/poverty/family/money/mother/circumstances" or "family circumstances" or "my family/mother"	"[if family circumstances is unknown]'I wish you'd give some thought to marrying me. You see,' you say, your jaw tensing. 'I wouldn't ask if it weren't for my [family circumstances]...' [otherwise]'I don't think you fully understand the [family circumstances],' you say.[end if]"
"secret/identity/gender/girl/female/woman" or "secret identity" or "my secret" or "my secret identity" or "my gender"	"[if dramatic revelation ended in failure]'Look,' you say, trying again. 'Pay attention: I need you to understand my [secret identity].' [otherwise]'You clear your throat and allow your voice to stray upward, into its natural register and out of this husky false tenor you've been affecting. 'There's, er, something you should know about my [secret identity],' you say...' [end if]' [if dramatic revelation ended in success] She looks impatient. 'I get it, you know,' she says. 'I'm not stupid.'"

After printing the name of a fact (called target): now the target is current; now the target is known.

Poverty flashback is a recurring scene. Poverty flashback begins when family circumstances is current. When poverty flashback begins: strip the player; move the player to the Woodcutter's Shack.

The Woodcutter's Shack is a room. "Your family lives in a shack in the forest. There are holes in the roof, and in the winter the snow comes in -- rain, too, for that matter. The walls aren't very well-boarded, and don't keep out the wind, and even though you live in the middle of dense woods, you can never gather enough fuel to keep this place fully heated. And then there's the stench. Pigs wander freely in and out, and your three youngest brothers play on the floor."

Pigs are an animal in the shack. The pigs are scenery. The description is "They really are very grubby, dirty animals, but what's worse than that, the value of pigs has declined a lot over the last few decades. This is hard to explain to someone who has been out of touch with the world for a while, but keeping pigs for meat is a dubious prospect when there's less and less for them to forage on." Instead of smelling the pigs: say "They smell the way animals do, when they live among their own refuse."; increase the pity of Beauty by 2.

The brothers are a man in the shack. The brothers are scenery. The description of brothers is "Hans, Franz, and Lukas. Twins and then the baby. So young, and growing up fatherless; and soon to be orphaned entirely, if your mother's health does not improve." Understand "brother" or "twin" or "twins" or "baby" or "franz" or "hans" or "lukas" as the brothers.

The untidy bed is scenery in the Shack. Mother is a woman on the untidy bed. The description of mother is "She is wasting away of a slow disease, her skin

stretched tautly over bone. She hasn't been the same since your father left." On the bed is a folded letter.

The description of the letter is "Many times read over and creased, the letter explains how your father has gone away with a wealthy countess and will not return. Your mother was not able to read it herself, of course, and had to have it explained to her by the parish priest. Now she keeps it by the bed and crumples it in her fits of delirium."

Instead of kissing or touching Mother for the first time:

say "You place a gentle kiss on her feverish brow. She looks up at you, her oldest -- yes, never mind that bit -- with a look of sincere trust and admiration.

'You'll find a way through this for us,' she says, squeezing your fingers. 'I know you will.'"; increase the pity of Beauty by 3.

Instead of kissing Mother: say "You have no more heart-rending memories of affection to recount; that one incident will have to serve, for rhetorical purposes."

Instead of waiting in the Shack: say "The wind blows sharply through the walls."

Instead of attacking someone in the Shack:

say "Though sometimes the conditions of your life make you grouchy and impatient, you would never dream of striking a member of your own family. But from time to time you do feel the temptation."

Beauty has a number called pity. After examining something in the Woodcutter's Shack, increment the pity of Beauty. After examining mother, increase the pity of Beauty by 2. After examining the letter, increase the pity of Beauty by 3.

Poverty flashback ends when waiting or the time since poverty flashback began is five minutes.

When Poverty flashback ends:

now family circumstances is past;
say "...you finish describing the miserable circumstances of your home life, and allow your attention to return to the present.";
restore the player;
now Beauty is passive;
if Beauty is clever and Beauty is sympathetic:
say "'I understand,' she says slowly. 'Yes, I do. I'll do it.' She takes a deep breath and looks at you. 'We will be king together! and your family will be royalty!'";
end the story finally;
otherwise:
if Beauty is sympathetic, say "'Oh dear!' she says. 'What a dreadful life!' She wrings her hands. 'No wonder you are eager to improve your lot...! But --' Her brow clears, a new thought occurring. 'You needn't marry me, you know! We could arrange it differently! I am certain that my father would give you a large reward, instead, and then I would not be separated from my current boyfriend!'";
otherwise say "She makes a disgusted face, but she doesn't seem nearly so heart-wrung as you had hoped to make her. Tough audience, these modern princesses."

Definition: Beauty is sympathetic if the pity of Beauty is greater than 4.

To say attitude:

if Beauty is sympathetic, say "distressed on your behalf";
otherwise say "a little confused".

Saved location is a room that varies. Locker is a container. Wardrobe is a container.

To strip the player:

now every thing carried by the player is in the locker;
now every thing worn by the player is in the wardrobe;
now saved location is location.

To restore the player:

now every thing carried by the player is in the location;
now every thing in the locker is carried by the player;
now every thing in the wardrobe is worn by the player;
move the player to saved location.

Part 3 - The Other Secret

This time, we're waiting for the princess either to understand or not understand -- so we don't want to rerun the scene once it has happened successfully.

Beauty has a number called clue count.

Dramatic revelation is a recurring scene. Dramatic revelation begins when attempting confidence.

To decide whether attempting confidence:

if dramatic revelation ended in success, no;
if secret identity is current, yes;
no.

When dramatic revelation begins:

strip the player;
say "You reminisce about one of the many stops on the way here: you had a long journey from your homeland, and it wasn't made any easier by your poverty, the inability to afford decent inns or plentiful food or any kind of ride along the way.";
move the player to the Forest Clearing;
move the pack to the player; now the player wears the trousers; now the player wears the shirt.

Forest Clearing is a room. "It's mid-autumn in your memory, the pool clear and cold, gold and red leaves floating on the surface."

The pool is scenery in the Clearing. Understand "reflection" or "surface" or "water" as the pool. "The pool is cold but beautiful, and the stopping place a welcome rest." The leaves are scenery in the clearing. The description is "Bright gold and orange and red: it's been a sharply chilly autumn, as you have reason to know in detail."

The trousers and the shirt are wearable things. The pack is a container. The pack contains ale, food, and skirt. A distraction is a kind of thing. The ale, the food, the pair of trousers, and the shirt are distractions. The description of a

distraction is usually "[The item described] is not the point of this story." The shirt and the trousers are wearable. The description of the trousers is "Borrowed from your oldest brother, who is only a year younger than you. They are too long for your legs and overly snug at the hip, but no one around here pays much attention to fashion, and you're getting away with it, more or less." After examining the trousers, increment the clue count of Beauty.

Instead of examining the player during dramatic revelation:

- increment the clue count of Beauty;
- say "You cannot see yourself without reflection, but you can feel your hair loose and unbound over your shoulders."

Rule for printing the name of the skirt while taking inventory: say "one skirt you have not been able to bring yourself to part with". The description of the skirt is "Made for you by your mother, and it looks quite pretty on you. If your primary plan does not work, you may be forced to wear it again, and hope to catch a male eye... but with luck that will not be necessary." After taking inventory: increment clue count of Beauty.

Swimming is an action applying to nothing. Understand "swim" or "dive" as swimming.

Instead of swimming in the presence of the pool:

- increment clue count of Beauty;
- say "You consider going for a swim, but don't dare be caught unclad and unarmed, not here. There are too many men around, and any of them discovering you here would surely take advantage."

Instead of searching or drinking the pool:

- increment clue count of Beauty;
- say "You lean over the pool and look carefully at your reflection, your hair loose and unbound, falling around your face in waves. (That should surely give it away!)"

Instead of waiting during dramatic revelation: say "You wait for the penny to drop, for her to understand."

Dramatic revelation ends in failure when waiting or the time since dramatic revelation began is five minutes. When dramatic revelation ends in failure:

- now secret identity is past;
- restore the player;
- now Beauty is passive;
- say "She wrinkles her nose. 'I don't understand!' she says. 'What are you trying to tell me?'"

You could weep for womankind. But you don't quite dare spell it out in so many words, not when someone might come up the stair and overhear a chance revelation."

Dramatic revelation ends in success when Beauty is clever. When dramatic revelation ends in success:

- restore the player;
- now Beauty is passive;
- say "'You're -- a girl? Like me?'"

'Not much like you,' you say, glancing over her petite frame and pert nose. 'But female, at any rate.'

Definition: Beauty is clever if the clue count of Beauty is greater than 2.

And now, since we don't really want to return to the rest of the 'marriage proposal' scene once she has learned our ID:

Marriage proposal ends in distraction when Dramatic Revelation ends in success.

Compromise proposal is a scene. Compromise proposal begins when Dramatic Revelation ends in success. When Compromise Proposal begins: now Beauty is passive.

Instead of asking an awake beauty about a topic listed in the Table of Secondary Conversation when Compromise Proposal is happening:

now Beauty is passive;
say "[reply entry][paragraph break]";
blank out the whole row.

Every turn during Compromise Proposal:

if Beauty is active and Beauty is visible:
repeat through Table of Secondary Conversation:
say "[quip entry][paragraph break]";
if the number of filled rows in the Table of Secondary Conversation is greater than 1, blank out the whole row;
make no decision.

Every turn: now Beauty is active.

Notice that we moved the re-activation rule down here so that the Compromise Proposal rule would fire first. There are other more complicated ways of handling order of every turn rules than by relying on text sequence alone; but we will save that for a later chapter. For now it is sufficient to depend on the order in which the rules are declared.

Table of Secondary Conversation

topic	reply	quip
"girls/me/women/female/truth/identity"	"Marrying me would be no interference,' you go on. 'You could carry on whatever romances you wished, without your father noticing.' (Probably. You'll let the pragmatic details of this work themselves out later, and hope that any children she has will look vaguely like you.)"	"Girls can't rescue people."

'Wrong,' you say, feeling a little annoyed. 'But you see why marrying me wouldn't be an interference. You could carry on whatever romances you wished, without your father even noticing.'

"king/man" "If you're thinking that a woman can't be the prince -- and then king -- well, there was a woman Pope, once.'

She looks awed." "The crease in her forehead does not go away. 'But if everyone thinks you are a man... later you would be king!'

Before she can go on, you say, 'There was a woman Pope, once. Compared to that, a woman king is nothing.'

"decision/proposal/marriage/choice" ""So,' you say. 'What do you think?'

[final decision]" "Her pretty nose twitches again, which you are coming to recognize as a sign of hard mental labor. 'I think I see,' she says. [final decision]"

To say final decision:

if Beauty is sympathetic:

say "She considers the matter silently for some minutes, then says: 'I will do it. My beloved William will be so glad!' You imagine that William's feelings on the matter will be a tad more complex than that, but do not bother quashing her exuberance...";

end the story finally;

otherwise:

say "'I still don't quite understand why you would want this so badly as to go to all that trouble,' she admits uneasily. Evidently you have not explained enough to her about the poverty of your home life."

Test me with "x beauty / pour water on beauty / ask beauty about sleep / tell beauty about poverty / smell pigs / x mother / x letter / kiss mother / ask beauty about marriage / tell beauty about identity / x me / look in water / i / z / ask beauty about marriage".

209



Example Fate Steps In

WI

Fate entity which attempts to make things happen, by hook or by crook, including taking preliminary actions to set the player up a bit.

One of the nice things about before rules for actions is that they allow us to express some planning for characters other than the player: we've already seen how this works, a bit. But we could also use before rules to write plans for an abstract story-driving entity, rather than for other individual characters. This story-driver could be in charge of all the non-player characters, as well as spontaneous or natural changes in the environment, shaping the narrative around the player's behavior.

The following example is a very simple one, but the same concept could be worked out in a great deal more complexity, with all sorts of alternative procedures available to our story-manager:

"Fate Steps In"

Fate is a woman. After deciding the scope of the player: place Fate in scope. The description of Fate is "Not smiling." Instead of doing something other than examining to fate: say "As if."

Every turn: try fate tripping.

Tripping is an action applying to nothing.

Carry out someone tripping:

if something dangerous (called the trap) is in the location:
say "Lise chooses this moment to lick her fingers -- it's not gross, it's natural, you decide -- stand up, and head for the door. Unfortunately, her path crosses directly over [the trap]. There is a vaudevillesque moment where you try to warn or catch her; the next moment she's on the floor, looking shocked and also in quite a lot of pain. 'I'm not sure,' she says to you steadily but with unfocused gaze, 'but I think I might have broken my tailbone.'";
end the story saying "Well, she's paying attention to you now".

Before someone tripping when the location does not contain a dangerous thing: try the person asked making a mess instead.

Making a mess is an action applying to nothing.

Carry out someone making a mess:

let calamitous object be a random visible supporter which supports at least three things;
if calamitous object is a supporter:
say "[The calamitous object] tips over, spilling [the list of things on calamitous object] all over the place.";
move the calamitous object to the location;
now every thing on the calamitous object is in the location.

Definition: a thing is dangerous if it is not the carton and it is not the table and it is not a person.

Before someone making a mess when a safe supporter (called target) is visible: if Lise carries something, try Lise putting a random thing carried by Lise on the target instead.

Instead of someone making a mess when the tray is on the table:

say "Just at that moment, a large blond man-thing in a red jacket walks more or less through you, and you come into violent contact with the table, knocking [the list of the things on the table] onto the floor.";
now every thing on the table is in the location;
now every thing on the tray is in the location instead.

Definition: a supporter is safe if the number of things on it is less than two.

McQuerry Dining Hall is a room.

The table is scenery in the dining hall. The table is a supporter.

Lise Fitzwallace is a woman in the Dining Hall. "Lise is at the nearest table, not apparently paying any attention to you." The description of Lise is "A capella singer, women's rugby champion, general object of attention from all genders. Unlikely to notice you unless fate smiles broadly." Lise carries a fork, a napkin, an empty glass, and a plate of half-eaten eggplant parmesan.

Report Lise putting something on something:

say "Lise, still deep in thought, absently puts [the noun] on [the second noun]." instead.

The carrying capacity of the player is 2. The carton of chocolate milk is in the Hall. "There's a carton of milk beside you, which you set down for a moment -- but you do want it."

Instead of taking something when the player carries the tray:
say "You've got both hands full with this tray."

The player carries the tray. On the tray is some macaroni and some overdone chicken. The macaroni and the chicken are edible. The tray is portable.

Test me with "get milk / put tray on table / get milk".

Test again with "drop tray".

Chapter 5: The Viewpoint Character

§5.1. *The Human Body*; §5.2. *Traits Determined By the Player*; §5.3. *Characterization*;
§5.4. *Background*; §5.5. *Memory and Knowledge*; §5.6. *Viewpoint*

-  Contents of *The Inform Recipe Book*
-  Chapter 4: Time and Plot
-  Chapter 6: Commands
-  Indexes of the examples

§5.1. The Human Body

By default, Inform gives the player character (and every other person) a simple unitary body, one without hands or feet or any other defined parts. In many games this is adequate; but in others it is not enough, and we may want to endow all people with some more specific physical features, as in

A face is a kind of thing. A face is part of every person.

Once we've done this, we may invite ambiguities if the player types LOOK AT FACE; it is this challenge that is addressed in **The Night Before**.

rBGH gives the player a random height and then uses this to determine how the room should be described around him.

Slouching lets the player (and other characters as well) take different sitting, standing, and lying down positions.

-
-  Start of Chapter 5: The Viewpoint Character
 -  Back to Chapter 4: Time and Plot: §4.6. Plot Management
 -  Onward to §5.2. Traits Determined By the Player
 -  Example 254:  **rBGH** The player character's height is selected randomly at the start of play.
 -  Example 396:   **Slouching** A system of postures allowing the player and other characters to sit, stand, or lie down explicitly or implicitly on a variety of enterable supporters or containers, or in location.
 -  Example 59:    **The Night Before** Instructing Inform to prefer different interpretations of EXAMINE NOSE, depending on whether the player is alone, in company, or with Rudolph the Red-nosed Reindeer.
-

§5.2. Traits Determined By the Player

Some IF tries to make the viewpoint character more congenial to the player by allowing some customization.

Identity Theft demonstrates asking the player to supply the viewpoint character's name.

Pink or Blue demonstrates a way to let the player choose a gender at the start of play: this will mostly be interesting if the rest of the story makes some use of the player's choice. Since that example is written expressly to demonstrate included Inform 6 code, however, we may find it more congenial to generalize from the more flexible **Baritone, Bass**.

This is not the only way to go - as we'll see in the next section, there's also something to be said for making the viewpoint character a strongly distinct creature with well-defined preferences and attitudes.

-
-  Start of Chapter 5: The Viewpoint Character
 -  Back to §5.1. The Human Body
 -  Onward to §5.3. Characterization
 -  Example 416:  **Identity Theft** Allowing the player to enter a name to be used for the player character during the game.
 -  Example 309:  **Baritone, Bass** Letting the player pick a gender (or perhaps other characteristics) before starting play.
 -  Example 451:  **Pink or Blue** Asking the player to select a gender to begin play.
-

§5.3. Characterization

Much of the personality of the player character in IF emerges from what he can and cannot (or will and will not) do; part of the pleasure of playing a character arises from this opportunity for role-playing and role-exploration. Some characters are consciousnessless daredevils, willing to jump off cliffs, crawl through narrow gaps, and rob widows if the player commands it; others are repressed neurotics who barely dare to speak to other characters or touch anything that doesn't belong to them.

Finishing School and **Dearth and the Maiden** both treat the case of a character constrained by good manners and a sense of polite society: the former forbids only one action, while the latter condemns a whole range of them.

Constraining the character is only the half of it: we might also want to think about what sorts of unusual actions that character might be especially likely to take, and account for these. Of course, major actions that affect the story world will require some thought and implementation work, and we should consider carefully before making the player a character like, say, the Noble of Glamour, a spirit in human form who can charm all comers, transform bespectacled secretaries into divas, and cause spontaneous cloudbursts of scarlet glitter.

But even simple humans have some characteristic traits and gestures. We will probably want to write some characteristic reaction to EXAMINE ME, as demonstrated in **Bad Hair Day**. We might provide a few pieces of clothing or props that aren't strictly critical in the story, like a policeman's helmet or a feather boa:

The player is wearing a policeman's helmet.

We can liven up the interactive aspect of characterization if we give the player a little scope for role-playing: this may mean responding to gestures, like

Understand "bite nails" as a mistake ("Your only nail remaining is the one on your left thumb, and you're saving it for the AP Calculus exam.").

(Of course, we would need to have hinted to the player that nail-biting is characteristic of his character.)

★ See **Clothing** for more on dressing characters up

★ See **Saying Complicated Things** for conversation, another area in which the player character's personality might come into play

-
-  Start of Chapter 5: The Viewpoint Character
 -  Back to §5.2. Traits Determined By the Player
 -  Onward to §5.4. Background
 -  Example 73:  **Finishing School** The "another" adjective for rules such as "in the presence of another person".
 -  Example 85:  **Bad Hair Day** Change the player's appearance in response to EXAMINE ME.
 -  Example 109:  **Dearth and the Maiden** Our heroine, fallen among gentleman highwaymen, is restrained by her own modesty and seemliness.
-

§5.4. Background

In IF, as in all interactive storytelling, an essential problem is that the player does not begin the story knowing everything that the player character should, and so may implausibly bumble through situations that the player character should be quite comfortable in. If the player character has friends, an unusual job, a home or environment we're not familiar with, a secret past, these will all be a blank to the player.

Some games get around this by making the player character an amnesiac, or positioning him as a newcomer to a strange world in which his disorientation is explicable; but there are stories that cannot be told this way, and so we need other methods of getting the player to know what the player character already does.

Our first opportunity to inform the player about the player character is in the opening text of a story:

When play begins:
say "The funeral is exactly a month ago now, but Elise's shoes are still on the shoe tree."

We may also want to write descriptions of objects to give extra background information the first time the player encounters them:

A thing can be examined or unexamined. A thing is usually unexamined. After examining something: now the noun is examined; continue the action.

The description of the newspaper is "A rolled-up newspaper[if unexamined], and thus a symbol of your newly-single state: Elise always had it open and the Local Metro section next to your plate by the time you got out of the shower[end if]."

To expand on this, we could give the player a THINK ABOUT or REMEMBER command, with which he can call up information about people he meets or references he encounters in descriptions, so that he could (for instance) next type REMEMBER ELISE. **Merlin** demonstrates one way to implement a character with memory; **One of Those Mornings** puts a twist on this by letting the player FIND things which he knows his character possessed at some time before the story started.

-
-  Start of Chapter 5: The Viewpoint Character
 -  Back to §5.3. Characterization
 -  Onward to §5.5. Memory and Knowledge
 -  Example 277:  **Merlin** A REMEMBER command which accepts any text and looks up a response in a table of recollections.
 -  Example 301:  **One of Those Mornings** A FIND command that allows the player to find a lost object anywhere
-

§5.5. Memory and Knowledge

All of us carry around in our heads an (incomplete, imperfect) model of the world around us: an idea of where we left the keys, whether the oven is on or off, how many clean pairs of socks are left in the drawer, what we look like in our best pair of jeans. The differences between that mental model and reality are to some degree a reflection of personal character: our forgetfulness, our wishful thinking, our innocence or cynicism.

By default, Inform does not keep track of the player character's knowledge (or any other character's knowledge, for that matter) as a separate thing from the model world, relying on descriptive prose rather than modeling to introduce these quirks of characterization.

All the same, there are often times when we would like to keep track of discrepancies between the world model and the narrator's mental model. Perhaps the most common way to do this is simply to mark everything that the player encounters as "seen" when the player first examines it, thus:

A thing can be seen or unseen.

Carry out examining a thing:
now the noun is seen.

or -- to have things remembered from the first moment they're mentioned in a room description:

Rule for printing the name of something (called the target):
now the target is seen.

The mental model need not always be accurate, of course. We might, for instance, have occasion to keep track of where the player character last saw something, even if the object has since been moved; or keep track of falsehoods the player character has been told in conversation; or make the player refer to a character as "the bearded man" until he is properly introduced.

Included with Inform is the extension Epistemology, by Eric Eve, which provides one way of tracking this kind of information. Epistemology distinguishes between items that the player character has seen, because they're objects in a room the player has been to, and items that are familiar to the player for other reasons, such as a quest item he knows about but hasn't found yet, or an abstract conversation topic. Anything that is either seen or familiar is counted as "known".

Modeling what the player does and does not know is only half the job, of course: we also need that information to affect the behavior of the story in plausible ways.

One obvious occasion to use player character knowledge is in the output of descriptions. We might want to respond to actions differently depending on what the player has previously done, as in **Tense Boxing**, or change the way we describe objects in light of new knowledge about them, as in **Zero. Casino Banale** takes that idea much further, with a whole system of facts that can be narrated to the player in a somewhat flexible but interdependent order, as the player looks at relevant objects or notices them in room descriptions.

Along similar lines, we may want an object to change its name for the player depending on what the player knows. That name change should affect both what Inform displays and what it understands about the object. For instance:

An Amherz Amulet is a thing. It can be known or unknown. It is privately-named.

The printed name is "[if known]Amherz Amulet[otherwise]lizard-shaped pewter charm[end if]".

The description is "[if known]It's a unique and magically powerful pewter charm shaped like a lizard[otherwise]It's some cheap tacky pewter charm shaped like a lizard. At least, as far as you can tell -- it's pretty grubby[end if]."

Understand "amherz" or "amulet" as the Amulet when the Amulet is known.

Understand "lizard" or "lizard-shaped" or "pewter" or "charm" as the Amulet when the Amulet is unknown.

Instead of rubbing the amulet when the amulet is unknown:

say "You rub off a bit of the dirt, and... what do you know? It's actually the priceless and fabulously powerful Amherz Amulet!";

now the Amherz Amulet is known.

Finally, the player's knowledge may affect how the story interprets commands, in the determining what is called "scope". When Inform tries to make sense of something the player has typed, it makes a list of everything that the player is allowed to refer to at the

moment, and then checks whether all of the objects in the player's command refer to items in that list. Only things that are "in scope" are open for discussion.

If the player mentions an object that is not "in scope" -- say, a red hat left behind in the next room -- Inform will issue the response "You can't see any such thing." This is also Inform's reply if the player mentions a nonsense object ("EXAMINE FURSZWIGGLE") or an object that does not exist in the story world at all ("EXAMINE CELL PHONE" in a story set in Carolingian France).

This is not the only possible way for interactive fiction to handle such communication. Some games will respond differently to EXAMINE RED HAT and EXAMINE FURSZWIGGLE, saying in the first case something like "You can't see that now" and in the second "I don't know the word 'furszwiggle'."

The drawback of such behavior is that the player can make premature discoveries. If he hasn't found a sword yet, but thinks there may be a sword later in the story, he can type EXAMINE SWORD and see from the response whether his guess is correct. Nonetheless, there are people who prefer this alternative exactly because it does expose the limits of the story's understanding, preventing fruitless attempts to use a word that is not recognized at all. If it is desirable, there is an extension that will reproduce this behavior in Inform as well.

Using Inform's default behavior, however, scope is an ad-hoc way of keeping a list of things that are common knowledge between the story and the player. The player knows many things that the story might not (like what a cell phone is); the story knows a few things the player may not (like the fact that there is a sword in an as-yet unvisited room). Neither of those things can fruitfully enter into commands because they have no mutually agreed-upon referent.

By default, Inform assumes that "scope" includes only those things that are currently visible by line of sight. This works pretty well for a wide range of situations, but there are still plenty of occasions when we want to admit that the story and the player share a knowledge of things not seen. GO TO THE KITCHEN might be a useful command even when the player can't currently view the kitchen. ASK FRED ABOUT THE FOOTPRINTS should perhaps work even when the footprints are far away in the garden. SMELL STINKY CHEESE might need to work even when the cheese is invisibly locked away in a porous container but is exuding a stench. In a dark room, the player can't see his own inventory, but he should still remember that he's carrying it and be able to mention it. And sometimes we might want the story to acknowledge that the player is referring to an object that he has seen somewhere, even if that thing is now out of sight.

In practice, we have two ways to tinker with scope: we can change the scope for a specific command, using a token with any, as in

Understand "go to [any room]" as approaching.
Understand "find [any thing]" as finding.
Understand "ask [someone] about [any known thing]" as interrogating it about.

Or we can add areas and items to scope for all commands, as in

After deciding the scope of the player when the surveillance camera is switched on:
place the jail cell in scope.

Puncak Jaya demonstrates understanding references to characters who are currently off-stage.

★ See **Helping and Hinting** for objects tagged with a "seen" property when the player first encounters them

★ See **Getting Acquainted** for a character whose name is changed during the course of play as the player gets to know him better

★ See **Room Descriptions** for more ways to change the description of a room depending on player experience

★ See **Going, Pushing Things in Directions** for ways to understand the names of distant rooms and move towards them

★ See **Character Knowledge and Reasoning** for models of knowledge for other characters than the player

★ See **Sounds** for ways of tracking audible objects separately from visible ones

★ See **Lighting** for ways to change what the player knows about and can manipulate in dark rooms

★ See **Clocks and Scientific Instruments** for a telescope that lets the player view objects in another location

★ See **Continuous Spaces and The Outdoors** for more on seeing into adjacent locations

⬆ Start of Chapter 5: The Viewpoint Character

⬅ Back to §5.4. Background

➡ Onward to §5.6. Viewpoint

⬇ Example 149: ★ **Tense Boxing** An overview of all the variations of past and present tenses, and how they might be used.

⬇ Example 320: ★ **Puncak Jaya** When a character is not visible, responding to such commands as EXAMINE PETER and PETER, HELLO with a short note that the person in question is no longer visible.

⬇ Example 148: ★★★ **Zero** A box which called "horribly heavy box" after the player has tried to take it the first time.

⬇ Example 360: ★★★ **Casino Banale** Creating room descriptions and object descriptions that change as the player learns new facts and pieces things together.

§5.6. Viewpoint

Inform automatically creates a character for the player - a bland, personality-free entity at the outset, as we've seen. But there is no reason why the player need stick to this same identity throughout the story. Conventional fiction often jumps from one viewpoint character to another, and so can IF.

To do this at the most elementary level, we simply at some point

now the player is Janine;

where Janine is a person we've already defined in the code. Now the player is in whatever location Janine inhabits, carries whatever Janine carries, and wears whatever Janine is wearing. **Terror of the Sierra Madre** shows off this effect, and also demonstrates how to make the command prompt remind the player which character he currently controls. Some games instead give this information in the status line or after the name of the location when looking, producing output like

The Bottomless Acherousia (as Charon)

We could do the same by adding a line such as

After printing the name of a room while constructing the status line or looking:
say "[roman type] (as [the player])"

Of course, we'll need a good deal of other work to make Janine a distinct person from whichever character the player was before. The distinction may come from changed capabilities of the new character, which we can express through new rules about actions; e.g.,

Instead of listening when the player is Janine:
say "Your childhood accident left you unable to hear any but the loudest noises.
Currently there is only silence."

Janine may also have new, different perspective on her surroundings, expressed through the descriptions of the things she looks at; **Uncommon Ground** makes a "by viewpoint" token for text alternatives, allowing us to tag our descriptions to indicate which variations should be shown to which viewpoint characters. **The Crane's Leg 1** and **2** offer more elaborate and specialized ways of customizing the player character's observations to depend on how he relates (physically and in attitude) to the things around him.

If we want to change the tense and person of narration from the conventional present second person, we may do this as well:

When play begins:
now the story viewpoint is first person plural;
now the story tense is past tense.

Though this only changes the form of the text produced automatically by Inform (responses such as "you can't go that way" might become, say, "I couldn't go that way"), and all author-written text in the story must be written in the tense and person intended.

- ↑ Start of Chapter 5: The Viewpoint Character
 - ← Back to §5.5. Memory and Knowledge
 - Onward to Chapter 6: Commands: §6.1. Designing New Commands
 - ↓ Example 390: ★ **The Crane's Leg 2** A description text generated based on the propensities of the player-character, following different rulebooks for different characters.
 - ↓ Example 455: ★★ **Uncommon Ground** Making a "by viewpoint" token, allowing us to design our own text variations such as "[show to yourself]quaint[to Lolita]thrilling[to everyone else]squalid[end show]" depending on the identity of the player at the moment.
 - ↓ Example 52: ★★★ **The Crane's Leg 1** A description text that automatically highlights the ways in which the object differs from a standard member of its kind.
 - ↓ Example 121: ★★★ **Terror of the Sierra Madre** Multiple player characters who take turns controlling the action.
-

Examples from Chapter 5: The Viewpoint Character

- ↑ Start of this chapter
- Chapter 6: Commands
- ↘ Indexes of the examples

254

★ Example rBGH

WI

The player character's height is selected randomly at the start of play.

As with ordinary numbers, we can choose random units when this is useful:

"rBGH"

The Pharmaceutical Testing Facility is a room. "A [if the player is short]large [end if][if the player is tall]cramped [end if]white space with sterile counters and a[if the player is tall]n uncomfortable little[end if] stool. There is also a mirror, behind which someone must be watching you. But you can't see through to that."

A counter, a one-way mirror, and a stool are scenery in the Facility. The stool is an enterable supporter. The counter supports a plate.

Height is a kind of value. 5 feet 11 inches specifies a height. 5'11 specifies a height. A person has a height.

Definition: a person is tall if its height is 6 feet 0 inches or more.

Definition: a person is short if its height is 5 feet 4 inches or less.

When play begins:

now the height of the player is a random height between 5 feet 2 inches and 6 feet 4 inches;

now the right hand status line is "[height of player]".

Instead of examining the player:

say "You, Test Subject, are [height of the player] tall."

The growth pill is a kind of thing. A growth pill is always edible. The description is usually "It is leaf-green and has a reassuring logo of a curling vine on the side. Nothing to worry about, nothing at all." Two growth pills are on the plate.

After eating the growth pill:

increase the height of the player by 0 feet 6 inches;

say "Your spine does something frightening and painful, and you find yourself looking down on the room from a wholly new angle.";

try looking.

Test me with "examine me / eat pill / examine me / eat pill / examine me".

396

★★ Example Slouching

WI

A system of postures allowing the player and other characters to sit, stand, or lie down explicitly or implicitly on a variety of enterable supporters or containers, or in location.

Suppose we want to let the player explicitly sit, stand, or lie down on different enterable objects. (Normally Inform treats all these actions as entering, but there may be cases where we want to distinguish between the player sitting on a chair and the player standing on it.)

Our implementation gives each kind of enterable a range of allowed postures, and one preferred posture. If the player enters the supporter or container without specifying a posture (as in ENTER CHAIR), he will be put in the preferred posture. If he explicitly says, e.g., STAND ON CHAIR, he will be put in the standing position as long as standing is a posture that suits the chair.

"Slouching"

Section 1 - Posture Rules

A posture is a kind of value. The postures are seated, standing, and reclining.

A person has a posture. The posture of a person is usually standing.

A supporter has a posture. A container has a posture.

Posture-permission relates various things to various postures. The verb to allow means the posture-permission relation.

Understand the commands "stand" and "sit" and "lie" as something new.

Understand "sit on/in [something]" as sitting on.

Understand "lie on/in [something]" as lying on.

Understand "stand on/in [something]" as standing up on.

Sitting on is an action applying to one thing.
Lying on is an action applying to one thing.
Standing up on is an action applying to one thing.

Carry out an actor sitting on:
if the holder of the actor is not the noun, silently try the actor entering the noun;
if the holder of the actor is the noun:
if the actor is not seated, try the actor taking position seated;
otherwise follow the report taking position rules.

Carry out an actor lying on:
if the holder of the actor is not the noun, silently try the actor entering the noun;
if the holder of the actor is the noun:
if the actor is not reclining, try the actor taking position reclining;
otherwise follow the report taking position rules.

Carry out an actor standing up on:
if the holder of the actor is not the noun, silently try the actor entering the noun;
if the holder of the actor is the noun:
if the actor is not standing, try the actor taking position standing;
otherwise follow the report taking position rules.

Understand "lie down" as lying down.
Understand "sit down" or "sit" or "sit up" as sitting down.
Understand "stand" or "stand up" as standing up.

Lying down is an action applying to nothing.
Sitting down is an action applying to nothing.
Standing up is an action applying to nothing.

Taking position is an action applying to one posture.

Instead of an actor lying down:
try the actor taking position reclining; rule succeeds.
Instead of an actor sitting down:
try the actor taking position seated; rule succeeds.
Instead of an actor standing up:
try the actor taking position standing; rule succeeds.

Check an actor taking position:
if the holder of the actor is not a room and the holder of the actor does not allow the posture understood:
if the actor is the player:
say "You can't take that position [in-on the holder of the actor].";
otherwise if the actor is visible:
say "[The actor] can't take that position.";
stop the action.

Check an actor taking position:
if the posture understood is the posture of the actor:
if the actor is the player:
say "You are already [the posture understood].";

otherwise:
if the actor is visible, say "[The actor] is already [the posture understood].";
stop the action.

Carry out an actor taking position:
now the posture of the actor is the posture understood.

Report someone taking position (this is the position-report rule):
say "[The actor] is now [the posture of the actor][if the holder of the actor is not the location of the actor] [in-on the holder of the actor][end if]."

Report taking position:
say "You are now [the posture of the player][if the holder of the player is not the location] [in-on the holder of the player][end if]."

To say in-on (item - a thing):
if the item is a container, say "in [the item]";
otherwise say "on [the item]".

Carry out an actor exiting (this is the departure-posture rule):
let N be the holder of the actor;
if N is a container or N is a supporter,
now the posture of the actor is the posture of N;
otherwise now the posture of the actor is standing.

The departure-posture rule is listed after the standard exiting rule in the carry out exiting rulebook.

The departure-posture rule is listed after the standard getting off rule in the carry out getting off rulebook.

Carry out an actor entering something (this is the arrival-posture rule):
if the noun is a container or the noun is a supporter,
now the posture of the actor is the posture of the noun.

The arrival-posture rule is listed after the standard entering rule in the carry out entering rulebook.

Check an actor going somewhere:
if the actor is in a room and the actor is not standing:
say "([if the actor is not the player][the actor] [end if]first standing up)
[command clarification break]";
silently try the actor taking position standing;
if the actor is not standing, stop the action.

Section 2 - Some Generic Kinds

A chair is a kind of supporter. A chair is always enterable. Every chair allows seated and standing. A chair is usually seated.

A sofa is a kind of supporter. A sofa is always enterable. Every sofa allows seated, standing and reclining. A sofa is usually seated.

A hammock is a kind of container. A hammock is always enterable. Every hammock allows seated and reclining. A hammock is usually reclining.

Section 3 - The Scenario

The Resort is a room.

The banana hammock is a hammock in the Resort. The stone bench is a sofa in the resort.

Clark is a man in the Resort. A persuasion rule: persuasion succeeds.

Rule for writing a paragraph about someone (called target):
say "[The target] is [posture] [if the holder of the target is the location]nearby[otherwise][in-on the holder of the target][end if]."

Rule for writing a paragraph about something which encloses an unmentioned person (called target):

carry out the writing a paragraph about activity with the target instead.

Test me with "sit on bench / stand on bench / get up / lie on hammock / sit up / g / clark, sit on bench / look / clark, lie down / g / look / clark, get up / look / clark, lie down / look / enter bench".

59



Example The Night Before

WI

Instructing Inform to prefer different interpretations of EXAMINE NOSE, depending on whether the player is alone, in company, or with Rudolph the Red-nosed Reindeer.

Suppose that we're going to give every person in the game a nose, but we want references to a nose always to mean the nose of someone *else*, if the player is with one other person. Moreover, on some occasions we're going to be in sight of Rudolph, so actions directed at an unspecified nose should always prefer his.

This relies on a somewhat advanced technique from the Understanding chapter, but since it may become useful with assemblies and body parts, it is worth mentioning here.

"The Night Before"

The North Pole is a room. "Here it is: the famous Pole. From here you can go south (or south-south, or south-south-by-south); or, alternatively, take refuge inside a red-and-white-striped cabin." The cabin is scenery in the North Pole. Instead of entering the cabin, try going inside.

Santa is a man in the North Pole. "Santa is pacing around in the snow and trying to psych himself up for the big night."

Inside from North Pole is the Candy Cane Cabin. The description of the Cabin is "Striped red and white, but nothing can make this place seem warm and inviting since Mrs. Santa ran off with the Tooth Fairy."

The Ice Shelf is south of North Pole. "The ice here has been smoothed into a kind of runway for easy take-off, and ends in a cliff and cold arctic sea." Donner, Vixen, Blixen, and Rudolph are animals in the Ice Shelf.

A nose is a kind of thing. A nose is part of every person. The description of Santa's nose is "It's a bit ruddy. You don't like to mention it, but Santa's been dipping heavily into the Grey Goose since Mrs. Santa left town." The description of a nose is usually "Not terribly exciting." The description of Rudolph's nose is "See how it glows!"

Next, we'll teach Inform some vocabulary to distinguish between the player and everyone else:

Definition: a person is other if it is not the player.

Definition: a thing is selfish if it is part of the player and the player can see an other person.

Instead of examining a selfish nose:
say "You cross your eyes, but can't get a good look."

Here is the part that actually determines the preferences. "Does the player mean..." can result in five outcomes: "it is very unlikely", "it is unlikely", "it is possible" (the neutral default), "it is likely", and "it is very likely". This is discussed in greater detail in the Understanding chapter. Here, we want to discourage references to the player's own nose and encourage references to the nose of Rudolph, so:

Does the player mean doing something when the noun is a selfish nose or the second noun is a selfish nose: it is very unlikely.

Does the player mean doing something to Rudolph's nose: it is very likely.

And this part is just for decoration:

Rule for writing a paragraph about Rudolph:
say "The reindeer are already harnessed and waiting impatiently. The brilliance of [Rudolph]'s nose casts an eerie red glow over [the list of unmentioned animals in the location]."

Test me with "x nose / x my nose / x santa's nose / in / x nose / out / s / x my nose / x nose / x rudolph's nose / x donner's nose".

416

★ Example Identity Theft

WI

Allowing the player to enter a name to be used for the player character during the game.

Let's say we want to allow the player to enter any name he likes for his character. Moreover, we want to reject very long names (which are likely to be mistakes anyway), and we want to extract the player's chosen first name from the rest.

"Identity Theft"

The player's forename is a text that varies. The player's full name is a text that varies.

When play begins:

now the command prompt is "What is your name? > ".

To decide whether collecting names:

if the command prompt is "What is your name? > ", yes;

no.

After reading a command when collecting names:

if the number of words in the player's command is greater than 5:

say "[paragraph break]Who are you, a member of the British royal family?

No one has that many names. Let's try this again.";

reject the player's command;

now the player's full name is the player's command;

now the player's forename is word number 1 in the player's command;

now the command prompt is ">";

say "Hi, [player's forename]![paragraph break]";

say "[banner text]";

move the player to the location;

reject the player's command.

We also want to postpone the proper beginning of the game until we've gotten the name:

Instead of looking when collecting names: do nothing.

Rule for printing the banner text when collecting names: do nothing.

Rule for constructing the status line when collecting names: do nothing.

Your Bedroom is a room. The printed name of Your Bedroom is "[player's forename]'s Bedroom".

The player carries a letter. The description of the letter is "Dear [player's full name], [paragraph break]You have won the Norwegian Daily Lottery! ...".

If we are compiling for Glulx, this is enough to capture not only the player's name but also the capitalization he uses.

If we are compiling for the Z-machine, the player's input will unfortunately be reduced to lower case before we can inspect it. If we would like by default to capitalize the first letter of each word of the name, we might substitute the following after reading a command rule:

After reading a command when collecting names:

if the number of words in the player's command is greater than 5:

say "[paragraph break]Who are you, a member of the British royal family?

No one has that many names. Let's try this again.";

reject the player's command;

now the player's full name is the substituted form of "[the player's command in title case]";

now the player's forename is word number 1 in the player's full name;

```
now the command prompt is ">";
say "Hi, [player's forename]![[paragraph break]";
say "[banner text]";
move the player to the location;
reject the player's command.
```

309



Example **Baritone, Bass**

WI

Letting the player pick a gender (or perhaps other characteristics) before starting play.

The "reading a command" activity is rather advanced; for the moment, what we need to understand is that we're intervening in commands at the start of play and insisting that the player's first instruction to the game consist of a choice of gender. After that point, the gender will be set and play will proceed as normal.

In order to do the parsing, we define gender as a kind of value, and give several alternate names to each gender.

"Baritone, Bass"

Getting Started is a room.

Gender is a kind of value. The genders are masculine, feminine, and unknown. Understand "male" or "man" or "M" as masculine. Understand "female" or "woman" or "F" as feminine.

A person has a gender. The gender of the player is unknown.

When play begins:

```
now the command prompt is "Please choose a gender for your character. >".
```

After reading a command when the gender of the player is unknown:

```
if the player's command includes "[gender]":
  now the gender of the player is the gender understood;
  if the gender of the player is unknown:
    say "This story requires a selection of male or female. [run paragraph
on]";
    reject the player's command;
  if the gender of the player is masculine, now the player is male;
  if the gender of the player is feminine, now the player is female;
  say "[line break]Thank you. We now begin...";
  now the command prompt is ">";
  move the player to Sandy Beach;
  reject the player's command;
otherwise:
  say "Sorry, we're not ready to go on yet. [run paragraph on]";
  reject the player's command.
```

Sandy Beach is a room.

Instead of examining the player when the player is female:
say "Congratulations, you are a girl!"

Instead of examining the player when the player is male:
say "Congratulations, you are a boy!"

If we had a whole series of things to ask the player about, we might define a whole series of kinds of value

The vocal ranges are soprano, mezzosoprano, contralto...

and use a "construction stage" variable to keep track of the current stage of character-construction, as in

After reading a command when the current construction stage is choosing a vocal range:

...

451



Example Pink or Blue

WI

Asking the player to select a gender to begin play.

Suppose we would like to allow the player to choose a gender for the main character. We'd also like this to happen at the beginning of the game and outside the main parsing sequence. "When play begins" seems like a good place to put this.

"Pink or Blue"

When play begins:

```
say "Should your character be male or female? >";  
if men win, now the player is male;  
otherwise now the player is female;  
say paragraph break.
```

Now a piece of Inform 6 code handles the unusual input. It's not necessary to understand this to use it, and the code should work for any question you'd like to ask the player. The first three words in quotation marks ('male', 'M', 'man'...) correspond to positive feedback; the later three words correspond to negative feedback. So "to decide whether men win" will be true if the player types one of the first three, and false if he types one of the last three.

To decide whether men win:

```
(- Question('male','M//','man','female','F//','woman') -)
```

Include (-

```
[ Question pos1 pos2 pos3 neg1 neg2 neg3 first_word_typed;  
while (true) {  
  VM_ReadKeyboard(buffer, parse);  
  wn = 1; first_word_typed = NextWordStopped();
```

```

    if (first_word_typed == pos1 or pos2 or pos3) rtrue;
    if (first_word_typed == neg1 or neg2 or neg3) rfalse;
    print "Please choose ", (address) pos1, " or ", (address) neg1, ". > ";
}
];
-)

```

Instead of examining the player when the player is female:
 say "Congratulations, you are a girl!"

Instead of examining the player when the player is male:
 say "Congratulations, you are a boy!"

The Room of Self-Knowledge is a room. "Mirrors cover every available wall-surface of this hexagonal chamber, allowing you to examine yourself from all angles."

73

Example Finishing School

WI

The "another" adjective for rules such as "in the presence of another person".

It is often useful to write action rules which apply only when the player is observed by a third party. "In the presence of a person", however, will react even if only the player is in the room, because the player is, of course, a person as well.

A convenient way around this problem is to define an "other" adjective:

"Finishing School"

The apple is an edible thing carried by the player.

The Hall is a room. Miss Wicket is a woman in the Hall.

The Dormitory is south of the Hall.

Definition: a person is another if it is not the player.

Instead of eating something in the presence of another person:
 say "Your mannerly upbringing prevents you from eating without a fork or knife in front of someone."

Test me with "eat apple / south / eat apple".

If we did not have "another" here, Inform would interpret even the player as a possible schoolmarm, leading to such lines as "yourself stares at you coldly...". Clearly not quite the thing.

✦ Example **Bad Hair Day**

Change the player's appearance in response to EXAMINE ME.

"Bad Hair Day"

The Foyer is a room. "A mirror hangs over the table, tempting you to check your appearance before going in with all the others."

Instead of examining the player:
say "Oh, stop fussing. You look fine."

Test me with "examine me".

✦ Example **Dearth and the Maiden**

Our heroine, fallen among gentleman highwaymen, is restrained by her own modesty and seamliness.

The following example, indebted to the late Georgette Heyer, is suggestive:

"Dearth and the Maiden"

The Chequers Inn is a room. "The room is panelled and ceilinged in oak, with blue curtains to the windows and blue cushions on the high-backed settle by the fire."

An oil painting is in the Inn. "An oil painting hangs upon one wall, a lascivious work from the Indies in which a very bendy, sloe-eyed courtesan - but no."

A man called Mr Carr is in the Inn. "Standing bashfully aside is one Mr Carr, who we have been led to understand is by profession a Highwayman (yet whose visage oddly recalls Lord John Carstares, disgraced eldest son of the Earl of Wyncham)."

Kissing Mr Carr is unmaidenly behaviour. Doing something to the painting is unmaidenly behaviour.

Instead of unmaidenly behaviour in the Inn, say "How unmaidenly! Why, one might just as wantonly strip a rose of its petals, letting each fragrant leaf flutter slowly to the ground."

Test me with "examine painting / take painting / kiss mr carr".

✦ Example **Merlin**

A REMEMBER command which accepts any text and looks up a response in a table of recollections.

"Merlin"

Understand "remember [text]" as remembering.

Remembering is an action applying to one topic.

Carry out remembering:
say "Nothing comes to mind."

Instead of remembering a topic listed in the Table of Recollections:
say "[response entry][paragraph break]".

Table of Recollections

Topic	Response
"rain/weather"	"You've seen worse, but not often: it's falling so hard now that the tin rattles and the runoff, on the low side of the roof, would be a tenable source of hydroelectric power."
"hydroelectric power" or "power/hydroelectric"	"It's not as though you have any sort of light bulb in here to turn on, even if you could power it."
"light bulb" or "light/bulb"	"Light bulbs, like so much else, are a thing of your past. Or is it your future? Tricky, the way the world loops round on itself."
"past/time/future"	"Living backwards has its drawbacks. A tendency to confuse and annoy your friends, being one; the total inability to maintain a stable relationship; and a deep dissatisfaction with most of the bodily processes people enjoy, since they ultimately make you hungrier, colder, or-- no point dwelling on it, really."
"backwards"	"It's not even exactly *backwards*, now is it? It's more like a series of forwardses stuck back to back. As though someone had taken each track of a CD and put them in the exactly wrong order. You miss that. The music on demand."

The Inadequate Shelter is a room. "A piece of corrugated tin, leaned on two sticks, and pathetically augmented with a tire (on one side) and a cardboard box (on the side towards the wind). And that's what you've got between you and the driving rain.

At the moment rain is all you can remember, in fact."

Test me with "remember rain / remember power / remember light bulb / remember future / remember backwards".

301

★ Example One of Those Mornings

WI

A FIND command that allows the player to find a lost object anywhere

Suppose that, contrary to the usual rules of interactive fiction, we want to allow the player to discover the locations of things he hasn't actually seen yet:

"One of Those Mornings"

Understand "find [any thing]" as finding.

Finding is an action applying to one visible thing.

Carry out finding:
if the player is carrying the noun:

```
say "You're holding [the noun]!";
otherwise:
say "You left [the noun] [if the noun is on a supporter]on[otherwise]in[end if]
[the holder of the noun]."
```

The holder of the noun can be a room, a supporter, or a container: the phrase is not picky. We would want to be a little more careful if it were ever possible for an item to have been "removed from play" in our game, since then the holder could be nothing, and that would have odd results. In this particular example, though, that will not arise.

And that's it, as far as the find command goes. The rest is local color.

```
The Exhibition Room is a room. It contains a closed locked lockable transparent
openable container called the display case. The display case contains a
priceless pearl. The display case is scenery. The description of the Exhibition
Room is "By far the finest thing in the room is a priceless pearl in a glass display
case. It should of course be yours[if key is not visible], if only you can remember
where you hid the key[end if]."
```

```
The silver key unlocks the display case.
```

```
A jade vase, a teak chest, a bronze teakettle, and a child's burial casket are
openable closed containers in the Exhibition Room.
```

```
After taking the pearl:
```

```
say "The pearl rolls into your hand, gleaming in the oblique light; your fortune
is made.";
end the story finally.
```

If we want to have the key found in different places when the game is replayed:

```
When play begins:
```

```
let the space be a random container which is not the display case;
move the silver key to the space.
```

```
Every turn:
```

```
say "Your watch ticks with maddening loudness."
```

```
The time of day is 1:02 AM.
```

```
At 1:08 AM: say "The security guard arrives to find you fumbling about with keys.
Curses."; end the story.
```

```
Test me with "find pearl / find teakettle / get teakettle / find teakettle / find key".
```

Here we have a box that prints out its current state and its history each time we open and close it:

"Tense Boxing"

The Temporal Prism is a room. "A room of angled mirrors, in whose surfaces you can see what is now; what just was; what has always been. A final mirror is broken and its frame gapes blackly."

The mysterious box is in the Temporal Prism. It is an openable closed container.

To assess the box:

- if the box was not open, say "The box was not open.";
- if the box was open, say "The box was open.";
- if the box had not been open, say "The box had not been open.";
- if the box had been open, say "The box had been open.";
- if the box is not open, say "The box is not open.";
- if the box is open, say "The box is open.";
- if the box has not been open, say "The box has not been open.";
- if the box has been open, say "The box has been open."

Before opening the mysterious box:

- say "You are about to open the box.";
- assess the box.

Before closing the mysterious box:

- say "You are about to close the box.";
- assess the box.

After opening the mysterious box:

- say "You now open the box.";
- assess the box.

After closing the mysterious box:

- say "You now close the box.";
- assess the box.

Note that "was..." and "was not..." and so on may describe conditions more complicated than simple properties: we could equally well ask "if the box has been in the sack", "if the box had been carried by the player", and so on.

The past ("if the box was...") and past perfect ("if the box had been...") are especially useful for cases where we want to report on an action after the state of the item has changed; so, for instance:

After taking the mysterious box:

- if the box had not been carried by the player, say "You lift the mysterious box for the first time.";
- if the box had been carried by the player, say "You again pick up the mysterious box."

Test me with "open box / close box / open box / take box / drop box / take box".

This is in many respects similar to a rule beginning "After taking the mysterious box for the first time...", but it is superior in most circumstances, for two reasons.

First, it will respond correctly even if the player has somehow carried the box before without taking it explicitly: for instance, if another character gave him the box, if the box were moved into his inventory as a result of another action, or if the player carried the box at the start of play. Inform begins its reckoning of time when the game begins, so if the box is defined as being open at the outset, "if the box has been open" will always be true.

Second, "after taking... for the first time" fires only the first time the player attempts to take something. If the player tried to take the box, failed, and then tried again later, the "for the first time..." rule would not fire; our "if the box has not been carried..." rule would.

320

★ Example **Puncak Jaya**

WI

When a character is not visible, responding to such commands as EXAMINE PETER and PETER, HELLO with a short note that the person in question is no longer visible.

By default, when something is not present, Inform does not allow a player to refer to it. But there are times when we might like to acknowledge that the thing mentioned in a command does exist somewhere in the game; it just happens not to be on hand right now.

One way to do this is to make an object that appears everywhere and responds to the name of its owner only when the owner itself is not in view.

"Puncak Jaya"

A ghost is a kind of person. A man-ghost is a kind of ghost. A man-ghost is always male. A woman-ghost is a kind of ghost. A woman-ghost is always female.

We make the ghost a person rather than some other kind of thing so that it will be able to respond to commands such as KISS BOB or (even trickier) BOB, JUMP: if Inform did not recognize the ghost as an animate creature, it would not accept such input.

Representation relates one ghost to one person. The verb to represent means the representation relation.

One man-ghost represents every man. One woman-ghost represents every woman.

This is, technically, an assembly -- except instead of saying that every device has a button part, or that there are three daffodils in every garden room, the assembly is based on a non-physical relation that we just designed.

Based on the "representation" relation, we now devise a conditional relation that applies only when the represented thing is not itself in view:

Indication relates a ghost (called X) to a person (called Y) when X represents Y and Y is not visible.

Understand "[something related by indication]" as a ghost.

When play begins:
now every ghost is in the concept-repository.

Instead of doing something to a ghost:
say "You seem to have left [a random person which is represented by the noun] behind."

Instead of doing something when the second noun is a ghost:
say "You seem to have left [a random person which is represented by the second noun] behind."

The concept-repository is an open unopenable transparent container. It is part of the air. The air is a backdrop. It is everywhere.

Base of Puncak Jaya is a room. Temple, Kippax, and Huizenga are men in Base. Peak of Puncak Jaya is above Base of Puncak Jaya.

Test me with "x kippax / up / x kippax / kiss kippax / kippax, hello".

Further complications of this example might require that the player meet a character before being able to refer to him or her.

148



Example Zero

WI

A box which called "horribly heavy box" after the player has tried to take it the first time.

The names of objects might want to change also:

"Zero"

Weight is a kind of value. The weights are light, manageable, and horribly heavy. Everything has a weight.

A thing is usually manageable.

This does require a rule for an activity (see the chapter on Activities), but it's a fairly straightforward one:

Before printing the name of a horribly heavy thing (called weighty object):
if we have taken the weighty object, say "[weight]".

The Nearly Empty Living Room is a room. "Nearly everything is out, all of Helen's possessions and most of yours." A man called Mr Zero is in the Nearly Empty Living Room. "Mr Zero, despite being heavily paid to assist in this operation, is giving you a look that clearly conveys his lack of interest in budging even one more item." The description of Mr Zero is "Many muscles, no hair."

The book box is a horribly heavy thing in the Living Room. The clothing box is a manageable thing in the Living Room. A broom, a dustpan, some packing tape, and a discarded newspaper are light things in the Living Room.

After taking a horribly heavy thing:
say "Taken. (Oof.)"

Test me with "get clothing / look / get book box / look / i / drop book box / look".

360



Example Casino Banale

WI

Creating room descriptions and object descriptions that change as the player learns new facts and pieces things together.

In a work of interactive fiction that involves many new discoveries, we might want to change the way we narrate room descriptions and describe objects as the player learns new information.

One approach to this is to create a model of the facts we want the player to find out, and attach some narrative text to each. When a fact becomes relevant to the story, that narrative text is shown to the player. So:

"Casino Banale"

Section 1 - Procedure

First we create the concept of facts, and the idea that facts can make some things more important than others.

A fact is a kind of thing. A fact can be known or unknown. A fact can be ready to learn or hidden. A fact has some text called the narration.

Definition: a thing is narratively significant if it conveys an interesting fact.

Definition: a thing is narratively dull if it is not narratively significant.

Conveyance relates various things to various facts. The verb to convey means the conveyance relation.

Definition: a fact is interesting if it is unknown and it is ready to learn.

Now, we also need a way to tell Inform to introduce certain new facts when the right previous ones have been introduced. We'll create a "following" relation, according to which a new fact can be told to the player when the player has already learned all the

facts it follows. This way, we can simulate the effect of putting together several pieces of evidence to come to a conclusion:

Following relates various facts to various facts. The verb to follow means the following relation.

To say (new fact - a fact):

say "[narration of the new fact]";

now the new fact is known;

repeat with possible outcome running through facts which follow the new fact:

if every fact which is followed by possible outcome is known:

now the possible outcome is ready to learn.

Next we need a way for the game to introduce these new facts. Let's say we want them to come up when the player examines something appropriate, or sees it in the room:

After examining something which conveys an interesting fact (called discovery):
say "[discovery][paragraph break]".

After choosing notable locale objects:

repeat through the Table of Locale Priorities:

if the notable-object entry is narratively significant:

set the locale priority of the notable-object entry to 1.

For writing a paragraph about a narratively significant thing (called item):

now the item is mentioned;

let chosen fact be a random interesting fact which is conveyed by the item;

say "[chosen fact][paragraph break]".

The "after choosing notable locale objects" line here handles things so that any interesting conclusions we want to draw are always given first, followed by the less interesting description.

And finally, we need to give the player a little evidence to piece together:

Section 2 - Scenario

The Casino is a room.

Frince is a man in the Casino. The description is "Frince is a friend of yours -- if you reckon friendship on the same terms that one reckons a cat as a pet. He spends time with you when he wants to, but if your wishes or convenience ever run counter to a whim of his, it's the whim that wins. Always. [paragraph break]He's also wearing a somewhat ludicrous shirt."

Frince wears a ludicrous shirt. The description of the ludicrous shirt is "Fine white fabric with satiny white pinstripes: it's that expensive, effeminate look that Frince is so fond of, and which -- combined with his name -- gives people completely the wrong idea about him."

Tim is a man in the Casino. The description is "You don't know Tim well. Kind of wall-flowerish. The only thing that seems to excite him is craps."

Penny is a woman in the Casino. The description is "Loud. Brash. Hot, probably, if you can look past the loud and brash."

Rule for writing a paragraph about a narratively dull person:

let is-are-n be "is";
if the number of unmentioned narratively dull people is not 1:
let is-are-n be "are";
say "[A list of unmentioned narratively dull people] [is-are-n] [one of]watching the croupier[or]following the spin of the roulette[or]chattering[at random][one of] breathlessly[or] impatiently[or][at random]."

penny-annoying is a fact.

It is ready to learn.
The narration is "[if looking]Penny grimaces at you-- [end if]Penny is the same woman who stepped on your toe in the buffet line. The third time, she blurted, 'You have big shoes, don't you?'"
Penny conveys penny-annoying.

lipstick-smudges is a fact.

It is ready to learn.
The narration is "There are a couple of smudges of coral-colored lipstick on the collar."
The ludicrous shirt conveys lipstick-smudges.

penny-wears-coral is a fact.

It follows penny-annoying.
The narration is "[if looking]Penny catches your eye again. [end if]The bright coral lipstick was really not a wise choice."
Penny conveys penny-wears-coral.

Affair-with-penny is a fact.

It follows lipstick-smudges and penny-wears-coral.
The narration is "You avoid [if examining Frince]his[otherwise]Frince's[end if] eye. You need some time to adjust to the image of him making out with Penny in a storage closet before you can talk to him without appalled giggling."
Frince conveys affair-with-Penny.

Test me with "x penny / x frince / x shirt / look".

★ **Example The Crane's Leg 2**

A description text generated based on the propensities of the player-character, following different rulebooks for different characters.

Names of rules can be listed in tables. This is convenient if, for instance, we decide that we'd like to swap the rules we use for a specific purpose, as in this continuation of our earlier example of automated description:

"The Crane's Leg, Grown Longer"

Material is a kind of value. The materials are wood, glass, stone, cloth, paper, clay, and metal. A thing has a material.

Color is a kind of value. The colors are red, orange, yellow, green, blue, indigo, violet, black, brown, and white. A thing has a color. A thing is usually white.

A height is a kind of value. 3 feet 11 inches specifies a height. A thing has a height. Definition: a thing is tall if its height is 6 feet 0 inches or more. Definition: a thing is short if its height is 2 feet 0 inches or less.

Imitation relates various things to one thing (called the ideal). The verb to imitate means the imitation relation.

A table is a kind of supporter. A table is usually wood. The height of a table is usually 3 feet 8 inches. The ordinary table is a table. Every table imitates the ordinary table.

A rock is a kind of thing. A rock is usually stone. The ordinary rock is a rock. The height of a rock is usually 0 feet 3 inches. Every rock imitates the ordinary rock.

The description of a thing is usually "[comparison with ideal][run paragraph on]".

To say comparison with ideal:

- say "You observe [the noun]:[paragraph break]";
- choose row with character of the player in Table of Descriptive Reporting;
- follow instructions entry.

This is the comparative observation rule:

- let the sample be the ideal of the noun;
- if the sample is not a thing:
 - say "Nothing special, really.";
 - rule succeeds;
- if the material of the noun is not the material of the sample:
 - if the height of the noun is not the height of the sample:
 - if the noun is shorter than the sample, say "Unusually short at [height of the noun], and made of [material of the noun].";
 - otherwise say "Unusually tall at [height of the noun], and made of [material of the noun].";
 - otherwise:
 - say "Distinct mostly in being made of [material of the noun].";
- otherwise:
 - if the height of the noun is not the height of the sample:
 - if the noun is shorter than the sample, say "Unusually short at [height of the noun].";
 - otherwise say "Unusually tall at [height of the noun].";
 - otherwise:
 - say "In every respect [a sample]."

The Pleasure Garden is a room. "At the riverbank, a pleasing garden, having many curving paths and one straight."

The low table is a table in the Pleasure Garden. The height of the low table is 2 feet 3 inches. On the low table is a yellow metal rock called a gold nugget. A willow is in the Pleasure Garden. The height of the willow is 20 feet 2 inches.

Understand "possess [any person]" or "be [any person]" as possessing.

Possessing is an action applying to one thing. Carry out possessing: now the player is the noun; say "You swap bodies!"

The crane is a person in the Garden. The height of the crane is 4 feet 0 inches.

Table of Descriptive Reporting

character	instructions
yourself	comparative observation rule
crane	bird observation rule

This is the bird observation rule:

if the noun is shorter than the player, say "Small, like a duck[if the color of the noun is not white]; and [color of the noun][end if].";

otherwise say "Supremely tall[if the color of the noun is not white] and [color of the noun][end if]."

Test me with "examine table / examine nugget / examine willow / possess crane / examine table / examine nugget / examine willow".

455



Example Uncommon Ground

WI

Making a "by viewpoint" token, allowing us to design our own text variations such as "[show to yourself]quaint[to Lolita]thrilling[to everyone else]squalid[end show]" depending on the identity of the player at the moment.

A slightly more challenging case than the "by atmosphere" example is one in which we want to create text variations depending on the identity of our player character.

What we want to do is build a switch statement in I6, one that looks something like

```
switch(player)
{
  yourself: print "quaint";
  Lolita: print "thrilling";
  default: print "squalid";
}
```

out of I7 that looks like this:

say "[show to yourself]quaint[to Lolita]thrilling[to everyone else]squalid[end show]".

"Uncommon Ground"

The Mud Village is a room. "You stand at the center of a [show to yourself]quaint[to Lolita]thrilling[to everyone else]squalid[end show] mud village."

Leforge is a man in the Mud Village. Lolita is a woman in the Mud Village.

Instead of waiting:

```
if the player is Lolita, now the player is Leforge;  
if the player is yourself, now the player is Lolita;  
say "You jump bodies. Whoops!"
```

To say show to (N - a person) -- beginning say_seen_by:

```
(-  
  switch(player)  
  {-open-brace}  
  {N}:  
-).
```

To say to (N - a person) -- continuing say_seen_by:

```
(-  
  {N}:  
-).
```

To say to everyone else -- continuing say_seen_by:

```
(-  
  default:  
-)
```

To say end show -- ending say_seen_by:

```
(-  
  {-close-brace}  
-)
```

Test me with "look / z / look / z / look".



Example The Crane's Leg 1

A description text that automatically highlights the ways in which the object differs from a standard member of its kind.

When you see a long object, you don't have to think that it is too long if being long is the property given by the nature. It is proved by the fact that a duckling, having short legs, will cry if you try to draw them out by force, and that a crane, having long legs, will protest you with tears if you try to cut them with a knife. -- Zhuangzi

Suppose we have an extremely detailed world model in which every object is characterized by many features -- in this example, material and height, though one could add more. Suppose further that we would like to generate descriptions of these things automatically for the most part, drawing the player's attention only to those aspects of the object that are particularly interesting.

"The Crane's Leg"

Material is a kind of value. The materials are wood, glass, stone, cloth, paper, clay, and metal. A thing has a material.

A height is a kind of value. 3 feet 11 inches specifies a height. A thing has a height. Definition: a thing is tall if its height is 6 feet 0 inches or more. Definition: a thing is short if its height is 2 feet 0 inches or less.

So far, we have generally dealt with cases where the property of a thing can be a number (such as 3), a value (such as brightness), or a unit (like height, here). It is also possible for a thing to have a property which names another thing, as in "The mother of the baby trout is the large trout" -- where "mother" is a property, and its value, in the case of the baby trout, is large trout. We would define such a property with a line such as "A fish has a thing called the mother."

In practice, though, this is a bit confusing as syntax; moreover, Inform has a much more powerful construct for talking about the ways in which one object relates to another object. A full discussion of this will have to wait for the chapter on Relations. For now, it is enough to say that we can do this:

Imitation relates various things to one thing (called the ideal). The verb to imitate means the imitation relation.

This will allow us to declare that some objects imitate other objects, like so:

A chair is a kind of supporter. A chair is always enterable. A chair is usually wood. The height of a chair is usually 4 feet 0 inches. The ordinary chair is a chair. Every chair imitates the ordinary chair.

A table is a kind of supporter. A table is usually wood. The height of a table is usually 3 feet 8 inches. The ordinary table is a table. Every table imitates the ordinary table.

A rock is a kind of thing. A rock is usually stone. The ordinary rock is a rock. Every rock imitates the ordinary rock. The height of a rock is usually 0 feet 3 inches.

A jug is a kind of container. A jug is usually clay. The ordinary jug is a jug. Every jug imitates the ordinary jug. The height of a jug is usually 0 feet 8 inches.

Now each of these types has one ideal representative which has the fundamental attributes of its kind: the ordinary chair is the most chairlike chair imaginable, the ordinary table is the epitome of tableness, and so on. We are also allowed to refer to "the ideal of the chair", thanks to the way we defined imitation. (Again, the relations chapter offers a much more detailed explanation of how relations may be defined.)

The description of a thing is usually "[comparison with ideal][run paragraph on]".

To say comparison with ideal:

- let the sample be the ideal of the noun;
- if the sample is not a thing:
 - say "Perfectly conforming to your expectations of its type.";
 - rule succeeds;
- if the material of the noun is not the material of the sample:
 - if the height of the noun is not the height of the sample:
 - if the noun is shorter than the sample, say "Unusually short at [height of the noun], and made of [material of the noun].";

otherwise say "Unusually tall at [height of the noun], and made of [material of the noun].";
 otherwise:
 say "Distinct mostly in being made of [material of the noun].";
 otherwise:
 if the height of the noun is not the height of the sample:
 if the noun is shorter than the sample, say "Unusually short at [height of the noun].";
 otherwise say "Unusually tall at [height of the noun].";
 otherwise:
 say "In every respect [a sample]."

The Pleasure Garden is a room. "At the riverbank, a pleasing garden, having many curving paths and one straight."

The low table is a table in the Pleasure Garden. The height of the low table is 2 feet 3 inches. On the low table is a metal rock called a gold nugget.

Test me with "x table / x nugget".

So far the effect is not very deep, but we could take the auto-description a great deal further: providing a larger and more interesting set of variations; or writing a complicated set of rules such that the player only notices height variations when carrying a ruler; or switching between several player-characters, each of whom notices a different subset of characteristics. But these refinements would require more input from later chapters.



Multiple player characters who take turns controlling the action.

Suppose we have a game where we want the player to control two different characters, swapping bodies from one turn to the next. First, the setting, and the two people who will alternately play:

"Terror of the Sierra Madre"

The Hay-Strewn Corridor is a room. "[if the player is Maleska]The horse stalls are empty: you have already drained the animals, and carried off their corpses. The house will not long sustain you now.

The window throws on the floor a bright square of malevolent sunlight[otherwise]The stalls for horses run down one side of the room, but the house has long stood empty. A square window without shutters looks out over the ranch, away toward the Sierras[end if]."

Teresa is a woman in the Hay-Strewn Corridor. "Teresa stands opposite you[if Teresa carries something], her fingers wrapped tightly around [a list of things carried by Teresa][end if]." Teresa carries a bulb of garlic and a cross.

Maleska is a man in the Hay-Strewn Corridor. "Maleska watches you from eyes entirely black." Maleska carries a skull.

If we tried the text above in Inform, we would find ourselves in the Hay-Strewn Corridor and confronted by both Teresa and Maleska. If "player" is not set to any named person, Inform creates a bland person called "yourself" to represent the player. To avoid this, we set "player" to the person we want to begin as. The player character is normally privately-named, so we'll need to make sure "Maleska" still means what it should.

The player is Maleska. Understand "Maleska" as Maleska.

Now the Corridor contains just two people, and we arrive on the scene as Maleska, with only Teresa facing us.

At the end of every turn we will use the 'now the player is...' phrase. (This looks as if it simply changes the value of "player": which it does, but it also carries out a complicated operation behind the scenes to effect the switch.)

Every turn:
if the player is Maleska, now the player is Teresa;
otherwise now the player is Maleska.

Our two characters already see the Corridor differently, but let's differentiate them further:

Every person has a number called strength. The strength of Teresa is 3. The strength of Maleska is 5.

In this small example, strength is not used for anything, except that we will display it on the status line:

When play begins:
now the command prompt is "[bold type][player][roman type] > ";
now the left hand status line is "[player]";
now the right hand status line is "STR: [strength of the player]".

That last rule doesn't quite do what we might have expected. When we print "[player]", we find that Inform usually prints "yourself". This is because Inform says "you" to mean Teresa when talking to the player-being-Teresa, and likewise for Maleska. We want to override that in this particular story, because the rapid switches of personality are otherwise hard to follow. So:

Rule for printing the name of Teresa: say "Teresa".

Rule for printing the name of Maleska: say "Maleska".

Test me with "look / look".

Chapter 6: Commands

§6.1. *Designing New Commands*; §6.2. *Writing New Commands*;
§6.3. *Modifying Existing Commands*; §6.4. *Looking*; §6.5. *Examining*;
§6.6. *Looking Under and Hiding*; §6.7. *Inventory*;
§6.8. *Taking, Dropping, Inserting and Putting*; §6.9. *Going, Pushing Things in Directions*;
§6.10. *Entering and Exiting, Sitting and Standing*; §6.11. *Waiting, Sleeping*;
§6.12. *Other Built-In Actions*; §6.13. *Magic Words*;
§6.14. *Remembering, Converting and Combining Actions*;
§6.15. *Actions on Multiple Objects*; §6.16. *Alternate Default Messages*;
§6.17. *Clarification and Correction*; §6.18. *Alternatives To Standard Parsing*

-  Contents of *The Inform Recipe Book*
-  Chapter 5: The Viewpoint Character
-  Chapter 7: Other Characters
-  Indexes of the examples

§6.1. Designing New Commands

Quite a bit of interactive fiction design involves the creation of custom commands to expand on the library's existing set. There is more to know than we can review in this section; instead, this is to serve as an overview of the process, with hints about where in *Writing with Inform* we might find more technical details.

Before we even start to write our source text, we should think about the following things:

- (1) What words will the player use to make this new action happen?
- (2) What will the action change about the world model?
- (3) What circumstances might make the new action go wrong or produce silly outcomes?

To take these one one by one:

(1) We may have a general idea of the phrasing we want the player to use -- say we want to add an SHOOT command which allows the player to fire a gun at something. (This is an intentionally tricky choice of verb, because it shows off so many possibilities.) So we might decide the base form of the action will be

SHOOT THE PISTOL AT HENRY

So now we're going to need an action that applies to two objects -- the pistol as the noun, and Henry as the second noun. The problem is, though, that there are lots of other ways that the player could reasonably formulate the command, some of which leave out information:

SHOOT HENRY
SHOOT PISTOL
FIRE PISTOL

SHOOT AT HENRY SHOOT AT HENRY WITH GUN

To avoid frustrating the player, we should make a guess about what the player means whenever we're sure that guess will be reliable (we might, for instance, have only one gun in the story, so we know that SHOOT HENRY will always mean SHOOT HENRY WITH PISTOL), but ask the player for clarification whenever there might be ambiguity (SHOOT PISTOL gives no clue about the target, nor can we safely guess, so we want Inform to ask "What do you want to shoot the pistol at?"). The next section goes into more detail about how to handle these variations.

Conversely, there are cases where the player is offering too *much* information for the command we've defined - say we have a BURN command which doesn't look for a specified fire source, but the player is trying to BURN BOX WITH MATCH. We probably don't want to throw away the extraneous information as though it had never been typed, because the player might have typed something quite specific. BURN BOX WITH ACID, say, should not be cavalierly reinterpreted as BURN BOX (with a fire source). Instead, we want to give the player a bit of gentle guidance, perhaps using "Understand as a mistake", as in

Understand "burn [something] with [text]" as a mistake ("Your choice of lighter isn't important in this story: BURN SOMETHING will suffice.")

Finally, there are some cases where we want to understand a phrase to mean a specific form of a more general action. For instance, we might want TURN DOWN THE MUSIC to mean the same thing as SET VOLUME KNOB TO 1. In this case, we may want to make a sort of dummy action which converts into the main action, as in

Understand "turn down volume" or "turn down music" or "turn down the volume" or "turn down the music" as lowering the volume. Lowering the volume is an action applying to nothing.

Instead of lowering the volume, try setting the volume knob to 1.

More about this can be found later in this chapter, under Remembering, Converting and Combining Actions.

Sometimes these kinds of details can be caught in play-testing, but it's a good idea to think about them specifically and in advance rather than leaving them to our beta-testers to sort out.

(2) To generalize very broadly, there are two possible kinds of command in IF: those that only exist to give the player new information (like EXAMINE, INVENTORY, LOOK, TASTE), and those that change the world model (like TAKE FISH, OPEN DOOR, UNLOCK GATE WITH BLUE KEY). The Inform library has some commands that really do none of these things by default - commands like JUMP that do nothing interesting at all most of the time - but those exist as hooks, in case there is ever something important for them to do.

Commands that ask for information are usually easier to implement. Very often we're looking to offer the player a new kind of information about specific objects, and these can be handled by adding new text properties, as in

A thing has some text called the sacred emanation.

Carry out perceiving something:

say "[sacred emanation of the noun][paragraph break]".

Commands that affect the world model, on the other hand, can range from simple to very complex indeed. Sometimes we need to do nothing more than add an attribute to an object, like

A thing can be folded or flat. A thing is usually flat.

so that our FOLD command can change the object into its folded form. At other times, we need quite intricate rules to account for a subtle multi-stage process - how fire is burning and spreading to objects, say, or how a conversation is progressing. Other parts of the Recipe Book offer solutions to some of these challenges.

(Strictly, we might count a third kind of command: the kind that controls the story itself. The Advanced Actions chapter discusses how to add actions out of world, as these are called, but the difficult ones are already built into Inform - saving, restoring, restarting, undoing a turn, and so on. Mostly when we need to add new actions out of world, they will be help or hint systems of some kind. More about these can be found in the Helping and Hinting section of the Recipe Book, under Out of World Actions and Effects.)

(3) Most commands that change the world require certain preconditions: the player needs to be holding the gun before he can fire it; the gun must be loaded with ammunition; if we're being especially detailed in our simulation, the safety must be off.

Often, there are also subtler details about how the command should interact with special items. For any new command we create, it's worth asking: should anything special happen if the player performs this action...

On himself?

On another living character?

On an object he (or another character) is carrying or wearing?

On an object he (or another character) is inside or on?

On a door?

On an object that is impossible to move (defined as "scenery" or "fixed in place")?

On an intangible object (such as a beam of sunlight)?

On an object far away (such as the sun)?

On an object that is part of something else (such as a doorknob)?

On an object that itself has parts (such as a desk with drawers)?

If there are two objects required by the action, can both the noun and the second noun be the same thing?

For instance, we might have written code so that if the gun is fired at anything but a person or a fragile object, the default response is "The bullet bounces harmlessly off [the second noun]." Our checklist would remind us to write special cases to prevent

SHOOT GUN AT MY SHOE (while he's wearing it)

SHOOT GUN AT ME

SHOOT GUN AT GUN

and so on. Actions that destroy objects are especially tricky, because there are many things that aren't safe to destroy without carefully adjusting the world model. (What happens if we burn a door connecting two rooms? a wooden desk with a drawer containing an asbestos vest? the armchair Cousin Fred is sitting on?)



Start of Chapter 6: Commands



Back to Chapter 5: The Viewpoint Character: §5.6. Viewpoint



Onward to §6.2. Writing New Commands

§6.2. Writing New Commands

Once we've considered all the design issues pertaining to a new action, we're ready to start writing the source text. First we need to give the player a way to issue the command:

Understand "smile" as smiling.
Understand "fold [something]" as folding.
Understand "shoot [something preferably held] at [something]" as shooting it with.
Understand "wrap [something preferably held] in [something preferably held]" as wrapping it in.

(Note how "it" stands in for the first item when we have an action requiring two objects.) The things that go in square brackets are called "tokens": they are blank spaces for the player to fill in with story objects. The different kinds of tokens are explained in the chapter on Understanding.

We can add synonyms with

Understand the command "grin" as "smile".

and we can create reversed versions of commands with

Understand "shoot [something] with [something preferably held]" as shooting it with (with nouns reversed).

These variations are also covered in the Understanding chapter. If the action needs to work on things that aren't within the player's sight or reach in the normal way, we may need to use an [any thing] token (see the Understanding chapter), as in

Understand "contemplate [any thing]" as considering.

We may also need to modify reach or light levels (see Changing reachability and Changing visibility in the Advanced Actions chapter), or rely on the Deciding the scope of... activity.

As for guessing the player's intention when he isn't clear, we may want to consult the Does the player mean rules (to help Inform make guesses between multiple possible targets) and the activities Supplying a missing noun and Supplying a missing second noun (to help Inform guess an appropriate item when the player leaves something entirely out of his command). For instance, if the player typed SHOOT HENRY, it is the supplying a missing

noun/second noun activity that would allow us to make Inform draw the obvious conclusion that he shoots Henry with the pistol in his hand. The Does the player mean rules are discussed in the chapter on Advanced Actions; the activities in the Activities chapter.

Next we need to define our new action, as in

Smiling is an action applying to nothing.
Folding is an action applying to one thing.
Wrapping it in is an action applying to two carried things.

In cases where we're using an "[any thing]" token to let the player affect objects that aren't normally visible or reachable, we'll need to define the action to apply to *visible objects*. This tells Inform that the player doesn't have to be able to touch the object for it to work. So for instance

Considering is an action applying to one visible thing.

For more on this topic, see Visible vs. touchable vs. carried in the Advanced Actions chapter.

The next step is to create rules for Inform to follow when the action happens. These can be check rules (which make sure that the conditions for the action to occur are fulfilled); carry out rules (which perform the action); and report rules (which describe the results of the action to the player). Any new action should have at least a report rule to let the player know what has happened (if anything), and a carry out rule if there are any ramifications for the world model. For instance:

Carry out folding:
now the noun is swan-like.

Report folding:
say "You deftly fold [the noun] into the shape of a swan."

It's important to remember that report rules may be describing something whose name is plural, such as papers or shoes, and write our text so that it sounds right either way; see the chapter on Adaptive Text and Responses.

More about defining actions and creating carry out and report rules may be found in the chapter on Advanced Actions.

Meanwhile, the check rules give us a chance to provide sensible restrictions on how the command works, as in

Check folding:
if the noun is not a napkin:
say "[The noun] won't bend." instead.

Check shooting something with the noun:
say "[The noun] is incapable of aiming at itself." instead.

Check burning something which contains the player:
say "You're not quite desperate enough to make a funeral pyre for yourself just yet."
instead.

The chapter on Advanced Actions explains how check rules work. In the special case where we want the player to take things automatically before using them, we may want to define the action to work only on carried objects, as in

Wrapping it in is an action applying to two carried things.

The activity Implicitly taking something (documented in the Activities chapter) allows us to modify what should happen during this process.

Lastly, a word or two about trouble-shooting. If a newly created command seems not to be working, we can discover what action Inform is really generating with the ACTIONS testing command, as in

```
>actions
Actions listing on.

>i
[taking inventory]
You are carrying nothing.

[taking inventory - succeeded]
```

If the desired command is not happening, we may need to review our understand lines. A common problem is that our new action conflicts with one already defined by default. In that case, we may want to check the Actions index and see whether there are already-defined actions which might conflict with it. If so, we may need to redefine a command with a line like

Understand the command "stand" as something new.

If that's not enough, we can get a comprehensive view of everything that happens during an action with RULES: this will list all the check, carry out, and report rules that Inform is using to perform the command.

★ *See **Memory and Knowledge** for more about the any token and the concept of scope to control what the player may refer to in a command*

-
-  Start of Chapter 6: Commands
 -  Back to §6.1. Designing New Commands
 -  Onward to §6.3. Modifying Existing Commands
-

§6.3. Modifying Existing Commands

Much of the rest of this chapter discusses the behavior of specific commands in Inform's command library, and how we might change and build on these. This section is instead an overview of the general principles: where and how can one intervene?

Whenever we are dealing with actions, the Actions Index is likely to be useful: it lists all the actions currently implemented, whether in our own source or in extensions or the Standard Rules, and lists the rules pertaining to each.

The lightest and easiest way to change behavior is with an Instead rule:

Instead of eating the apple:
say "It turns out to be made of beeswax, so that's a non-starter."

Instead of tasting an edible thing:
say "It's delicious!"
rule succeeds.

The addition of "rule succeeds" tells Inform that the instead action was a success rather than a failure; this is not usually very important with the player's own actions, but can be useful for actions performed by other characters, so that a successfully replaced action is not followed by the disconcerting line

Clark is unable to do that.

Before and After offer alternative easy forms of modification; the Basic Actions chapter explains all three.

Changing the way an action works in all cases is usually better addressed by changing the main rulebook, rather than with one (or many) instead rules. We may add new check, carry out, and report rules to existing action rulebooks. The Advanced Actions chapter describes these, and ends with some guidelines on when to use before, instead, and after, and when to use check, carry out, and report.

Similarly, we may delete, move, or replace rules that are already present (see the chapter on Rulebooks). This is handy if we decide that an action has restrictions that we dislike and want to abolish. If the restriction we need to change is part of the accessibility rules - those which check whether the player can take, see, and touch items - we may need to look at Changing reachability or Changing visibility in the Advanced Actions chapter (to revise what is allowed), at Deciding the scope of something in the Activities chapter (to influence what can be seen when).

If, for instance, the player character is a burly fellow who can lift any other character he likes:

The can't take other people rule is not listed in any rulebook.

...and rip knobs off doors:

The can't take component parts rule is not listed in the check taking rulebook.

...and commit petty theft:

The new can't take people's possessions rule is listed instead of the can't take people's possessions rule in the check taking rulebook.

```
This is the new can't take people's possessions rule:  
if someone (called the owner) carries the noun:  
    say "(first waiting until [the owner] is distracted)";
```

The right approach to use also depends a bit on how systematic a change we anticipate. We may find that instead rules become cumbersome when we want to specify behavior for a very large number of objects. It's fine to have

```
Instead of tasting the arsenic:  
    say "You'll live to regret this very very shortly."  
end the story.
```

but a bit more tedious to have to write

```
Instead of tasting the peppermint: ...  
Instead of tasting the plate: ...  
Instead of tasting the banister: ...  
Instead of tasting the donkey: ...  
(etc.)
```

in a story in which most items have unique flavor descriptions. In that situation, it may be more sensible to overhaul the design of the action: create a new text property for things, and revise "tasting" so that it now consults this property:

The block tasting rule is not listed in any rulebook.

A thing has some text called the flavor. The flavor of a thing is usually "Nothing special."

```
Report tasting something:  
if the flavor of the noun is "Nothing special."  
    say "You taste nothing unexpected." instead;  
otherwise:  
    say "[the flavor of the noun][paragraph break]" instead.
```

```
Report someone tasting something:  
    say "[The actor] licks [the noun]."
```

Finally and most sweepingly, we can rip out whole passages of the Standard Rules and replace them - or not. This is a drastic measure and rarely necessary (or so we hope); but see the Extensions chapter for ways to replace sections of existing source, or even revise the Inform 6 template files on which Inform depends. By these means almost anything can be changed. We can throw out a whole range of existing commands and start from scratch, for instance, if we want Inform to know about a completely new and different command set.

★ *See **Magic (Breaking the Laws of Physics)** for a hat that lets the player walk through closed doors, and an NPC able to reach through solid containers*

-  Start of Chapter 6: Commands
 -  Back to §6.2. Writing New Commands
 -  Onward to §6.4. Looking
 -  Example 214:  **Slogar's Revenge** Creating an amulet of tumblers that can be used to lock and unlock things even when it is worn, overriding the usual requirement that keys be carried.
-

§6.4. Looking

Looking is quite a complicated command, since the production of a room description takes many steps. A detailed description of this process may be found in the Room Descriptions section.

By convention, a player sees full descriptions of rooms he enters more than once, but may type BRIEF in order to see shorter descriptions, and SUPERBRIEF tells the story never to print room descriptions at all. VERBOSE restores the default behavior.

These conventions are not always appropriate, however, especially in works where experiencing a changing environment is essential. The use option

[Use brief room descriptions.](#)

changes the default behavior so that rooms are not always described fully to the player. **Verbosity** demonstrates how this works.

The player always has the option of turning room descriptions to BRIEF or SUPERBRIEF mode. **Verbosity 2** demonstrates how we might remove the player's ability to change the default behavior.

★ See **Room Descriptions** for a detailed description of how Inform creates room descriptions and how to change the results

★ See **Going, Pushing Things in Directions** for ways to change just those room descriptions that are shown as the result of the player's movement

★ See **Memory and Knowledge** for ways to change the room description in response to the player character's knowledge at any given stage of play

-  Start of Chapter 6: Commands
 -  Back to §6.3. Modifying Existing Commands
 -  Onward to §6.5. Examining
 -  Example 3:  **Verbosity 1** Making rooms give brief room descriptions when revisited.
 -  Example 395:  **Verbosity 2** Making rooms give full descriptions each time we enter, even if we have visited before, and disallowing player use of BRIEF and SUPERBRIEF.
-

§6.5. Examining

By default, examining an object shows its description, and - for devices - tells us whether the object is switched on or switched off.

This kind of additional information is not always what we want, so if we have a device whose on/off status we want to conceal, we may write

[The examine described devices rule is not listed in any rulebook.](#)

On the other hand, there are times when we may want to add a similar line or two to the descriptions of other kinds of objects. **Crusoe** allows us to append an "It is charred." sentence to the end of descriptions of things we have burned in the fire. Since it works by introducing a "printing the description" activity, Crusoe is also a good example to start from if we want to introduce more complex, flexible descriptions of items throughout our story.

Odin rewrites the "You see nothing special..." line with other text of our own, for items that otherwise do not have a description.

Finally, we may want to look at multiple things at once. **The Left Hand of Autumn** demonstrates how we might provide a different response for EXAMINE PAINTINGS than for examining each individually; **Beekeeper's Apprentice** provides a SEARCH command that will show the descriptions of all the scenery in the current location.

★ *See **Actions on Multiple Objects** for an alternative EXAMINE ALL command*

-
-  Start of Chapter 6: Commands
 -  Back to §6.4. Looking
 -  Onward to §6.6. Looking Under and Hiding
 -  Example 44:  **Odin** Replacing "You see nothing special..." with a different default message for looking at something nondescript.
 -  Example 68:   **Beekeeper's Apprentice** Making the SEARCH command examine all the scenery in the current location.
 -  Example 295:    **The Left Hand of Autumn** The possibility of using a [things] token opens up some interesting complications, because we may want actions on multiple items to be reported differently from actions on just one. Here we look at how to make a multiple examination command that describes groups in special ways.
 -  Example 337:    **Crusoe** Adding a "printing the description of something" activity.
-

§6.6. Looking Under and Hiding

Finding hidden objects is a classic puzzle in IF. **Beachfront** provides the most basic example, an object that becomes visible only when we have searched the papers on a cluttered desk. **Beneath the Surface** takes this further, giving all large furnishings the ability to conceal items, and allowing the player to put things underneath other things, as well as find them. **Flashlight** adds an extra twist to the puzzle by requiring that the player have a flashlight to shine under a bulky object in order to find what lies underneath.

Looking inside an object is generally handled by the searching action, and we could extend that to allow the player to search multiple or complex objects. **Matreshka** turns the puzzle on its head by allowing the player to search a whole room systematically with only a single command.

★ See **Kitchen and Bathroom** for the related case of needing to look in a mirror

- ⬆ Start of Chapter 6: Commands
 - ⬅ Back to §6.5. Examining
 - ➡ Onward to §6.7. Inventory
 - ⬇ Example 98: ★ **Beachfront** An item that the player can't interact with until he has found it by searching the scenery.
 - ⬇ Example 172: ★ **Matreshka** A SEARCH [room] action that will open every container the player can see, stopping only when there don't remain any that are closed, unlocked, and openable.
 - ⬇ Example 218: ★ **Flashlight** Visibility set so that looking under objects produces no result unless the player has a light source to shine there (regardless of the light level of the room).
 - ⬇ Example 233: ★ **Beneath the Surface** An "underlying" relation which adds to the world model the idea of objects hidden under other objects.
-

§6.7. Inventory

Occasionally we would like to change the way the name of something is printed as part of our inventory, and we can do this with a printing the name rule such as

Rule for printing the name of the dead rat while taking inventory:
say "dead rat (at arm's length)"

There are also several possibilities for redesigning the inventory list as a whole. **Persephone** shows how to divide an inventory list into two parts, a "You are carrying: " section and a "You are wearing: " section. **Equipment List** goes further, and shows how we might use Inform's specialized listing functions to create a variety of differently formatted inventories.

Sometimes the way Inform by default lists properties such as "(closed)" or "(open but empty)" isn't quite what we want. **Oyster Wide Shut** offers a flexible alternative to the standard behavior, allowing finer control over which properties are listed and how they are described.

Trying Taking Manhattan replaces the inventory behavior for other characters: instead of silently looking through their possessions (but not describing them to the player), they now describe to the player what they're carrying and wearing.

- ⬆ Start of Chapter 6: Commands
 - ⬅ Back to §6.6. Looking Under and Hiding
 - ➡ Onward to §6.8. Taking, Dropping, Inserting and Putting
 - ⬇ Example 425: **★ Oyster Wide Shut** Replacing Inform's default printing of properties such as "(closed)", "(open and providing light)", etc., with our own, more flexible variation.
 - ⬇ Example 177: **★★★ Equipment List** Overview of all the phrase options associated with listing, and examples of how to change the inventory list into some other standard formats.
 - ⬇ Example 65: **★★★★ Persephone** Separate the player's inventory listing into two parts, so that it says "you are carrying..." and then (if the player is wearing anything) "You are also wearing..."
 - ⬇ Example 204: **★★★★ Trying Taking Manhattan** Replacing the inventory reporting rule with another which does something slightly different.
-

§6.8. Taking, Dropping, Inserting and Putting

We may want to change the default refusal message when the player tries to pick up scenery: **Replanting** demonstrates this case simply.

Removal modifies responses to successful TAKE commands, with the effect that when the player picks up an item, he gets a response such as "You take the book from the shelf."

Croft modifies the DROP command, so that objects dropped on specific surfaces get reported in a special way. **Celadon** allows the player to drop even objects he is carrying indirectly, for instance on a tray or in a sack.

Morning After introduces a simple rule that changes the behavior of the whole story: whenever the player takes an item he hasn't already looked at, he automatically examines it. This picks up the pace of exploration passages where the player is likely to be collecting a large number of objects.

By default, when the player tries to put or insert an item that he isn't holding, Inform prints a refusal message; **Democratic Process** and **Sand** offer ways instead to have the player first pick up the relevant items. (The former applies to single items the player is trying to place; the latter expands coverage to work even if the player uses a command affecting multiple objects.)

Taking also happens as a result of other commands. Such takes can be made unnecessary by turning off the "carrying requirements rule" under particular circumstances, or presented differently using the implicitly taking activity.

- ⬆ Start of Chapter 6: Commands
 - ⬅ Back to §6.7. Inventory
 - ➡ Onward to §6.9. Going, Pushing Things in Directions
 - ⬇ Example 16: **★ Replanting** Changing the response when the player tries to take something that is scenery.
 - ⬇ Example 90: **★ Morning After** When the player picks something up which he hasn't already examined, the object is described.
 - ⬇ Example 198: **★ Removal** TAKE expanded to give responses such as "You take the book from the shelf." or "You pick up the toy from the ground."
 - ⬇ Example 224: **★ Celadon** Using the enclosure relation to let the player drop things which he only indirectly carries.
 - ⬇ Example 86: **★★★★ Democratic Process** Make PUT and INSERT commands automatically take objects if the player is not holding them.
 - ⬇ Example 202: **★★★★ Croft** Adding special reporting and handling for objects dropped when the player is on a supporter, and special entering rules for moving from one supporter to another.
 - ⬇ Example 379: **★★★★ Lollipop Guild** Overriding the rules to allow the player to show something to another character without first taking it.
 - ⬇ Example 87: **★★★★ Sand** Extend PUT and INSERT handling to cases where multiple objects are intended at once.
-

§6.9. Going, Pushing Things in Directions

Going is the most complex of actions after looking (or perhaps including looking): the success of every movement depends on the direction the player goes; the room he starts from; the room he intends to reach; whether there are any doors intervening (and, if so, whether these are closed or locked); whether he is traveling by vehicle; and whether he is pushing anything in front of him. When he gets there, the description he sees is itself generated by a looking command.

Pushing something in a direction is really a sort of going. The command `>PUSH WHEELBARROW WEST` first checks certain qualifying rules: by default, only things defined as pushable between rooms may be pushed, and they may be pushed only in horizontal directions (not UP or DOWN) -- though these rules can be overridden, as we see in **Zorb**. If the player's pushing attempt passes these criteria, the action is translated automatically into a going action, with all the usual checks about whether that direction leads anywhere, whether a door is in the way, and so on. The converted action afterward can be caught with such rules as

Instead of going to the Alpine Meadow with the wheelbarrow:
say "You don't want to crush the delicate blooms."

Instead of going north with the handcart:
say "The headwind is so stiff that you are unable to make much northerly progress at all while encumbered by the handcart."

Since the two actions are internally being handled as one, both are discussed here.

It is very common for players to make a mistake and type the wrong direction command, or even to misunderstand the room description and not recognize all the possible exits.

Bumping into Walls helpfully adds a facility so that when the player tries to go in the wrong direction, the story lists the correct possibilities, as in

From here, the viable exits are to the south, the east and the west.

Assuming that travel succeeds, another useful technique is to provide some sense of the journey between locations, especially if they are remote from one another or the player has to do something unusual to get from one to the other. **Up and Up** adds a short description of travel when we approach a new room, before the room description is printed; **Veronica**, conversely, adds a comment when the player leaves a region of the map. **The Second Oldest Problem** intervenes and kills a player who tries to travel from one dark room to another. **Mattress King** embellishes the description that automatically results from PUSH MATTRESS WEST, adding a line that describes the player pushing the object before describing the new room approached.

We may also want to add a brief comment when we arrive in a new room, after the room description is printed. One trivial way to do this is to append the line to the room's main description, conditionally, like this:

The Hammock Emporium is a room. "This is Cousin Ed's shop, the big dream he left accounting to pursue. You can't help gawking at the Luxury Leather Space Hammock, made of genuine red buffalo skins[if unvisited]. [paragraph break]So this is why Grampa makes all those 'lying down on the job' jokes every Thanksgiving[end if]."

But often we want our first-glance comment to come after some items in the room are described; and for this effect, we would use the "first look rule" defined in **Saint Eligius**.

If these methods are not enough, the looking action has an action-specific variable called "the room-describing action", which records whether this particular instance of looking comes about because the player typed LOOK or because the player traveled to a new location. We can consult this variable if we want to make looking work differently after going, as for instance here:

Check looking when the room-describing action is the going action:
say "You are temporarily too blinded to see." instead.

Another category of examples treat how we handle the movement commands themselves. The eight compass directions, with UP and DOWN, IN and OUT, are used as standard in most interactive fiction, but they are not the only possible way of navigating, and strike many newcomers to the genre as counter-intuitive, since when strolling around in real life most of us rarely think about our travel in terms of compass orientation. **Misadventure** allows the player to GO TO a named room, instead, and calculates the best route to reach the destination; **Safari Guide** builds on this by letting the player make the whole trip in a single move, automatically opening any doors that stand in his way en route.

In the same spirit of interpreting the player's intentions sensibly, **Provenance Unknown** modifies the pushing command so that if the player pushes the top object in a stack of objects towards a direction, Inform attempts to move the bottom item instead. This is convenient if, for instance, we have a heavy television on a movable cart and want PUSH TELEVISION WEST to work just as well as PUSH CART WEST.

We also sometimes want to respond sensibly to terse movement commands or ones that rely on some knowledge of where the player has already been. **Polarity** provides a GO BACK command, allowing the player to retreat in the direction from which he came, while **Minimal Movement** understands LEAVE, GO, and so on as OUT, in the absence of other information. **Owen's Law** takes this further, calculating from the best routes on a map how to make OUT mean "move towards the exit of this indoor room", and IN mean "proceed further into the interior". **Wonderland** assigns altitudes to all rooms and works out the local best meaning of UP and DOWN accordingly.

★ See **Map** for how to create other kinds of new direction

★ See **Varying What Is Read** for further divisions of the standard compass, such as north-northwest

★ See **Ships, Trains and Elevators** for ship-board directions

★ See **Bicycles, Cars and Boats** for common vehicles in which to travel the map

- ↑ Start of Chapter 6: Commands
- ← Back to §6.8. Taking, Dropping, Inserting and Putting
- Onward to §6.10. Entering and Exiting, Sitting and Standing
- ↓ Example 100: ★ **Veronica** An effect that occurs only when the player leaves a region entirely.
- ↓ Example 105: ★ **Mattress King** Adding extra phrasing to the action to PUSH something in a direction.
- ↓ Example 200: ★ **The Second Oldest Problem** Adapting the going action so that something special can happen when going from a dark room to another dark room.
- ↓ Example 306: ★ **Misadventure** A going by name command which does respect movement rules, and accepts names of rooms as commands.
- ↓ Example 372: ★ **Minimal Movement** Supplying a default direction for "go", so that "leave", "go", etc., are always interpreted as "out".
- ↓ Example 393: ★ **Saint Eligius** Adding a first look rule that comments on locations when we visit them for the first time, inserting text after objects are listed but before any "every turn" rules might occur.
- ↓ Example 6: ★★ **Up and Up** Adding a short message as the player approaches a room, before the room description itself appears.
- ↓ Example 256: ★★ **Wonderland** Hiking Mount Rainier, with attention to which locations are higher and which lower than the present location.
- ↓ Example 307: ★★ **Safari Guide** The same functionality, but making the player continue to move until he reaches his destination or a barrier, handling all openable doors on the way.
- ↓ Example 102: ★★★ **Bumping into Walls** Offering the player a list of valid directions if he tries to go in a direction that leads nowhere.
- ↓ Example 103: ★★★ **Polarity** A "go back" command that keeps track of the direction from which the player came, and sends him back.
- ↓ Example 107: ★★★ **Provenance Unknown** Allowing something like PUSH TELEVISION EAST to push the cart on which the television rests.
- ↓ Example 108: ★★★ **Zorb** Replacing the message the player receives when attempting to push something that isn't pushable, and also to remove the restriction that objects cannot be pushed up or down.
- ↓ Example 179: ★★★ **Owen's Law** OUT always means "move to an outdoors room, or else to a room with more exits than this one has"; IN always means the opposite.

§6.10. Entering and Exiting, Sitting and Standing

Under ordinary circumstances, Inform does not keep track of the player's posture, nor of his exact location in a room. **Lies** implements a room in which the player can lie in different positions on the floor, getting different views as a result.

Our other examples are all modifications of the way Inform handles player movement to make better default guesses at what he wants to do: **Anchorite** adds a GET DOWN and DOWN command that work when the player is on a supporter, to accompany GET UP, GET OFF, and GET OUT (already understood). **Get Axe** makes the player get out of a portable container before attempting to lift it - a consideration that comes up relatively rarely, but that might pertain to inflatable rafts, beanbag chairs, and other lightweight but capacious pieces of furniture.

★ See **Position Within Rooms** for a box the player can push around the room and stand on in different locations

★ See **The Human Body** for letting the player sit, stand, or lie down systematically on furniture or on the floor

★ See **Furniture** for various objects on which the player can sit or stand

↑ Start of Chapter 6: Commands

← Back to §6.9. Going, Pushing Things in Directions

→ Onward to §6.11. Waiting, Sleeping

↓ Example 206: ★ **Get Axe** Changing the check rules to try automatically leaving a container before attempting to take it. (And arranging things so that other people will do likewise.)

↓ Example 290: ★ **Anchorite** By default, Inform understands GET OFF, GET UP, or GET OUT when the player is sitting or standing on an enterable object. We might also want to add GET DOWN and DOWN as exit commands, though.

↓ Example 310: ★ **Lies** Commands to allow the player to lie down in three different ways.

§6.11. Waiting, Sleeping

The standard WAIT command makes time pass at the same rate that it would anyway - one minute per turn. In a story where events happen at specific times of day, though, we might want to give the player more control. **Nine AM Appointment** shows how to give the player a WAIT 10 MINUTES command, while **Delayed Gratification** lets him WAIT UNTIL a specific time of day.

Ordinarily, Inform also refuses to allow the player to SLEEP and WAKE UP: the commands exist, but have no effect. **Change of Basis** lets the player put himself into a sleep state in which he cannot do anything. A somewhat more interesting expansion on this idea would be to let the player sleep and have dreams; there are no examples specifically of dream states, but we might consult the examples on scenes about how to disrupt one environment and move the player to another, entirely new one.

★ See **Scene Changes** for ways to move the player to a new environment such as a dream state

-  Start of Chapter 6: Commands
 -  Back to §6.10. Entering and Exiting, Sitting and Standing
 -  Onward to §6.12. Other Built-In Actions
 -  Example 388:  **Nine AM Appointment** A WAIT [number] MINUTES command which advances through an arbitrary number of turns.
 -  Example 47:  **Change of Basis** Implementing sleeping and wakeful states.
 -  Example 389:  **Delayed Gratification** A WAIT UNTIL [time] command which advances until the game clock reaches the correct hour.
-

§6.12. Other Built-In Actions

Many other actions are themselves very simply implemented and provide only a shell for us to expand on according to the needs of a particular story. Many of these are discussed at more length in sections on various kinds of props and objects; in particular:

- ★ See **Modifying Existing Commands** for ways to override automatic takes or restrictions on what the player must be able to hold or touch
 - ★ See **Sounds** for *LISTEN*
 - ★ See **Barter and Exchange** for *GIVE* and *SHOW*
 - ★ See **Combat and Death** for *ATTACK*
 - ★ See **Saying Simple Things** for *ASK*, *TELL*, and *ANSWER*
 - ★ See **Food** for *TASTE* and *EAT*
 - ★ See **Liquids** for *DRINK*
 - ★ See **Clothing** for *WEAR* and *TAKE OFF*
 - ★ See **Bags, Bottles, Boxes and Safes** for *OPEN*, *CLOSE*, *LOCK*, and *UNLOCK* as applied to containers
 - ★ See **Doors, Staircases, and Bridges** for *OPEN*, *CLOSE*, *LOCK*, and *UNLOCK* as applied to doors
 - ★ See **Furniture** for things the player can *ENTER* and *GET OUT* of
 - ★ See **Money** for *BUY*
 - ★ See **Fire** for *BURN*
 - ★ See **Glass and Other Damage-Prone Substances** for *CUT*
-

-  Start of Chapter 6: Commands
 -  Back to §6.11. Waiting, Sleeping
 -  Onward to §6.13. Magic Words
-

§6.13. Magic Words

Many fantasy games incorporate the idea of magic words that can be spoken. In implementing these, we want to be a bit flexible and accept a range of input regardless of whether the player explicitly speaks the command aloud: `XYZZY`, `SAY XYZZY`, or perhaps even `CAST XYZZY`. The inventively named `Xyzzzy` demonstrates how we might define such a command.

If we want to go even further and to allow the player also to use quotation marks, as in `SAY "XYZZY"`, we may want to include Punctuation Removal by Emily Short, which allows for quotation marks to be stripped out of the player's input before it is understood.

-  Start of Chapter 6: Commands
 -  Back to §6.12. Other Built-In Actions
 -  Onward to §6.14. Remembering, Converting and Combining Actions
 -  Example 287:  `XYZZY` Basics of adding a new command reviewed, for the case of the simple magic word `XYZZY`.
-

§6.14. Remembering, Converting and Combining Actions

Sometimes we want `Inform` to apply a player's action to a different target than the one specified: for instance, directing all (or almost all) commands from the doorknob to the door of which it is a part. `Fine Laid` demonstrates how to do this. Along the same lines, `Lucy` shows how to direct a player's conversation action to apply to a new conversation topic.

We can also record a series of actions performed by the player or by another character.

`Cactus Will Outlive Us All` demonstrates characters each of whom reacts to a very specific provocation; `I Didn't Come All The Way From Great Portland Street` implements a game show in which the player is not allowed ever to repeat an action he has already performed; and `Leopard-skin` implements a maze which the player can escape only by performing a specific sequence of actions.

`Anteaters` provides a peculiar gizmo that can remember actions performed in its presence and force the player to reiterate them.

- ⬆ Start of Chapter 6: Commands
 - ⬅ Back to §6.13. Magic Words
 - ➡ Onward to §6.15. Actions on Multiple Objects
 - ⬇ Example 88: **★ Fine Laid** Making writing that can be separately examined from the paper on which it appears, but which directs all other actions to the paper.
 - ⬇ Example 92: **★ Lucy** Redirecting a question about one topic to ask about another.
 - ⬇ Example 220: **★ Cactus Will Outlive Us All** For every character besides the player, there is an action that will cause that character to wither right up and die.
 - ⬇ Example 430: **★ Leopard-skin** A maze that the player can escape if he performs an exact sequence of actions.
 - ⬇ Example 435: **★ I Didn't Come All The Way From Great Portland Street** In this fiendishly difficult puzzle, which may perhaps owe some inspiration to a certain BBC Radio panel game (1967-), a list is used as a set of actions to help enforce the rule that the player must keep going for ten turns without hesitation, repetition, or deviating from the subject on the card.
 - ⬇ Example 222: **★ Anteaters** The player carries a gizmo that is able to record actions performed by the player, then force him to repeat them when the gizmo is dropped. This includes storing actions that apply to topics, as in "look up anteater colonies in the guide".
-

§6.15. Actions on Multiple Objects

Inform allows a handful of actions - TAKE, DROP, PUT, INSERT - to apply to more than one item at a time, so that the player can move things around easily.

The general principle is that multiple objects are allowed if the actions are likely to be successful but not interesting most of the time, and if they're things that the player could plausibly do all at once. For most actions, the use of ALL would seem weirdly indiscriminate: EAT ALL, say, describes very implausible behavior, and EXAMINE ALL would likely generate a screenful of text at once.

But this is all under our control. To create an action that uses multiples, or to allow the use of multiple objects with an already-existing action, we need to create an understand statement that uses the "[things]" token (note the plural). For instance:

Understand "give [things] to [someone]" as giving it to.

This would let the existing give action apply to multiple objects, in just the same way that "take" does. **Shawn's Bad Day** demonstrates how we might allow EXAMINE ALL to print descriptions of every visible item.

Alternatively, we could generate a new action:

Understand "give [things] to [someone]" as multiply-giving it to. Multiply-giving it to is an action applying to one carried thing and one thing.

(In theory the language here should perhaps be "several carried things" -- but Inform is still going to process multiply-giving item by item, unless we redirect it. More about this in a moment.)

When handling an action that uses the "[things]" token, the parser makes a list of every item to which it is going to apply the action: this is called the multiple objects list. The multiple objects list can be the result of a vague request (GET ALL) or a specific one involving identical multiples (GET PENNIES, GET THREE APPLES) or a very specific one involving unique, named nouns (GET GERBIL, APPLE, AND POMEGRANATE).

We can manipulate what Inform includes in "ALL" in sentences like TAKE ALL with the "deciding whether all includes..." activity; for instance

Rule for deciding whether all includes scenery: it does not.

prevents TAKE ALL from applying to things that can't be moved anyway, avoiding lots of lines like

tree: That's hardly portable.
swing set: That's hardly portable.

A slightly tedious technical note: the multiple objects list is not strictly a list in the standard Inform sense, because it is used so frequently in parsing that it would be cumbersome to handle it with the more flexible but less efficient structure used for lists. However, if we want to manipulate the multiple objects list as though it were an ordinary list -- that is, sort it, rotate it, truncate it, remove entries from it, etc -- we may do so by creating a list like this:

let L be the multiple object list.

and later after making L conform to our desires:

alter the multiple object list to L.

Inform next repeatedly runs the action rulebook for the action generated, using each item from the multiple object list as "noun" in turn (or as "second noun", if that's where the [things] token appeared in the understand line). Since it is possible to alter the multiple object list before the "generate action rule" portion of the turn sequence consults the rulebooks, we can also affect the order in which the player's matched objects are handled; see **Formicidae**. We should not attempt to change the multiple object list after this point, because this is likely to introduce bugs.

Each time Inform tries the action on a new noun, it prefixes the action-attempt with the name of the item it's currently working on. This is where we get such output as "frog eyeballs:" and "newt toes:" in long lists like

frog eyeballs: Taken.
newt toes: Taken.

These names are generated by the "announce items from multiple object lists rule" in the action-handling rules; **Escape from the Seraglio** shows how to alter them. In the context of this rule, the thing we are currently printing the name of can be called "the current item from the multiple object list".

Suppressing names of objects entirely, while occasionally tempting, may have unintended consequences, especially if some of the attempted actions are prevented by check rules that

themselves print things. It is safest to suppress the multiple object names in the case where we already know that the action will succeed wherever it is attempted (more often for observational actions like examining than for manipulative actions like taking, or where we mean to completely override default handling).

Given that our hypothetical "multiply-giving" applies to each given object in turn, it might seem to be useless to create "multiply-giving" as an action different from "giving" -- but the convenience is that manipulating the multiple object list makes it possible to group behavior artificially. The trick here is that, on the first pass of the multiply-giving rulebook, we look at the entire multiple object list, perform actions, print output, and set a flag saying that the action has been handled. The flag tells Inform not to do or print anything for any of the subsequent passes through that action rulebook; thus we artificially create a situation where, instead of performing an action on each object in turn, Inform acts once on the entire group. That allows us to assess the cumulative qualities of the group and have the action respond differently than it might when assessing each item individually.

The Facts Were These demonstrates how we might write an action for GIVE THREE DOLLARS TO MAN or GIVE PIE AND HAT TO MAN where the man would only accept the collective gift when its total proved satisfactory.

Western Art History 305 demonstrates how we might allow EXAMINE, which doesn't normally permit multiple objects, to take them, but to give vaguer responses to a mass examination than an individual one.

★ See **Examining for groups of objects that have a collective description different from their individual descriptions, and for commands that search multiple things at once**

★ See **Dispensers and Supplies of Small Objects** for ways to let the player pick up a number of identical items from a dispenser or supply

-
- ↑ Start of Chapter 6: Commands
 - ← Back to §6.14. Remembering, Converting and Combining Actions
 - Onward to §6.16. Alternate Default Messages
 - ↓ Example 294: ★ **Shawn's Bad Day** Allowing the player to EXAMINE ALL.
 - ↓ Example 327: ★★ **The Best Till Last** Reordering multiple objects for dramatic effect.
 - ↓ Example 328: ★★ **Western Art History 305** Allowing EXAMINE to see multiple objects with a single command.
 - ↓ Example 411: ★★ **Escape from the Seraglio** Replacing the usual response to TAKE ALL so that instead of output such as "grapes: Taken. orange: Taken.", Inform produces variable responses in place of "grapes:".
 - ↓ Example 428: ★★ **Formicidae** Manipulating the order in which items are handled after TAKE ALL.
 - ↓ Example 431: ★★ **The Facts Were These** Creating a variant GIVE action that lets the player give multiple objects simultaneously with commands like GIVE ALL TO ATTENDANT or GIVE THREE DOLLARS TO ATTENDANT or GIVE PIE AND HAT TO ATTENDANT. The attendant accepts the gifts only if their total combined value matches some minimum amount.
-

§6.16. Alternate Default Messages

Often we will want to replace the text produced by Inform by default: this includes quite a wide range of text, much of which either describes the success of a command or explains why the action failed.

Inform provides the Responses system to enable default messages like "You can't go that way" to be changed, and this is capable of making large-scale changes. This is especially useful if we want to give the viewpoint character a distinctive voice and set of mannerisms.



Start of Chapter 6: Commands



Back to §6.15. Actions on Multiple Objects



Onward to §6.17. Clarification and Correction

§6.17. Clarification and Correction

Some commands and some objects raise special challenges when it comes to working out the player's intention.

Sometimes this can be done with good rules about the assumptions Inform should make. **Alpaca Farm** demonstrates a USE command, always a challenge because USE can mean very different actions with different items.

There are also times when we need to ask the player for more information. **Apples** demonstrates how sensibly to use properties to disambiguate between similar objects, while **Walls and Noses** rephrases the disambiguation question when special objects are involved: examining one of the walls of the room will make the story ask "In which direction?" and EXAMINE NOSE will lead to "Whose nose do you mean, Frederica's, Betty's, Wilma's or your own?"

At other times, the player types something that is wrong in a predictable way: for instance, we might want to remove all the "with..." phrases from commands like

```
HIT DOOR WITH FIST
KICK DRAGON WITH FOOT
LOOK WEST WITH EYES
```

and merely parse the remainder of the command. (That last command may be unlikely, but novice players do quite often type commands that refer unnecessarily to body parts.) **Cave-troll** demonstrates how.

WXPQ demonstrates how to modify the error message the parser gives in response to a command it doesn't understand; this particular example focuses on the "That noun doesn't make sense in this context" message that arises from using the "[any thing]" or "[any room]" tokens, but the techniques could be adapted to handling other parser errors as well.

For catching typing errors, Cedric Knight's extension Mistype may also be of use: it provides an automatic typo-correction function that the player can turn on or off.

-
- ⬆ Start of Chapter 6: Commands
 - ⬅ Back to §6.16. Alternate Default Messages
 - ➡ Onward to §6.18. Alternatives To Standard Parsing
 - ⬇ Example 289: ★ **Alpaca Farm** A generic USE action which behaves sensibly with a range of different objects.
 - ⬇ Example 368: ★ **Apples** Prompting the player on how to disambiguate otherwise similar objects.
 - ⬇ Example 380: ★ **WXPQ** Creating a more sensible parser error than "that noun did not make sense in this context".
 - ⬇ Example 370: ★★ **Walls and Noses** Responding to "EXAMINE WALL" with "In which direction?", and to "EXAMINE NOSE" with "Whose nose do you mean, Frederica's, Betty's, Wilma's or your own?"
 - ⬇ Example 424: ★★★ **Cave-troll** Determining that the command the player typed is invalid, editing it, and re-examining it to see whether it now reads correctly.
-

§6.18. Alternatives To Standard Parsing

Very occasionally, for out-of-the-ordinary games, we want to make major changes to the way that Inform ordinarily understands commands.

Cloves shows how we might read adverbs in the player's command: adverbs are challenging because they can legitimately appear anywhere in a command structure, so must be found and accounted for before the rest of the command is understood.

Fragment of a Greek Tragedy goes further, substituting a keyword-recognition parser for the usual structure of commands and objects.

Less drastically, menus of numbered options can temporarily replace or augment standard commands. **Down in Oodville** demonstrates how to add a list of transporter destinations from which the player may choose by numeral.

★ See **Traits Determined By the Player** for ways to ask the player a question at the beginning of play

★ See **Saying Simple Things** for a way to ask the player a yes-no question any time during play

- ↑ Start of Chapter 6: Commands
 - ← Back to §6.17. Clarification and Correction
 - Onward to Chapter 7: Other Characters: §7.1. Getting Acquainted
 - ↓ Example 304: **☆☆ Down in Oodville** Offering the player a choice of numbered options at certain times, without otherwise interfering with his ability to give regular commands.
 - ↓ Example 373: **☆☆ Cloves** Accepting adverbs anywhere in a command, registering what the player typed but then cutting them out before interpreting the command.
 - ↓ Example 374: **☆☆ Fragment of a Greek Tragedy** Responding to the player's input based on keywords only, and overriding the original parser entirely.
-

Examples from Chapter 6: Commands

- ↑ Start of this chapter
- Chapter 7: Other Characters
- ↖ Indexes of the examples

214

☆☆☆☆ Example **Slogar's Revenge**

WI

Creating an amulet of tumblers that can be used to lock and unlock things even when it is worn, overriding the usual requirement that keys be carried.

Under most circumstances, locking and unlocking require the player to be carrying the key he uses to unlock something. This makes sense -- unless the key is on a keychain, or on a chain around his neck, for instance. So here we explore one way to circumstantially override the carrying requirements, while still making sure that the player cannot unlock the door if the unlocking tool is nowhere in sight.

In essence, we are rewriting the carrying requirements rule with a different one of our own devising, and swapping it in only at those moments when it is correct to do so.

"Slogar's Revenge"

Section 1 - Procedure

The amulet carrying rule substitutes for the carrying requirements rule when locking something with the Amulet of Tumblers.

The amulet carrying rule substitutes for the carrying requirements rule when unlocking something with the Amulet of Tumblers.

We can now replace the usual behavior of the carrying requirements rule (to check whether the player is carrying something and, if not, to generate an implicit take) with a similar rule of our own; note that "if the player has the second noun" is a more compact way to write "if the player carries the second noun or the player wears the second noun":

This is the amulet carrying rule:
if the player has the second noun:
 continue the action;
 say "(first picking up the amulet)[command clarification break]";
 try silently taking the second noun;
if the player is not carrying the second noun:
 stop the action;

Section 2 - Scenario

The Daunting Dungeon is a room.

West of the Daunting Dungeon is the Disturbing Door. The Disturbing Door is a door. West of the Disturbing Door is the Fallow Field.

The Disturbing Door is closed and locked.

The player wears the Amulet of Tumblers. The Amulet of Tumblers unlocks the Disturbing Door.

Test me with "unlock disturbing door with amulet / open door / west / remove amulet / close door / lock disturbing door with amulet / drop amulet / unlock disturbing door with amulet".

For a more systematic handling of the keychain problem (and a number of other refinements to the behavior of doors), see the Locksmith extension included with Inform.

3 Example Verbosity 1

WI

Making rooms give brief room descriptions when revisited.

By default, the description of a room is printed every time the player enters a room.

On a device with very limited screen space, however, we might wish to supplant that behavior with "brief" descriptions. In Brief mode, Inform prints room descriptions only when the player enters that room for the first time. Afterwards, the text is skipped, for brevity, though the player can see it again at any time by typing LOOK.

As we saw in the previous chapter, we can set "use options" to control certain aspects of the player's experience. One of the use options is the option to

Use brief room descriptions.

which changes the defaults so that the description of a room is printed only the first time the player enters.

"Verbosity"

Use brief room descriptions.

The Wilkie Memorial Research Wing is a room. "The research wing was built onto the science building in 1967, when the college's finances were good but its aesthetic standards at a local minimum. A dull brown corridor recedes both north and south; drab olive doors open onto the laboratories of individual faculty members. The twitchy fluorescent lighting makes the whole thing flicker, as though it might wink out of existence at any moment.

The Men's Restroom is immediately west of this point."

The Men's Restroom is west of the Research Wing. "Well, yes, you really shouldn't be in here. But the nearest women's room is on the other side of the building, and at this hour you have the labs mostly to yourself. All the same, you try not to read any of the things scrawled over the urinals which might have been intended in confidence."

Test me with "west / east".

If we type "test me" during play, these commands will be carried out automatically, and we can see that when we return to the Research Wing, the description is not given a second time.

Some notes: the player can also turn full-length descriptions on or off with the commands "verbose" and "brief", or set a minimal-description setting with the command "superbrief". This power still belongs to the player even if we have set the use option to show brief room descriptions by default.

Moreover, we can ourselves check what the state of the descriptions is, with

```
if set to sometimes abbreviated room descriptions: ...  
if set to unabbreviated room descriptions: ...  
if set to abbreviated room descriptions: ...
```

Finally, it is possible to exercise more precise control over what the player sees on his first and subsequent visits to a room; see the next example for details.

395

★ Example Verbosity 2

WI

Making rooms give full descriptions each time we enter, even if we have visited before, and disallowing player use of BRIEF and SUPERBRIEF.

Suppose that we want the player always to see full room descriptions, even if he tries to reset the defaults -- perhaps because there is vital information there which he will miss if he turns off full-length room descriptions.

To do this, we might want to remove the standard behavior of the three actions associated with BRIEF, SUPERBRIEF, and VERBOSE, replacing them with explanatory messages about how the game behaves. We cannot use `Instead` to override these actions, because `Instead` rules do not apply to actions out of world. Instead, we will want to remove and replace the carry out rules.

We can do this easily by going to the Actions Index, looking up the detail panel for, say, "preferring abbreviated room descriptions", and click the "unlist" button to paste in the sentence that will remove this rule from the rulebook.

Let's remove all three of the carry out rules and substitute our own:

"Verbosity 2"

Section 1 - Procedure

The prefer unabbreviated room descriptions rule is not listed in the carry out preferring unabbreviated room descriptions rulebook.

The prefer sometimes abbreviated room descriptions rule is not listed in the carry out preferring sometimes abbreviated room descriptions rulebook.

The prefer abbreviated room descriptions rule is not listed in the carry out preferring abbreviated room descriptions rulebook.

Carry out preferring unabbreviated room descriptions:

say "[story title] always provides full-length descriptions for your reading pleasure."

Carry out preferring sometimes abbreviated room descriptions:

say "For your playing protection, [story title] provides only full-length room descriptions."

Carry out preferring abbreviated room descriptions:

try preferring sometimes abbreviated room descriptions instead.

The standard report preferring abbreviated room descriptions rule is not listed in the report preferring abbreviated room descriptions rulebook.

The standard report preferring unabbreviated room descriptions rule is not listed in the report preferring unabbreviated room descriptions rulebook.

The standard report preferring sometimes abbreviated room descriptions rule is not listed in the report preferring sometimes abbreviated room descriptions rulebook.

Use full-length room descriptions.

Section 2 - Scenario

The Wilkie Memorial Research Wing is a room. "The research wing was built onto the science building in 1967, when the college's finances were good but its aesthetic standards at a local minimum. A dull brown corridor recedes both north and south; drab olive doors open onto the laboratories of individual faculty members. The twitchy fluorescent lighting makes the whole thing flicker, as though it might wink out of existence at any moment.

The Men's Restroom is immediately west of this point."

The Men's Restroom is west of the Research Wing. "Well, yes, you really shouldn't be in here. But the nearest women's room is on the other side of the building, and at this hour you have the labs mostly to yourself. All the same, you try not to read any of the things scrawled over the urinals which might have been intended in confidence."

Test me with "west / east / brief / w / e / superbrief / w / e / verbose".

44

★ Example Odin

WI

Replacing "You see nothing special..." with a different default message for looking at something nondescript.

In recent years there has been a strong trend towards providing unique descriptions for all implemented objects. Often this is a good idea, but there are also contexts in which we may want to discourage the player from looking too closely at some things and concentrate his attention on just a few interesting ones.

The trick here is that leaving items completely undescribed leads to rather dull exchanges like this:

```
>x table  
You see nothing special about the table.
```

...which can leave the player with the impression that the author was simply too lazy to describe everything. So it can be a good idea to replace that default message with a different one more appropriate to the game. For instance:

"Odin"

The House of a Mortal Farmer is a room. "Having two separate rooms, this house testifies to considerable wealth and success at agriculture."

The Bedroom is inside from the House.

A chair is a kind of supporter. A chair is always enterable.

In the House are a table, two chairs, a basket, and a hearth. On the table is a loaf of bread.

The description of a thing is usually "You give [the noun] a glance, but it is plainly beneath your attention."

Because the description is attached to a whole kind ("thing"), it is really a blanket instruction about many objects at once. More specific instructions always override less specific ones, so we can easily make exceptions. For instance, the following will work correctly:

The infant is a man in the basket. The description of the infant is "So strong and fat that you wonder whether one of your fellow gods is acquainted with the

mistress of the house-- but it's no concern of yours, of course."

Test me with "x table / x chair / x infant".

68

★★ Example **Beekeeper's Apprentice**

WI

Making the SEARCH command examine all the scenery in the current location.

We have to create a suitable action and say what it does, and to repeat what we do through all the scenery items. That needs material from subsequent chapters, but is quite ordinary Inform all the same:

"Beekeeper's Apprentice"

Studying the vicinity is an action applying to nothing.

Report studying the vicinity:

if the location does not contain something which is scenery:

say "There's little of interest in the [location]." instead;

repeat with point of interest running through scenery in the location:

say "[point of interest]: [run paragraph on]";

try examining the point of interest.

Understand "search" as studying the vicinity.

The Yard is a room.

The hive and the honey are scenery things in the Yard. The description of the hive is "The honeycombed hive is all around you, thrumming with life." The description of the honey is "Wax-sealed honey has been cached in many of the hexagonal nurseries."

Test me with "search".

The reason for this example is to show the use of saying "[run paragraph on]". It means we have output such as:

>search

hive: The honeycombed hive is all around you, thrumming with life.

honey: Wax-sealed honey has been cached in many of the hexagonal nurseries.

Without the running on, the prompts "hive:" and "honey:" would be separated from the descriptions following them, which would look a little odd.

295

★★★ Example **The Left Hand of Autumn**

WI

The possibility of using a [things] token opens up some interesting

complications, because we may want actions on multiple items to be reported differently from actions on just one. Here we look at how to make a multiple examination command that describes groups in special ways.

Suppose that we have a game in which groups of objects can have meaning apart from their individual significance -- perhaps there are spells that can only be cast by collecting just the right items in the same place.

In this case, one of the things the player might like to be able to do is look at several items together and get a special response, different from looking at the items individually.

To make this happen, we need to do several things:

- (1) we need to create a version of the EXAMINE command that can apply to multiple objects at once.
- (2) we need to correct the way Inform normally deals with multiple-object commands, because we want our group description to print only one time, and we want to avoid stubs such as "pear: ... apple: ..." before or after the group description.
- (3) we need to define a way for Inform to identify interesting groups and describe them.

"The Left Hand of Autumn"

Section 1 - Procedure

Understand "examine [things]" or "look at [things]" as multiply-examining.
Multiply-examining is an action applying to one thing.

Understand "examine [things inside] in/on [something]" or "look at [things inside] in/on [something]" as multiply-examining it from. Multiply-examining it from is an action applying to two things.

Group-description-complete is a truth state that varies.

Carry out multiply-examining it from:
try multiply-examining the noun instead.

Check multiply-examining when group-description-complete is true:
stop the action.

Carry out multiply-examining:
let L be the list of matched things;
if the number of entries in L is 0, try examining the noun instead;
if the number of entries in L is 1, try examining entry 1 of L instead;
describe L;
say line break;
now group-description-complete is true.

Before reading a command:
now group-description-complete is false.

Now for step 2, overriding Inform's usual output of names of objects:

The silently announce items from multiple object lists rule is listed instead of the announce items from multiple object lists rule in the action-processing rules.

This is the silently announce items from multiple object lists rule:
unless multiply-examining or multiply-examining something from something:
if the current item from the multiple object list is not nothing, say "[current item from the multiple object list]: [run paragraph on]".

Definition: a thing is matched if it is listed in the multiple object list.

We'll save our "to describe" phrase until Section 2, when we can give the game specific instructions about how to report different lists of objects.

Now, the player might also want to be able to refer to a group of item by some kind of group name, so let's add the option of creating a Table of Collective Names which will interpret these:

After reading a command:
repeat through the Table of Collective Names:
let N be "[the player's command]";
let Y be relevant list entry;
while N matches the regular expression "[name-text entry]":
replace the regular expression "(.*)" [name-text entry] (.*)" in N with
"\1[Y]2";
change the text of the player's command to N.

Report taking something:
say "You pick up [the noun]." instead.

And as a bit of polish, because we'd like SEARCH TABLE to have the same effect as EXAMINE ALL ON TABLE:

Understand "look on [something]" as searching.

Instead of searching something which supports at least two things:
let L be the list of things supported by the noun;
describe L.

Instead of searching something which contains at least two things:
let L be the list of things contained by the noun;
describe L.

Section 2 - Scenario

Eight-Walled Chamber is a room. "A perfectly octagonal room whose walls are tinted in various hues."

The display table is a supporter in the Chamber. A twig of rowan wood is on the table.

The player carries an apple and a pear.

A glove is a kind of thing. A glove is always wearable. Understand "glove" as a glove. The player carries a left glove and a right glove. The left glove and the right glove are gloves.

Now we define a few actual lists of items:

Fruit list is a list of objects which varies. Fruit list is { apple, pear }.
Glove list is a list of objects which varies. Glove list is { right glove, left glove }.
Arcane list is a list of objects which varies. Arcane list is { left glove, twig, pear }.

To describe (L - a list of objects):

```
sort L;  
if L is fruit list:  
    say "Just a couple of fruits."  
otherwise if L is glove list:  
    say "It's a matched pair of fuzzy blue gloves."  
otherwise if L is arcane list:  
    say "To anyone else it might look like a random collection of objects, but  
these three things -- [L with definite articles] -- constitute a mystic key known as  
the Left Hand of Autumn. They practically hum with power."  
otherwise:  
    say "You see [L with indefinite articles]."
```

When play begins:

```
sort fruit list;  
sort glove list;  
sort the arcane list.
```

We sort the lists so that regardless of how we change the rest of the code (and the order in which objects are coded), the resulting list will always be in sorted order and ready to compare with the list of items the player wants to look at. And thanks to the "Reading a command" code we wrote earlier, we can also teach the game to understand the player's references to "the left hand of autumn" as a specific collection of items.

Table of Collective Names

name-text	relevant list
"left hand of autumn"	"[arcane list]"
"gloves"	"[glove list]"
"pair of gloves"	"[glove list]"

Test me with "x apple and pear / x left and right / put pear on table / put left glove on table / x all on table / put all on table / examine all on table / get apple, twig, pear / x all on table / search table".



Suppose we want to add rules so that any time we examine a charred object (and most of our objects can be charred), a line about the charring is appended to the end of the object description. We could use "after examining...", but perhaps we would prefer for the sentence about the charring not to appear in its own paragraph.

This is an ideal occasion for a new activity. We look at the action index for "examining" to identify the rule that causes the old behavior (in this case, the "standard examining rule"); replace this with a new rule that calls our activity; and write our "printing the description" activity in such a way that it uses an object's description without forcing a paragraph return afterward.

Then we will use "after printing the description" to add our line about charring, and make sure that the paragraph return does occur before the prompt.

So:

"Crusoe"

Section 1 - Creating our New Activity

The fancy examining rule is listed instead of the standard examining rule in the carry out examining rules.

This instruction replaces a normal piece of the examine action, the standard examining rule, with another one of our own devising. (The replacement of the standard examining rule will be explained in more detail in the chapter on rulebooks.)

Printing the description of something is an activity.

This is the fancy examining rule:

carry out the printing the description activity with the noun;
rule succeeds.

All we have done here is enclose what is usually just a rule inside an activity. This means that we can now write before and after rules for the activity, and also add special instructions like "Rule for printing the name of something while printing the description of something" -- this may not be likely to arise often, but Inform now has the concept of "printing the description of something" as a separate context of action. Next we add the modification that lets us append to the description without a new line:

Rule for printing the description of something (called item):

if the description of the item is not "":

say "[description of item] [run paragraph on]";

otherwise:

say "You see nothing special about [the item]. [run paragraph on]".

"run paragraph on" here will mean that we do not get a paragraph break following the description, even if it ends with a period. We also insert a space, so that our follow-on comments will be properly punctuated.

After printing the description of something charred:
say "It is charred." instead.

The instead at the end of this line stops Inform for going on with any other "after printing the description of..." rules.

The standard library also has rules for printing additional text about containers and supporters with visible contents, and devices that are switched on; with this current system, we could add those as "after printing the description" rules as well, building up a complete paragraph if we wanted. But for simplicity we won't exemplify all of that here. The effects would be much the same as with the "charred" line.

Now, because we want to make sure that we always do get a paragraph break after our description, we add this rule last after all the other rules. "Last" and "first" rules are covered in more detail in the chapter on rulebooks.

Last after printing the description of something:
say paragraph break.

Section 2 - The Scenario

The Desert Isle is a room. "A pale expanse of sand, here and there developing into hillocks of grass, and a small clump of palms. The water is shallow here, and there are other islands within swimming distance -- or even wading distance, perhaps -- but none of them is any larger than your island, so it doesn't seem worth the trouble of visiting.

A few hundred feet out, the water turns darker blue, the sea floor drops away, and there is nothing to be seen all the way down to the horizon, except a couple of fluffy clouds, and an occasional bird.

The remains of your fire smolder in the stone-lined pit."

A thing can be charred or whole. A thing is usually whole. Instead of burning something: say "You hold [the noun] to the fire until it flares and chars."; now the noun is charred.

The player carries a stick. The description of the stick is "A strip of palm from the woodiest part of the leaf, about a foot and a half long."

The player carries a glass bottle and a piece of paper. The description of the paper is "A single blank sheet." In the glass bottle is a grain of sand. The glass bottle is openable and open. Instead of burning the glass bottle: say "You hold the bottle to the flame, but it grows uncomfortably warm."

Instead of burning the grain of sand: say "You drop the grain into the fire pit, where it becomes indistinguishable from all the others."; now the grain of sand is nowhere. Instead of dropping the grain of sand: now the grain of sand is nowhere; say "You return the grain of sand to its brethren."

The player's description is handled in an unusual way, and this will produce a space paragraph break there where it should not. Instead, therefore, we will add an instead for examining the player (probably a good idea anyway):

Instead of examining the player:

say "You are sunburned and there is sand in cracks you didn't know existed."

Test me with "i / x stick / x bottle / x sand / x paper / x me / burn stick / x stick / burn paper / x paper".

The "printing a description" activity may be useful for other games, and can be imported just by lifting section 1.

98

Example Beachfront

WI

An item that the player can't interact with until he has found it by searching the scenery.

Suppose we have our player, a detective, searching for evidence; we don't want him to be able to use this evidence until he has performed the action that reveals it, but after that it should be visible in the room when he looks.

A simple way to do this is to start the object -- an envelope, in this scenario -- out of play, and only move it into the location when the player looks for it:

"Beachfront"

The Stuff Office is a room. "The windows are closed, making the sultry air even more unbearable. A narrow slice of Caribbean blue is visible between the scuba gear rental shop and the recreated 17th century pirate tavern.

The office is cheerfully furnished with wicker chairs and white curtains, but the tropical decorating scheme stopped at the desk, which is heavy oak and absolutely covered with papers."

The heavy oak desk is a supporter in the stuffy office. It is scenery. Understand "paperwork" as the desk.

The creamy envelope is an openable container. The description is "There is no return address on the outside of the envelope, just the address of the Doctor's office -- but the legs of the capital A are rubbed down in a characteristic way, and the top of every R is open. There's no question that it comes from the same typewriter as the blackmail note." In the envelope is a letter. The envelope can be found or lost. The envelope is lost.

Instead of searching the desk when the envelope is lost:

now the envelope is found;

say "You rifle through the piles of bills and notices; invitations to conventions; advertisements for high-end prescription drugs; pink carbon sheets bearing patients[] names and medical identification numbers in spidery, elderly handwriting. Almost at the bottom of the heap, you find what you were looking for: a creamy envelope with the address typed.";

move the envelope to the desk.

Here we've changed the property of the envelope to keep track of the fact that it has been found, so that if the player tries again, he won't find anything more.

Instead of searching the desk:

say "Further investigation of the desk reveals nothing else suspicious."

Notice that we have two rules that apply to "searching the desk", but one of them has a more specific set of parameters ("when the envelope is lost"). This means that Inform will consult that rule first and use it if it applies; it will only carry out our plain vanilla "instead of searching the desk" rule when the more restricted rule is not relevant.

Test me with "x envelope / x desk / search desk / look / get envelope / x envelope".

172

Example Matreshka

WI

A SEARCH [room] action that will open every container the player can see, stopping only when there don't remain any that are closed, unlocked, and openable.

"Matreshka"

Ransacking is an action applying to one thing.

Check ransacking:

if the noun is not the location, say "You can hardly search [the noun] from here." instead.

Carry out ransacking:

while the player can see a closed openable unlocked container (called target):
say "[target]: [run paragraph on]";
try opening the target.

Report ransacking:

say "You can see nothing further worth searching."

The Russian Gift Shop is a room. In the Russian Gift Shop is a large wooden doll. It is closed and openable. In the large wooden doll is a medium wooden doll. It is closed and openable. In the medium wooden doll is a small wooden doll. It is closed and openable. In the small wooden doll is a tiny solid wooden doll.

And now we need to borrow from a later chapter for the command that will make this work:

Understand "search [any visited room]" as ransacking.

Test me with "search gift shop".

★ Example Flashlight

Visibility set so that looking under objects produces no result unless the player has a light source to shine there (regardless of the light level of the room).

"Flashlight"

The Schoolhouse is a room. "Though normally comfortable, the room is dark and menacing during the storm; rain sheets on the windows, and you can barely see the flash of the lighthouse only a few miles away."

The cabinet is a fixed in place openable container in the Schoolhouse. The hurricane lantern is a thing in the Schoolhouse. "A hurricane lantern hangs from a peg." The lantern is lit.

Visibility rule when looking under something:
 if the player is carrying a lit thing (called lamp):
 say "You shine [the lamp] under [the noun]...";
 there is sufficient light;
 there is insufficient light.

There is a marble. The marble can be found or lost. The marble is lost.

Instead of looking under the cabinet when the marble is lost:
 move the marble to the player;
 now the marble is found;
 say "Billy's lost marble! So that's where it got to!"

Test me with "look under cabinet / get lantern / look under cabinet".

Because visibility is checked before instead rules, this discovery will (correctly) occur only when the player does have enough light.

★ Example Beneath the Surface

An "underlying" relation which adds to the world model the idea of objects hidden under other objects.

The standard world model provides for the idea of containers and supporters, but this is not the only way that objects can relate to one another in the real world. Here we try adding the idea of concealment beneath another object:

"Beneath the Surface"

Section 1 - In Which our Terms are Defined

Underlying relates various things to one thing. The verb to underlie means the underlying relation. The verb to be under means the underlying relation. The

verb to be beneath means the underlying relation.

Instead of looking under a thing which is underlaid by something (called the lost object):

say "You find [the list of things which underlie the noun]!";
now every thing which underlies the noun is carried by the player;
now every thing which underlies the noun does not underlie the noun.

Hiding it under is an action applying to one carried thing and one thing. Understand "put [something preferably held] under [something]" as hiding it under. Understand "hide [something preferably held] under [something]" as hiding it under. Understand the commands "shove" and "conceal" and "stick" as "hide".

Check hiding it under:

if the second noun is not fixed in place, say "[The second noun] wouldn't be a very effective place of concealment." instead.

Carry out hiding it under:

now the noun is nowhere;
now the noun underlies the second noun.

Report hiding it under:

say "You shove [the noun] out of sight beneath [the second noun]."

Section 2 - In Which They are Put To Use

The Room of Hidden Objects is a room. It contains a sofa, an easy chair, and a rug. The sofa supports a lime-green pillow and an innocent-looking Chinese finger toy. The rug is fixed in place. The chair is a supporter.

A treasure map underlies the easy chair. A skeleton is beneath the sofa. A blueprint of Atlantis, a lexicon of Linear A, and the key to Jimmy Hoffa's Mausoleum are under the rug.

Test me with "look under the sofa / look under the rug / look under the easy chair / hide lexicon under rug".

425

★ Example Oyster Wide Shut

WI

Replacing Inform's default printing of properties such as "(closed)", "(open and providing light)", etc., with our own, more flexible variation.

As we've seen in earlier examples such as "Equipment List", it is possible to vary the way Inform creates inventory listings in general (to create lists that look more like paragraphs of prose, lists divided between what the player is wearing and what he isn't, and so on). We can also use activities to alter the printing of specific objects' names and contents, as with the "omit contents in listing" feature after printing the name of something.

We may find, however, that we would like a great deal more control over Inform's printing of inventory details, not just as a special effect for a few items, but throughout the game.

We start by turning off Inform's native property writer:

"Oyster Wide Shut"

Section 1 - Procedure

The print standard inventory rule is not listed in any rulebook.

Carry out taking inventory (this is the new print inventory rule):

say "You are carrying: [line break]";
list the contents of the player, with newlines, indented, including contents, with extra indentation.

This is very much like the library's standard behavior, but with the exception that "giving inventory information" or even "giving brief inventory information" are omitted. Here's how we supplant it:

After printing the name of something (called target) while taking inventory:
follow the property-aggregation rules for the target.

Now, our property-aggregation rulebook is going to look at a given object and decide on a list of features that should be mentioned in inventory. We'll start by producing something quite similar to Inform's default behavior:

The property-aggregation rules are an object-based rulebook.
The property-aggregation rulebook has a list of text called the tagline.

A first property-aggregation rule for an openable open thing (this is the mention open openables rule):
add "open" to the tagline.

A first property-aggregation rule for an openable closed thing (this is the mention closed openables rule):
add "closed" to the tagline.

A property-aggregation rule for a closed transparent container which contains nothing (this is the mention empty transparent containers rule):
add "empty" to the tagline.

A property-aggregation rule for an open container which contains nothing (this is the mention empty open containers rule):
add "empty" to the tagline.

A property-aggregation rule for a lit thing (this is the mention lit objects rule):
add "providing light" to the tagline.

A property-aggregation rule for a thing worn by the player (this is the mention worn objects rule):
add "being worn" to the tagline.

```
The last property-aggregation rule (this is the print aggregated properties rule):
  if the number of entries in the tagline is greater than 0:
    say " ([tagline])";
    rule succeeds;
  rule fails.
```

Notice that we don't need to write any rules about how to print that list of text: because Inform is printing out a list, it will automatically insert commas, spaces, and the word "and" where appropriate; and it will automatically follow the "use serial comma" option, if we have it set.

Now we're free to meddle. Let's give the player a bunch of possessions that will be listed in interesting ways in inventory:

Section 2 - Scenario

```
The Curved Beach is a room. "White sand stretches away both northeast and
northwest, enclosing this attractive little bay. Gentle waves lap at the beach."
```

```
The player carries a glowing plastic sack. The glowing plastic sack is lit and
transparent and openable and open. It contains a rock. It is wearable.
```

```
The player wears a flashlight lanyard. The flashlight lanyard is a device.
```

```
Carry out switching on the lanyard: now the noun is lit.
Carry out switching off the lanyard: now the noun is unlit.
```

```
The player carries an oyster. The oyster contains a pearl. The oyster is
openable.
```

Now suppose that we don't want the oyster to say "closed" when it's closed. Instead, we'd like it to say "clamped shut". As this is the only property the oyster will ever have, we can simply override his whole property-aggregation rulebook:

```
A property-aggregation rule for the oyster:
  if the oyster is closed:
    say " (clamped shut)";
  rule succeeds.
```

That's fine for the oyster because "clamped shut" is the only property he'll ever have. What if we'd like instead just to revise the way the sack (and only the sack) gets described as providing light?

```
The sacklight rule is listed after the mention lit objects rule in the property-
aggregation rules.
```

```
A property-aggregation rule for the plastic sack (this is the sacklight rule):
  if "providing light" is listed in the tagline:
    remove "providing light" from the tagline;
    add "gently glowing" to the tagline.
```

Now the flashlight (and any other regular light sources we might add to the game) will be described as "providing light", but the sack will only be said to be gently

glowing -- a bit more appropriate for its rather fainter gleam.

We might also wish to add a systematic feature across the board to include a new property in the inventory list? Let's say the player can enchant his possessions, and enchanted possessions should thereafter be listed accordingly:

A thing can be magical or non-magical. A thing is usually non-magical.

Understand "enchant [something]" as enchanting. Enchanting is an action applying to one thing.

Carry out enchanting something:
now the noun is magical.

Report enchanting something:
say "Ding! You turn [the noun] magical."

A property-aggregation rule for a magical thing:
add "enchanted" to the tagline.

Test me with "i / close oyster / i / turn on flashlight / i / take off flashlight / i / turn off flashlight / i / close sack / i / open sack / i / take all from sack / i / close sack / i / wear sack / i / enchant sack / i / open sack / put all in sack / i / close sack / i".

Further variations are possible as well: if we used a "before printing the name..." rather than an "after printing the name..." rule, we could automatically generate lines like "an open and empty phosphorescent plastic sack", removing some of the artificiality of the parentheses.

Or we could add more logic to the rules about which properties are mentioned, so that some features of objects were mentioned in inventory only if the player was wearing the correct detection device, like so:

The player wears enchantment-detecting goggles.

A property-aggregation rule for a magical thing:
if the player is wearing the goggles:
add "enchanted" to the tagline.



Overview of all the phrase options associated with listing, and examples of how to change the inventory list into some other standard formats.

Most of the phrase options above are relatively self-explanatory; a few are less so. Here is an overview:

"With newlines" tells Inform to put a new line before each listed object. Indented tells it to indent contents of objects, when listing these.

"Giving inventory information" means to append information such as (closed) or (being worn) to objects.

"As a sentence" means to put "and" before the last object and commas between them; this is usually not used in conjunction with newline listing. "As a sentence" obeys whatever conventions about the use of the serial comma we may have established with the "Use serial comma" option.

"Including contents" means to list the contents of open or transparent containers and all supporters, whereas including all contents means to list the contents of all containers, even opaque closed ones.

"Tersely", perhaps unexpectedly, puts parentheses around objects listed as the contents of other objects.

"Giving brief inventory information" omits most of the inventory tags, such as "(open)" and "(worn)", but does list "(closed)" for closed containers which might not otherwise be obviously openable.

"Using the definite article" means prefixing objects with "the", if applicable, rather than "a".

"Listing marked items only" means including only objects that have already been declared "marked for listing".

"Prefacing with is-are" means that Inform will write "is" before the list if it contains only one item, and "are" if the list contains more than one.

"Not listing concealed items" means to omit from the list anything which is scenery.

Finally, "with extra indentation" means that the whole list should be indented slightly, in emulation of the default inventory listing.

With this information, we can try rewriting the inventory behavior to emulate the standard or to explore alternate versions:

"Equipment List"

The Watery Room is a room. The player carries a snorkel and a waterproof sack. The waterproof sack contains an undersea map, a diving guide, a cup, and 500 Argentine pesos. The cup contains a worm. The player wears a swimsuit and a pair of flippers. The sack is openable and open.

Inventory listing style is a kind of value. The inventory listing styles are tall, wide, curt, minimal, divided tall, and divided wide. Current inventory listing style is an inventory listing style that varies.

Understand "inventory [inventory listing style]" as requesting styled inventory. Requesting styled inventory is an action applying to an inventory listing style. It is an action out of world.

Carry out requesting styled inventory:
now current inventory listing style is the inventory listing style understood.

Report requesting styled inventory:
say "Inventory listing is now set to [current inventory listing style]."

We begin by emulating the standard inventory listing style:

Instead of taking inventory when current inventory listing style is tall:
if the number of things enclosed by the player is 0, say "You are empty-handed." instead;
say "You are carrying: [line break]";
list the contents of the player, with newlines, indented, giving inventory information, including contents, with extra indentation.

Here we offer the alternative of listing everything together as a paragraph:

Instead of taking inventory when current inventory listing style is wide:
if the number of things enclosed by the player is 0, say "You are empty-handed." instead;
say "You are carrying ";
list the contents of the player, giving inventory information, as a sentence, including contents;
say "."

This may be unsatisfactory, however. Items that are inside other items are not set off from those merely carried by the player. One way around this is to use terse listing, giving such descriptions as "a waterproof sack (in which are an undersea map, a diving guide, a cup (in which is a worm) and a 500 Argentine pesos)" as opposed to the more confusing " a waterproof sack (open), inside which are an undersea map, a diving guide, a cup, inside which is a worm and a 500 Argentine pesos".

Instead of taking inventory when current inventory listing style is curt:
if the number of things enclosed by the player is 0, say "You are empty-handed." instead;
say "You are carrying ";
list the contents of the player, tersely, giving brief inventory information, as a sentence, including contents;
say "."

If, using the above style, we close the sack, we will still get "(closed)" after the sack's listing. The following minimalist listing style abolishes even that nicety:

Instead of taking inventory when current inventory listing style is minimal:
if the number of things enclosed by the player is 0, say "You are empty-handed." instead;
say "You are carrying ";
list the contents of the player, tersely, as a sentence, including contents;
say "."

If we want to list worn things separately from carried things, we have occasion to put "listing marked items only" to work:

Instead of taking inventory when the current inventory listing style is divided wide:
if the number of things enclosed by the player is 0, say "You are empty-handed." instead;

```

say "You are wearing ";
now all things enclosed by the player are unmarked for listing;
now all things worn by the player are marked for listing;
if no things worn by the player are marked for listing, say "nothing";
otherwise list the contents of the player, as a sentence, listing marked items
only;
say ".[paragraph break]";
say "You are carrying ";
now all things carried by the player are marked for listing;
now all things worn by the player are unmarked for listing;
if no things carried by the player are marked for listing, say "nothing";
otherwise list the contents of the player, as a sentence, tersely, giving brief
inventory information, listing marked items only;
say ".[paragraph break]".

```

And similarly for a tall divided inventory:

```

Instead of taking inventory when the current inventory listing style is divided tall:
if the number of things enclosed by the player is 0, say "You are empty-
handed." instead;
if the player carries something:
now all things enclosed by the player are unmarked for listing;
now all things carried by the player are marked for listing;
say "You are carrying: [line break]";
list the contents of the player, with newlines, indented, giving inventory
information, including contents, with extra indentation, listing marked items only;
if the player wears something:
now all things enclosed by the player are unmarked for listing;
now all things worn by the player are marked for listing;
say "You are wearing: [line break]";
list the contents of the player, with newlines, indented, including contents,
with extra indentation, listing marked items only.

```

Test me with "i / inventory wide / i / inventory curt / i / close sack / i / open sack /
inventory minimal / i / close sack / i / open sack / inventory divided wide / i /
inventory divided tall / i / drop all / i / take all / take off swimsuit / take off flippers /
i / i divided wide / i / wear swimsuit / drop all / i".

☆☆☆ **Example Persephone**

Separate the player's inventory listing into two parts, so that it says "you are carrying..." and then (if the player is wearing anything) "You are also wearing..."

If we wanted, we might replace the rule for taking inventory as follows:

"Persephone"

```

Instead of taking inventory:
say "[if the player carries something][We][re] carrying [a list of things carried
by the player][otherwise][We][re] empty-handed";
say "[if the player wears something]. [We][re] wearing [a list of things worn by
the player][end if].".

```

The Fancy Party is a room. The player carries a sword, a strawberry stem, and 20 credits worth of platinum. The player wears a sash indicating lordhood.

Test me with "i / take off sash / i / drop all / i".

204



Example Trying Taking Manhattan

WI

Replacing the inventory reporting rule with another which does something slightly different.

Inform has built-in commands for other people, and sometimes we may want to adjust the way these work without completely disabling and replacing the command. Suppose, for instance, that instead of

Kermit the Frog looks through his possessions.

we'd like someone taking inventory to report what he's got, thus:

Kermit the Frog says, "I seem to be carrying a microphone and wearing a hat and a trenchcoat."

To do this, we could replace the built-in report rule with a different one.

"Trying Taking Manhattan"

The loud inventory rule is listed instead of the report other people taking inventory rule in the report taking inventory rules.

This is the loud inventory rule:

unless the player is the person asked:
say "[The person asked] says, 'I seem to be carrying [a list of things carried by the person asked][if the person asked is wearing something] and wearing [a list of things worn by the person asked][end if].'"

Persuasion rule for asking someone to try doing something: persuasion succeeds.

Grand Central Station is a room. "Here you are in New York, New York. Any minute now someone is going to burst into song."

Kermit the Frog is a man in Grand Central Station. "Kermit the Frog stands nearby, enjoying being green." Kermit is wearing a hat and a trenchcoat. He is carrying a microphone.

Test me with "inventory / kermit, inventory".

16



Example Replanting

WI

Changing the response when the player tries to take something that is

scenery.

By default, "TAKE OAK" in the example above will produce the response "That's hardly portable." This is fine under many circumstances, but also a bit generic, so we might want to override it for a specific game.

"Replanting"

The Orchard is a room. "Within this quadrille of pear trees, a single gnarled old oak remains as a memory of centuries past." The gnarled old oak tree is scenery in the Orchard.

Instead of taking some scenery: say "You lack the hulk-like strength."

Test me with "take oak".

Here we've used an "instead" rule; we will learn more about these in the section on actions. This allows us to define our own results for taking an object.

Note: "scenery" is a property of an object (about which we will hear more later). So when we use it in rules, we can talk about "some scenery", "something that is scenery", or even "a scenery thing" -- the last one doesn't sound much like English, but is a more plausible construction with other adjectives.

90

Example Morning After

WI

When the player picks something up which he hasn't already examined, the object is described.

Suppose we want to make the player's life slightly easier by examining everything he picks up, if he hasn't already examined it.

"Morning After"

A thing can be examined or unexamined.

After taking something unexamined:
say "Taken. [run paragraph on]";
try examining the noun.

Carry out examining something:
now the noun is examined.

Carry out rules are explained in more detail in the chapter on advanced action handling. For now, it may be enough to know that what we put into this carry out rule for examining will happen any time anything is examined, but that it will not interfere with the rest of the predefined behavior of the action. The player will still see the object description and so on, as usual.

The Red Door Saloon is a room. "This old place is in pretty bad shape since the mine shut down. Now there's not much to see but the pair of deep gouges in the floorboards where they dragged away the Sheriff's corpse with the spurs still on."

Jed is a man in the Red Door Saloon. "At 8:30 AM the only person around is old Jed, collecting his hangover cure."

The pistol is a thing in the Red Door Saloon. The description of the pistol is "It ain't too accurate, but for two dollars you can't expect much."

The hangover cure is a thing in the Red Door Saloon. The description of the hangover cure is "Two yellow egg-yolks unbroken in a red-brown liquid. Yep."

Test me with "x pistol / get all".

198

Example Removal

WI

TAKE expanded to give responses such as "You take the book from the shelf." or "You pick up the toy from the ground."

Suppose that we want to change the reporting of "take" so that the player is always told something like "You take the book from the shelf." or "You pick up the toy from the ground." In order to generate these reports, we will need to know where the object started, even though by the time we are printing the output, the object will have moved.

"Removal"

The Pharmacy is a room. A desk and a trash can are in the Pharmacy. The pill-counter, a prescription, and a computer are on the desk. The computer is fixed in place. The pill-counter contains some Vicodin. The trash can contains an empty box.

The taking action has an object called previous locale (matched as "from").

The previous locale could in theory be either a thing or a room, so we make it "an object" -- that is, the most generic possible kind, to which both things and rooms belong. Now we record what the previous locale is at the beginning of each taking action:

Setting action variables for taking:
now previous locale is the holder of the noun.

Report taking something from the location:
say "You pick up [the noun] from the ground." instead.

Report taking something:
say "You take [the noun] from [the previous locale]." instead.

Test me with "get all".

★ Example Celadon

Using the enclosure relation to let the player drop things which he only indirectly carries.

By default, Inform only lets the player drop those things which he is carrying -- that is, those directly in his possession. Things inside satchels or on portable trays have to be taken first.

If we want to change this behavior, we might add a dropping rule that distinguishes between carrying and mere enclosure (introduced back in "The location of something" in the chapter on Things):

"Celadon"

The Tea Room is a room. The player carries a black lacquer tray. The lacquer tray is portable. On the lacquer tray are a celadon teapot and a napkin.

Before dropping something:

if the player does not carry the noun and the player encloses the noun:
 say "(first taking [the noun] from [the holder of the noun])[command clarification break]";
 silently try taking the noun;
 if the player does not carry the noun, stop the action.

Instead of taking the napkin:

say "It seems to be stuck to the tray, possibly by an underlying wad of gum."

Test me with "i / drop teapot / i / look / drop teapot / drop napkin / i / drop tray".

★★★ Example Democratic Process

Make PUT and INSERT commands automatically take objects if the player is not holding them.

"Stop" and "Continue" are most useful when we need to write rules that will have to stop the action some of the time but at other times let it pass; so for instance:

"Democratic Process"

Before inserting something which is not carried by the player into something:

if the noun is in the second noun, say "Already done." instead;
 say "(first taking [the noun])[line break]";
 silently try taking the noun;
 if the player is not holding the noun, stop the action.

Before putting something which is not carried by the player on something:

if the noun is on the second noun, say "Already done." instead;

say "(first taking [the noun])[line break]";
silently try taking the noun;
if the player is not holding the noun, stop the action.

The Assembly Room is a room. "On most days, this room is used for elementary school assemblies; at the moment, it serves as a voting place." The ballot is on the desk. The desk is in the Assembly Room.

The machine is a container in the Assembly Room. "On the ballot machine is a sign which reads 'PUT BALLOTS IN ME :)'. Understand "ballot machine" as the machine.

Test me with "put ballot in machine".

202



Example Croft

WI

Adding special reporting and handling for objects dropped when the player is on a supporter, and special entering rules for moving from one supporter to another.

Suppose that we have a design in which the player spends lots of time on enterable supporters, and in which we want to report certain actions -- dropping things onto those supporters, or leaping from one to another -- in a new way. We might begin by adding some action variables to help us keep track of the situation:

"Croft"

The dropping action has an object called the container dropped into (matched as "into").

The dropping action has an object called the supporter dropped onto (matched as "onto").

Rule for setting action variables for dropping:

if the actor is in a container (called C), now the container dropped into is C;
if the actor is on a supporter (called C), now the supporter dropped onto is C.

Report dropping a heavy thing onto a metallic thing:

say "You drop [the noun], and [the supporter dropped onto] clangs protestingly." instead.

Report someone dropping a heavy thing onto a metallic thing:

say "[The actor] drops [the noun] onto [the supporter dropped onto], which clangs protestingly." instead.

A thing can be heavy or light. A thing can be metallic or ordinary. A thing is usually ordinary. A thing is usually light.

The Ancient Cambodian Temple is a room. "A vast space built for ancient and forgotten rituals. The stone floor crawls with vermin. Well above the floor, and

separated by some feet, are twin platforms built into the wall: the one carved of jointed wood, the other of sheets of graven bronze."

A platform is a kind of supporter. A platform is always enterable. A platform is usually scenery.

The bronze platform is a metallic platform in the Temple. Lara is a woman. She is on the bronze platform. She wears safari pants and a tank top. She carries a gun and a map. The gun is heavy.

The wood platform is an ordinary platform in the Temple. The player is on the wood platform. The player carries a rope, an Ancient Cambodian/English Phrasebook, a pickaxe, and a precious idol. The idol and the pickaxe are heavy.

Persuasion rule: persuasion succeeds.

The entering action has an object called the place left (matched as "from").

Check entering a platform from a platform:

if actor is the player, say "You leap into midair to cross the distance...";

otherwise say "[The actor] leaps gracefully across the distance...";

move the actor to the holder of the noun, without printing a room description.

Because this rule occurs before the "implicitly pass through other barriers rule", that rule will not occur when we move from platform to platform; we'll use our own custom rule instead.

Rule for setting action variables for entering:

now the place left is the holder of the actor.

Report entering a platform from a platform:

say "You land in a cat-like crouch on [the noun]." instead.

Report Lara entering a platform from a platform:

say "Lara lands soundlessly on [the noun][if the noun supports the player] beside you[end if]." instead.

Report entering a platform from the location:

say "You jump, catch the edge of [the noun] in your hands, and -- exerting considerable upper-body strength -- pull yourself up onto it." instead.

Report Lara entering a platform from a location:

say "Lara jumps, catches the edge of [the noun], and is standing upright on it, all in less time than it takes to tell."

Instead of examining a person who is not the player:

say "[The noun] carries [list of things carried by the noun] and wears [list of things worn by the noun]."

Instead of climbing a platform, try entering the noun.

Test me with "Lara, drop map / lara, drop gun / drop idol / enter bronze platform / drop pickaxe / get off / climb wood".



Overriding the rules to allow the player to show something to another character without first taking it.

As mentioned in this section, the "implicitly taking" activity does not allow us to skip an implicit take entirely. In order to do this, we need to borrow from the chapter on Rulebooks and tell Inform that one of the rules normally built in to the Standard Rules does nothing in certain circumstances:

"Lollipop Guild"

The carrying requirements rule does nothing when showing something to the guardian.

The can't show what you haven't got rule does nothing when showing something to the guardian.

The block showing rule does nothing.

Candyland is a room. "A fizzing, popping wonderland of sugary delights. A path tiled with butterscotch sweets leads to the horizon."

The butterscotch path is scenery in Candyland.

The player carries a basket. In the basket are a licorice gumdrop and a can of tuna. The gumdrop is edible. The description of the gumdrop is "Covered all over with grains of sugar." The can of tuna is edible. The description of the can of tuna is "A rare import in this place."

The giant lollipop is a fixed in place edible thing in Candyland. "Growing right next to the path, on a trunk of white paper, is a giant lollipop colored green and red and white." The description of the lollipop is "If you were very blind, like Aunt Myrtle, you might mistake it for a young sapling just planted: the lollipop is just that leafy shade of green, with swirls of white and red that might be branches or flowers."

The guardian is a man in Candyland. "Right beside you is a guardian in a mint-colored uniform." The description of the guardian is "A killjoy wielding a gigantic toothbrush." The guardian carries a gigantic toothbrush. The description of the toothbrush is "Bristles as long as your hand. Firm bristles, too, not those soft ones. The guardian doesn't care about your tender gums."

A thing can be sweet. The butterscotch path, the lollipop, and the gumdrop are sweet.

Carry out showing a sweet thing to the guardian:

say "The guardian shrieks! You don't understand its language, but from its ululations you understand the idea of decay. There may have been a bit in there about a root canal." instead.

Carry out showing something to the guardian:

say "The guardian nods approvingly at the unsweetened [noun]." instead.

Report eating a sweet thing in the presence of the guardian:

say "The guardian looks mournful, but unholsters his tube of paste and begins applying it to the toothbrush, as though to say that he really did not want to have to do this...";

end the story saying "Everything goes minty" instead.

Report eating something:

say "You consume [the noun] with gusto." instead.

Test me with "x guardian / x toothbrush / show gumdrop to guardian / show path to guardian / show tuna to guardian / look / eat gumdrop".

Note that because we only deactivate the carrying requirements rule for showing purposes, the player still takes the gumdrop before eating it.

87



Example Sand

WI

Extend PUT and INSERT handling to cases where multiple objects are intended at once.

The above example does not quite work when we want the player to be allowed to take multiple objects at once before putting them somewhere: we also need to add a couple of "understand" rules borrowed from many chapters later. While the reasons may not be immediately clear, we include the demonstration here for the sake of thoroughness:

"Sand"

Before inserting something which is not carried by the player into something:

if the noun is in the second noun, say "Already done." instead;

say "(first taking [the noun]) ";

silently try taking the noun;

if the player is not holding the noun, stop the action.

Before putting something which is not carried by the player on something:

if the noun is on the second noun, say "Already done." instead;

say "(first taking [the noun])[line break]";

silently try taking the noun;

if the player is not holding the noun, stop the action.

Understand "put [things] in [something]" as inserting it into. Understand "put [things] on [something]" as putting it on.

The Closet is a room.

A lentil is a kind of thing. A black-eyed pea is a kind of thing. The closet contains 3 lentils. The Closet contains 14 black-eyed peas. The round tin is a container in the closet. The round tin contains 17 lentils. The square tin is a container in the Closet. The square tin contains 20 black-eyed peas.

Sorting is a scene. Sorting begins when play begins. Sorting ends when all the lentils are in the round tin and all the black-eyed peas are in the square tin. When Sorting ends, end the story finally.

When play begins: say "Thanks to your cruel stepmother, you're not going anywhere until the lentils and peas are sorted."

Test me with "put peas in square tin / put lentils in round tin".

100

★ Example Veronica

WI

An effect that occurs only when the player leaves a region entirely.

Suppose that we want to have something happen when the player leaves a region we've defined. "Instead of going from (the region)..." will not suffice for this, because this rule will be invoked every time the player successfully leaves a room within the region, whether or not he is going to a room that is also in the same region.

Instead we need a rule that is a bit more specific, like this:

"Veronica"

Neptune is a region.

Tijuana is a room.

High School is north of Tijuana. It is in Neptune.

Detective Offices is west of High School. It is in Neptune.

The player is in High School.

Instead of going from Neptune to a room which is not in Neptune:
say "It's a bad time to leave Neptune."

Test me with "s / w / e".

105

★ Example Mattress King

WI

Adding extra phrasing to the action to PUSH something in a direction.

By default, when the player pushes something a direction, Inform checks to make sure that the object is pushable between rooms. If not, it blocks the action; if so, it carries out a normal going action with the pushed object taken along.

Also by default, this action produces only a description of the new room that we've traveled into. But suppose we would like to print a short message describing the

pushing action first:

"Mattress King"

Monica's Bedroom is a room. The Living Room is south of Monica's Bedroom.
Rachel's Bedroom is south of the Living Room.

After going a direction (called way-pushed) with something (called the thing-pushed):

say "You push [the thing-pushed] [way-pushed] to [the location].";
continue the action.

The race car bed is an enterable supporter in Monica's Bedroom. It is pushable between rooms.

Test me with "push bed south".

200

★ Example The Second Oldest Problem

WI

Adapting the going action so that something special can happen when going from a dark room to another dark room.

Text in this example is drawn from Will Crowther's original 1976 FORTRAN implementation of ADVENTURE, the founding work of the genre, whose source code was rediscovered by Dennis G. Jerz in 2007. Note the capitals: the program ran on an early computer without lower case lettering. They look a little mimsy now, but picture them glowing green on an old-style cathode ray tube monitor in a darkened room late at night.

The problem alluded to is that the player is forbidden to walk between two dark rooms, so that he must always have light to see by from at least one end of any movement. Writing source text to achieve this is tricky to get right in every case, because the determination of light is hard to do. Here we interleave the necessary rules into the existing "going" action, using a new action variable to record the number of ends which are dark as experienced by the player, which might be 0, 1 or 2:

"THE SECOND OLDEST PROBLEM"

The going action has a number called the dark terminus count.

Setting action variables for going:

now the dark terminus count is 0;
if in darkness, increment the dark terminus count.

The last carry out going rule:

if in darkness, increment the dark terminus count;
if the dark terminus count is 2, end the story saying "YOU FELL INTO A PIT AND BROKE EVERY BONE IN YOUR BODY!" instead.

And now three early rooms to try this out.

COBBLE CRAWL is a room. "YOU ARE CRAWLING OVER COBBLES IN A LOW PASSAGE. THERE IS A DIM LIGHT AT THE EAST END OF THE PASSAGE."

DEBRIS ROOM is west of COBBLE CRAWL. "YOU ARE IN A DEBRIS ROOM, FILLED WITH STUFF WASHED IN FROM THE SURFACE. A LOW WIDE PASSAGE WITH COBBLES BECOMES PLUGGED WITH MUD AND DEBRIS HERE, BUT AN AWKWARD CANYON LEADS UPWARD AND WEST."

AWKWARD CANYON is west of DEBRIS ROOM. "YOU ARE IN AN AWKWARD SLOPING EAST/WEST CANYON."

DEBRIS ROOM and AWKWARD CANYON are dark.

Rule for printing the name of a dark room: say "DARKNESS" instead.

Rule for printing the description of a dark room: say "IT IS NOW PITCH BLACK. IF YOU PROCEED YOU WILL LIKELY FALL INTO A PIT." instead.

Test me with "w / e / w / w".

This is only the second oldest problem in the IF literature: the earliest puzzle is unlocking the steel grate which bars entrance to the cave.

306

★ Example Misadventure

WI

A going by name command which does respect movement rules, and accepts names of rooms as commands.

The original Adventure allowed the player to type the names of rooms in order to move to them, and it is now not too difficult for us to do the same. Adventure restricted this option to adjacent rooms, but we might want to be a bit more flexible, so we will accept any room:

"Misadventure"

Plover Room is a room. "You're in a small chamber lit by an eerie green light. An extremely narrow tunnel exits to the west. A dark corridor leads northeast."

The Dark Corridor is northeast of Plover Room. Plover Room is south of the Dark Corridor. The printed name of the Dark Corridor is "Dark Room". The description of the Dark Corridor is "You're in the dark-room. A corridor leading south is the only exit."

The Alcove is west of Plover Room. "You are in an alcove. A small northwest path seems to widen after a short distance. An extremely tight tunnel leads east. It looks like a very tight squeeze. An eerie light can be seen at the other end."

Northwest of the Alcove is the Misty Cavern. The description of Misty Cavern is "You are following a wide path around the outer edge of a large cavern. Far below, through a heavy white mist, strange splashing noises can be heard. The

mist rises up through a fissure in the ceiling. The path exits to the south and west." West of Misty Cavern is the Alcove.

Understand "[any room]" as going by name. Understand "go to [any room]" as going by name.

Going by name is an action applying to one thing.

We should reject movement to the player's current location, or to anywhere he hasn't been and can't see:

Check going by name:

- if the noun is the location, say "You're already in [the location]." instead;
- if the noun is not adjacent and the noun is unvisited, say "That noun did not make sense in this context." instead.

The assumption here is that the player does know the names of the rooms adjacent to his current location, even if he hasn't been there yet.

Now for the travel itself. The simplest way to ensure that the usual movement rules will still apply is to convert GO BY NAME into a GO action, and here the best route comes to our aid:

Carry out going by name:

- let aim be the best route from the location to the noun, using doors;
- if aim is not a direction, say "You can't think how to get there from here."

instead;

- say "(heading [aim])[command clarification break]";
- try going aim;
- if the location is not the noun, say "You'll have to stop here."

This will allow the player to travel toward rooms he has already visited even if they are several moves away.

Finally, so that the player can also use the names of doors as commands:

Understand "[door]" as entering.

And in keeping with the original, we might add to our scenario a rule or two about restrictions on movement, just to test that it's all working right:

The player carries a plover egg and a platinum pyramid. The description of the egg is "Plover's eggs, by the way, are quite large." The printed name of the egg is "emerald the size of a plover's egg". Understand "emerald" as the egg. The description of the pyramid is "The platinum pyramid is 8 inches on a side!"

Instead of going to the Plover Room from the Alcove when the player carries something which is not the plover egg:

- say "Something you're carrying won't fit through the tunnel with you. You'd best take inventory and drop something."

Test me with "go to misty cavern / go to dark corridor / go to plover room / go to alcove / go to dark corridor / drop pyramid / go to dark corridor / g / go to alcove /

g / go to misty cavern".

372

★ Example Minimal Movement

WI

Supplying a default direction for "go", so that "leave", "go", etc., are always interpreted as "out".

Sometimes it would be nice to respond a little more sensitively to a vague command such as "leave" -- converting it, perhaps, to a "go out" command.

"Minimal Movement"

The Doll-like House is a room. The Postage-Stamp-Sized Garden is outside from the House.

Rule for supplying a missing noun while going:
now noun is outside.

This particular situation is very slightly complicated by the existing rules about vague movement, but fortunately we can easily turn those off.

The block vaguely going rule is not listed in the for supplying a missing noun rules.

Test me with "go".

393

★ Example Saint Eligius

WI

Adding a first look rule that comments on locations when we visit them for the first time, inserting text after objects are listed but before any "every turn" rules might occur.

A not-infrequent desire in IF is to provide a few lines of comment when the player first enters a new room, after the objects are described but before anything else (such as an every turn rule) can fire. The cleanest, most systematic solution is to add a rule to the carry out looking rulebook, so:

"Saint Eligius"

The first look rule is listed after the room description paragraphs about objects rule in the carry out looking rules. A room can be commented or uncommented. A room is usually uncommented.

This is the first look rule:

if the location is uncommented, carry out the gawking at activity with the location.

Gawking at something is an activity.

Rule for gawking at the Diamond Market:

say "Your throat closes and your eyes begin to sting. You have long disdained pomp and luxury, and railed against the oppression that brings such wealth to some men at the cost of the lives of others; you were not prepared for the magnificence."

After gawking at a room (called the target): now the target is commented.

And now the scene itself:

The Cobbled Alley is a room. "The Alley has never been made into a proper street: the buildings on either side are simply too important to tear down. For all that, there isn't much sign of the magnificence nearby. The entrance you seek is set below street level, four grimy steps down to a half-basement."

After going to Diamond Market:

say "You descend the steps quickly and step into the small foyer, allowing yourself to be searched for weapons, before going on to...";
continue the action.

Diamond Market is down from Cobbled Alley. "The roof is vaulted and painted in allegorical images representing Plenty, the Riches of the Earth, and Saint Eligius, patron of goldsmiths and jewelers.

Under their watchful eye, dozens of men in sober black robes sit; and on the tables before them are rubies, emeralds, sapphires from oversea, but most of all diamonds, both raw and cut."

The burly guard is a man in Diamond Market. "A burly guard patrols quite close to you, but even he is more sumptuously dressed than the average burly guard, and his buttons shine."

Test me with "d / look".

6



Example Up and Up

WI

Adding a short message as the player approaches a room, before the room description itself appears.

Sometimes when a player moves from one room to another, we want to imply that a considerable amount of time elapses, or that something interesting occurs on the way. In that case, we might want to print more than just the room description itself. Here is how we might define a couple of rooms that are far apart:

"Up and Up"

The Plain of the Skull is below the Endless Tower. The description of the Plain of the Skull is "A vast and trackless plain, enlivened only by the bones of those who have previously tried and failed to cross. Above you is the Endless Tower, which rises half-way to the moon."

The description of the Endless Tower is "From up here the Plain of the Skull seems only a small bald patch: the world is round and most of it is covered with trees. Far off to the southwest is a shimmering surface that might be water; but there are no signs of cities or civilizations, only the lizard-skeletons."

And now we borrow from the instructions on Actions to create our actual message. "Before..." introduces a rule that occurs when the player tries to do something; in this case, we will make a Before rule for going to the tower.

Before going to the Endless Tower:

say "You climb... and climb... and climb... The sun sets. The moon rises. The wind begins to blow. You continue to climb..."

The player carries a bit of harness. The description of the harness is "A strip of worked leather and a loop of metal, scavenged from one of the skeletons on the plain. Without it, you might think your entire quest was in vain."

Test me with "look / up".

256



Example Wonderland

WI

Hiking Mount Rainier, with attention to which locations are higher and which lower than the present location.

Suppose we have a landscape with a great deal of up and down variation, where GO UP and GO DOWN will be significant almost everywhere, and specifying them all individually a tremendous pain:

"Wonderland"

An altitude is a kind of value. 1000 feet specifies an altitude. A room has an altitude.

Definition: a room is low if its altitude is 3000 feet or less. Definition: a room is high if its altitude is 5000 feet or more.

Instead of going down:

if an adjacent room is lower than the location:

let the valley be the lowest adjacent room;

let the way be the best route from the location to the valley;

say "(that is, [way])[paragraph break]";

try going the way;

otherwise:

say "You're in a local valley: there's no down from here."

Instead of going up:

if an adjacent room is higher than the location:

let the peak be the highest adjacent room;

let the way be the best route from the location to the peak;

say "(that is, [way])[paragraph break]";

try going the way;

otherwise:
say "You're on a local peak."

Paradise is a room. Paradise has altitude 5400 feet. "A handsome parking lot, a picnic ground, and the Henry M. Jackson Memorial Visitor Center. The latter offers, for serious climbers, a hot shower; for nature enthusiasts, an interpretive museum; and for car-trippers, a gift shop selling canned slugs. All of which is a largely unsuccessful distraction from the peak of Mt. Rainier beyond."

Cougar Rock is southwest of Paradise. The altitude of Cougar Rock is 3180 feet. "Numerous individual campsites and (on the road inventively labeled 'F') a handful of larger campgrounds suitable for church groups and family reunions."

Longmire is southwest of Cougar Rock. It has altitude 2760 feet. "A tiny town: it has to offer a few groceries, a post office, and a lodge for people who do not care to camp, all built in a rustic Park Service way."

Panorama Point is north of Paradise. It has altitude 6800 feet. Camp Muir is north of Panorama Point. It has altitude 10188 feet. Columbia Crest is northwest of Camp Muir. It has altitude 14410 feet. St Andrews Rock is west of Columbia Crest. It has altitude 10992 feet. Camp Schuman is northeast of Columbia Crest. It has altitude 9510 feet.

Since Mount Rainier National Park runs to over 235,000 acres, we will omit the rest of the locations, but it does seem fair to give a little more credit to anyone who makes the summit:

Instead of going up in the highest room:
say "You're standing at the summit of Mt. Rainier, the highest point in the state of Washington. There is no up."

Test me with "up / up / up / down / down / up / up".

307



Example Safari Guide

WI

The same functionality, but making the player continue to move until he reaches his destination or a barrier, handling all openable doors on the way.

The foregoing example moves the player one location towards his destination, and requires that rooms have been visited before. But suppose we wanted to be a bit more lenient about movement, and let the player make as many steps as necessary per turn. We will also show consideration about doors, using the "Locksmith" extension supplied with Inform. (Now every time the code attempts opening a door, unlocking rules will also be invoked.)

"Safari Guide"

Include Locksmith by Emily Short.

The Monkey House is a room. The African Grasslands Exhibit is north of the Monkey House. The bird door is north of the African Grasslands Exhibit and south of the Aviary. The Ostrich Enclosure is west of the Aviary. The bird door is a door. It is closed, lockable, and locked. The silver key is a passkey. It unlocks the bird door. The player carries the silver key.

Understand "go to [any room]" as going by name. Understand "[any room]" as going by name. Understand "[door]" as entering.

Going by name is an action applying to one thing.

Check going by name:

if the noun is the location, say "You're already in [the location]." instead.

Carry out going by name:

while the player is not in the noun:

let heading be the best route from the location to the noun, using even locked doors;

if heading is not a direction, say "You can't think how to get there from here." instead;

let destination be the room heading from the location;

say "(heading [heading])[command clarification break]";

try going heading;

if the player is not in the destination, rule fails.

Test me with "go to aviary / go to ostrich enclosure / african grasslands".

Notice that we continue the movement until one of two things happens: either the player reaches the room that is his destination, or the going attempt doesn't work. In the latter case we stop the action in order to avoid hanging the game up in a loop. This event might occur when the player runs into a locked door, for instance.

102



Example Bumping into Walls

WI

Offering the player a list of valid directions if he tries to go in a direction that leads nowhere.

"Bumping into Walls"

First we add an instruction to determine which ways lead to other rooms.

Definition: a direction (called thataway) is viable if the room thataway from the location is a room.

Now we build in the instruction for what Inform should say if the player tries to head in a direction that leads nowhere:

Instead of going nowhere:

let count of exits be the number of viable directions;

if the count of exits is 0, say "You appear to be trapped in here." instead;

if the count of exits is 1, say "From here, the only way out is [list of viable

directions].";
otherwise say "From here, the viable exits are [list of viable directions]."

There is no theoretical reason why we have to define "count of exits" here: we could, if we wanted, just say "if the number of viable directions is 0", "if the number of viable directions is 1", and so on. However, each calculation of a "viable direction" takes a bit of computing power, so there is some slight savings in not requiring the game to count viable directions more than once in this routine.

Dome is a room. North of Dome is North Chapel. South of the Dome is South Chapel. West of the Dome is Western End. Quiet Corner is northwest of the Dome, north of Western End, and west of North Chapel. Loud Corner is east of North Chapel, northeast of Dome, and north of Eastern End. Eastern End is north of Dim Corner and east of Dome. Dim Corner is southeast of Dome and east of South Chapel. Ruined Corner is southwest of Dome, west of South Chapel, and south of Western End.

The Crypt is below the dome.

The church door is east of Eastern End and west of the Courtyard. The church door is a door.

Test me with "u / n / n / e / n / s / u / open door / e / n".

103



Example Polarity

WI

A "go back" command that keeps track of the direction from which the player came, and sends him back.

The main trick of this is always to record where the player has gone when he has just moved.

"Polarity"

The former location is a room that varies.

Here we record where the player has been before moving him; by calling this the "first carry out going rule", we make sure that this rule is followed during the going action before any other pieces of the movement occur. For more detail, see the chapters on advanced actions and on rules.

First carry out going rule:
now the former location is the location.

Understand "go back" as retreating. Understand "back" or "return" or "retreat" as retreating.

Retreating is an action applying to nothing.

Carry out retreating:

let way be the best route from the location to the former location, using doors;
if way is a direction, try going way;
otherwise say "You can't see an open way back."

And to deal with the case where the player has not yet moved:

When play begins: now the former location is the Dome.

Instead of retreating when the former location is the location: say "You haven't gone anywhere yet."

Dome is a room. North of Dome is North Chapel. South of the Dome is South Chapel. West of the Dome is Western End. Quiet Corner is northwest of the Dome, north of Western End, and west of North Chapel. Loud Corner is east of North Chapel, northeast of Dome, and north of Eastern End. Eastern End is north of Dim Corner and east of Dome. Dim Corner is southeast of Dome and east of South Chapel. Ruined Corner is southwest of Dome, west of South Chapel, and south of Western End.

The church door is east of Eastern End and west of the Courtyard. The church door is a door.

Test me with "back / n / go back / e / open door / go through door / go back".

107



Example Provenance Unknown

WI

Allowing something like PUSH TELEVISION EAST to push the cart on which the television rests.

Suppose we have a series of items that might be stacked on top of one another -- say a heavy television on a rolling cart, and we want the player to be able to move the cart with PUSH TELEVISION EAST just as well as with PUSH CART EAST.

This takes a little redirection, using a setting action variables rule. This is not a kind of rule we've encountered yet, and in fact we won't meet it until the Advanced Actions chapter; it is included here for the convenience of authors who want to modify the effect of pushing without reading that far ahead:

"Provenance Unknown"

Setting action variables for pushing something to:

if the noun is enclosed by a pushable between rooms thing (called the pushed item) which is in the location:
now the noun is the pushed item instead.

This rule says that any time we push an object that is on top of a stack of pushable objects, we should transfer the action to the item at the bottom of the stack.

The rest is merely a test case.

The heavy golden idol is on a roller board. The roller board is on a hovercraft.

The hovercraft, the tea trolley, and the skateboard are pushable between rooms.

The hovercraft is in Zeta Proximan Dig Field.

Zeta Proximan Dig Field is a room. "During the day, the field is massed with sweating native workers, overseers, and officials from central command. Now the spades, trowels, brushes, metal detectors, ground probes, plumb lines, and sighting tripods have been laid aside.

All that remains are the trenches and the fine grey dust that blows slowly across them; the moonlight; and the just-emerging outlines of an ancient and alien wall."

The Hover-Road is west of the Dig Field. "A long road hastily laid down, stretching east to west, from the dig site toward the safety of the city."

When play begins:

say "You have, at last, loaded your illicit cargo without setting off any of the many and sensitive alarms set here; now it remains only to sneak out of the area, under the light of Zeta Proxima's lone green moon."

Test me with "push idol west / look / push roller board east / look".

108



Example Zorb

WI

Replacing the message the player receives when attempting to push something that isn't pushable, and also to remove the restriction that objects cannot be pushed up or down.

There are two aspects of Inform's handling of pushable objects that are particularly prime for modification. One is that we may want to change the language used to refuse the pushing of unpushable objects.

Second, Inform by default assumes that it is impossible to push objects in up or down directions. This makes lots of sense if the player is trying to push a wheelbarrow up a ladder; it makes less sense if instead we're pushing a ball up a slope.

We solve both problems with some syntax borrowed from the chapter on rulebooks: in the first case, we replace the old rule with a new one with more friendly phrasing; in the second, we remove the rule entirely. More about how to do this is described in the rulebooks chapter; and in general we can find out what rules contribute to any given action by looking at the Actions index. In this case, the action is "pushing it to", which has its own set of prerequisites (called check rules) that make sure the object can safely be pushed, before turning processing over to the going action.

"Zorb"

Section 1 - Procedure

The new can't push unpushable things rule is listed instead of the can't push unpushable things rule in the check pushing it to rules.

This is the new can't push unpushable things rule:

if the noun is not pushable between rooms:
say "[The noun] [are] not amenable to being pushed from place to place."
instead.

The can't push vertically rule is not listed in any rulebook.

And now to provide a scenario where the player can push something up and down a hillside. Most of the rest of the example is there for local color and to provide a way to demonstrate these rule adjustments:

Section 2 - Scenario

The Steep Hill is a room. The Crest is above Steep Hill. The Valley is below Steep Hill.

The flat rock is a fixed in place thing in the Steep Hill.

The Zorb is a transparent open enterable container in the Steep Hill. "[if the player props the Zorb]The Zorb rests here, kept from further rolling by your support[otherwise]The Zorb is here[end if]". It is pushable between rooms. The description of the Zorb is "A giant plastic inflatable ball, like a hamster ball for humans[if someone is in the Zorb]. Inside [is-are list of people in the Zorb][end if]."

Lucy is a woman in the Zorb.

Carry out going with the Zorb when the Zorb contains Lucy:
say "Lucy whoops delightedly as she rides along in the Zorb."

Every turn when the Zorb is not in the Valley and the player does not prop the Zorb:
let next room be the room down from the location of the Zorb;
if the player is not in the Zorb and the player can see the Zorb:
say "The Zorb succumbs to gravity and rolls down toward [the next room].";
move the Zorb to the next room;
if the player is in the Zorb:
say "The Zorb rolls you down the hill!";
try looking;
otherwise if the player can see the Zorb:
say "The Zorb rolls ponderously but inevitably into the vicinity.";

Propping relates one person to one thing. The verb to prop means the propping relation.

Carry out going with the Zorb:
now the player props the Zorb.

Before doing something when the action requires a touchable noun:
if the noun is not the Zorb, now the player does not prop the Zorb.

Check waving hands when the player is propping something (called casualty):
try the player releasing the casualty.

Carry out entering the Zorb:
now the player does not prop the Zorb.

Understand "let go of [something]" or "let [something] go" or "release [something]" or "free [something]" as releasing. Releasing is an action applying to one thing.

Check releasing:
if the player carries the noun:
try dropping the noun instead.

Check releasing:
if the player does not prop the noun:
say "You are not supporting [the noun]." instead.

Carry out releasing:
now the player does not prop the noun.

Report releasing:
say "You let go of [the noun]."

Test me with "d / push zorb up / look / push zorb up / wave / d / d / push zorb up / release zorb / d / push zorb up / touch rock / push the flat rock south".

179



Example Owen's Law

WI

OUT always means "move to an outdoors room, or else to a room with more exits than this one has"; IN always means the opposite.

Suppose we want the game to interpret "GO OUT" as "move towards an outdoors room, or towards a room with more exits than the current room", while "GO IN" means "move toward a room with fewer exits, or towards an indoors room". Thus going in repeatedly within a building would lead towards dead-ends, while going out repeatedly would lead towards the center of the building and then towards an exit to the outside world.

We start by encoding these rules as definitions:

"Owen's Law"

A room can be indoors or outdoors. A room is usually indoors.

Definition: a room is outward:
if it is not adjacent, no;
if it is indoors and the location is outdoors, no;
if it is outdoors and the location is indoors, yes;
if the number of rooms adjacent to it is greater than the number of rooms

adjacent to the location, yes;
otherwise no.

Definition: a room is inward:
if it is not adjacent, no;
if it is outdoors and the location is indoors, no;
if it is indoors and the location is outdoors, yes;
if the number of rooms adjacent to it is less than the number of rooms
adjacent to the location, yes;
otherwise no.

Instead of going nowhere when the noun is outside: try exiting.

Instead of exiting when the player is in a room:
if at least one room is outward:
let the destination be a random outward room;
let the way be the best route from the location to the destination;
say "(that is, [way])[command clarification break]";
try going the way instead;
otherwise:
say "It's not entirely obvious which way you mean. ";
carry out the listing available exits activity.

Instead of going inside when the room inside from the location is not a room and
at least one room is inward:
if more than one room is inward:
carry out the listing available exits activity;
otherwise:
let the destination be a random inward room;
let the way be the best route from the location to the destination;
say "(that is, [way])[command clarification break]";
try going the way instead.

Instead of going nowhere:
carry out the listing available exits activity.

This "listing available exits" is a refinement borrowed from a future chapter, which
allows us to specify special listing and printing rules:

Listing available exits is an activity.

Rule for listing available exits:
if going inside and an adjacent room is inward:
say "From here 'in' could reasonably mean [a list of adjacent inward
rooms].";
rule succeeds;
if exiting and an adjacent room is outward:
say "From here 'out' could reasonably mean [a list of outward adjacent
rooms].";
rule succeeds;
say "From here you can go [a list of adjacent rooms]."

Before printing the name of a room (called the target) while listing available exits:
let aim be the best route from the location to the target;
say "[aim] to the ".

Rule for printing the name of an unvisited room which is not the location:
say "unknown location".

Dune is an outdoors room. "Hundreds of feet of dune stretch west to the beach, crisscrossed with dune-buggy tracks and the footprints of birds. To the east is a low-lying, boxy concrete installation."

Ocean Shores Military Installation is east of the Dune. It is an outdoors room. "The World War II emplacements, built in case of Japanese invasion, have never been destroyed, though with all the weapons and furnishings gone it is difficult to make much sense of the original structure. A doorway leads west into concrete-lined darkness; a rusty but reliable ladder ascends to a walkway overlooking the sea."

Walkway is above Ocean Shores Military Installation. "From here you have a long view of the dunes and the Pacific Ocean, complete with the rotting hull of a long-stranded vessel."

Dark Echoing Room is inside from Ocean Shores Military Installation. Dank Dripping Room is east of Dark Echoing Room. Narrow Room is south of Dark Echoing Room. Small Sealed Chamber is north of Dark Echoing Room. Room Smelling of Animal Urine is north of Dank Dripping Room. The description of a room is usually "It is dark in here, and feels unsafe."

Test me with "e / u / d / in / s / out / n / out / e / in / out / out / out".

206

★ Example Get Axe

WI

Changing the check rules to try automatically leaving a container before attempting to take it. (And arranging things so that other people will do likewise.)

We could now re-write the check rules so that any time someone (the player or someone else) tries to pick up a container which he is in, he will first get out:

"GET AXE"

This is the clever can't take what you're inside rule:
if the person asked is in the noun, try the person asked exiting;
if the person asked is in the noun, rule fails.

The clever can't take what you're inside rule is listed instead of the can't take what you're inside rule in the check taking rules.

Attic is a room. The unused coffin is in the Attic. The coffin is enterable and openable and open. Raskolnikov is a man in the coffin.

Persuasion rule for asking Raskolnikov to try doing something:
persuasion succeeds.

Test me with "raskolnikov, get coffin".

★ Example Anchorite

By default, Inform understands GET OFF, GET UP, or GET OUT when the player is sitting or standing on an enterable object. We might also want to add GET DOWN and DOWN as exit commands, though.

With GET DOWN, we can replace the whole command, which will not interfere with the normal function of the TAKE verb, or allow the player to attempt to GET any other directions:

"Anchorite"

The Solitary Place is a room. "A glittering, shimmering desert without either locusts or honey." The pillar is an enterable supporter in the Solitary Place. "The broken pillar is short enough to climb and sit on." The description of the pillar is "Once it was a monument: a long frieze of battles and lion-hunts spirals up the side, in honor of an earthly king." The player is on the pillar.

Understand "get down" as exiting.

This doesn't cover the case where the player just types "DOWN", and we don't want to preempt the normal operation of the GO action here. So instead of writing a new understand instruction, we might catch this one at the action-processing level:

Instead of going down when the player is on a supporter:
try exiting.

Test me with "down / enter pillar / get down / down / get down".

★ Example Lies

Commands to allow the player to lie down in three different ways.

To set the scene, and make new actions to provide for two of these ways:

"Lies"

The Laundry is a room. "An old Limehouse haunt, the Chinese laundry used by the down-trodden wives of the Tong of the Black Scorpion." The vast marble sink is here. "There is nothing obviously oriental about the vast marble sink, which is large enough to lie down inside. A wooden-rack floor, equipped for easy drainage, turns out also to be equipped for snagging the shoes of passers-by." The sink is an enterable container, fixed in place.

Lying down is an action applying to nothing. Report lying down: say "You lie down for a while in the middle of the Laundry, wondering about the point of existence, then get up again."

Lying near is an action applying to one thing. Report lying near: say "You lie down next to [the noun] for a while, mumbling to yourself."

Instead of lying near the sink, say "Lying down close to the cool butcher's marble slabs of the sink, your attention is caught by the sight of coolie shoes through a floor-level grille for ventilation. The game is afoot!"

So far, so good. Now for the grammar, where we create two new tokens: one for each of two groups of alternative prepositions.

Understand "beneath/under/by/near/beside/alongside/against" or "next to" or "in front of" as "[beside]".

Understand "on/in/inside" or "on top of" as "[within]".

Understand "lie down" as lying down.

Understand "lie down [within] [something]" as entering.

Understand "lie [beside] [something]" or "lie down [beside] [something]" as lying near.

Test me with "lie down / lie down on top of the sink / get out / lie down inside the sink / get out / lie down in front of the sink".

388

★ Example Nine AM Appointment

WI

A WAIT [number] MINUTES command which advances through an arbitrary number of turns.

If there's some reason the player needs to be at a specific place and time, we might want to allow him to wait a number of minutes at once.

"Nine AM Appointment"

Waiting more is an action applying to one number.

Understand "wait [a time period]" or "wait for [a time period]" or "wait for a/an [a time period]" or "wait a/an [a time period]" as waiting more.

Carry out waiting more:

- let the target time be the time of day plus the time understood;
- decrease the target time by one minute;
- while the time of day is not the target time:
 - follow the turn sequence rules.

The one nuance here is that after our wait command occurs, the turn sequence rules will occur one more time. So we need to subtract one minute from the parsed time to make the turn end on the desired number of minutes.

Report waiting more:
say "It is now [time of day + 1 minute]."

And if we want to ensure that the player doesn't (accidentally or intentionally) put the interpreter through a really long loop, we could put an upper limit on his patience:

Check waiting more:
if the time understood is greater than one hour, say "You really haven't got that kind of patience." instead.

The Specialist's Office is a room. The secretary is a woman in the Office. Instead of asking the secretary about "[appointment]", say "'Hang on just five more minutes,' she says, in a distracted manner."

Understand "appointment" or "specialist" or "doctor" as "[appointment]".

At 9:45 AM: say "At [the time of day in words], secretary glances at you and gives a reassuring smile."

Test me with "ask secretary about appointment / wait five minutes / g / g / wait 61 minutes / wait for half an hour / wait for a quarter of an hour / wait for an hour".

47



Example Change of Basis

WI

Implementing sleeping and wakeful states.

Suppose we want to allow the player to go to sleep some of the time:

"Change of Basis"

A person is either awake or asleep. A person is usually awake.

The important thing to note here is that it does not work to say "the player is either asleep or awake". This is because the player is not necessarily one specific person or thing during the game: the identity of the player can be changed, as we will see later.

So if we want to make rules about the properties of the player, we should attach these rules to the "person" kind.

Linear Algebra Class is a room. "The blackboard is covered with square arrangements of numbers. These are supposed to convey something to you, but mostly you're finding them soporific."

Now a few rules about changing from one state to the other:

Instead of sleeping: now the player is asleep; say "You drop off."

Instead of doing something other than waking up, waiting or sleeping when the player is asleep:
say "Ssh! You're sleeping!"

Instead of sleeping when the player is asleep:
say "Zzzz."

Instead of waking up when the player is asleep:
now the player is awake;
say "You come to suddenly, wiping drool from your lips."

Instead of doing something other than looking or sleeping when the player is awake:
say "You'd really rather just sleep through this."

Test me with "wake up / sleep / look / z / sleep / wake up / look".

389

☆☆ Example Delayed Gratification

WI

A WAIT UNTIL [time] command which advances until the game clock reaches the correct hour.

"Delayed Gratification"

Hanging around until is an action applying to one time.

Check hanging around until:
if the time of day is the time understood, say "It is [time understood] now!"
instead;
if the time of day is after the time understood, say "It is too late for that now."
instead.

Carry out hanging around until:
while the time of day is before the time understood:
follow the turn sequence rules.

Report hanging around until:
say "You yawn until [time understood]."

Understand "wait until [time]" as hanging around until.

The Empty Field is a room. "It's an ordinary empty field. Nothing to see here at all-- yet. Wait until 11:45 PM, though."

At 11:45 PM:
say "Suddenly the air is filled with light and the sounds of an approaching band. Over the crest of the hill comes a parade of singing, stomping, hooting people: and not just people, but dogs, horses, elephants, giraffes... There are banners, and candles, and a flag that glows eerie-green in the dark; there is a float shaped like an enormous turtle, its shell covered with winking green lights; there is an old man dressed as a skeleton, carried in a litter, his neck garlanded with dried chiles. There are small girls throwing rose petals from a basket, and grown women half-naked carrying the emblems of Bacchic revelry, and two little boys each with a silver basin of clear water. All these go by in procession, and you join on at the end.";
end the story finally.

Test me with "look / z / z / wait until 11:45 PM".

287

Example XYZZY

WI

Basics of adding a new command reviewed, for the case of the simple magic word XYZZY.

We have seen before how to define a new action from scratch, but we may want to review here, using a simple command that requires no objects.

"XYZZY"

Understand "xyzzy" or "say xyzzy" or "cast xyzzy" as casting xyzzy.

Casting xyzzy is an action applying to nothing.

Check casting xyzzy:

if the player does not wear the amulet of elocution, say "You are unable to articulate the second 'z' separately from the first, and the spell fails in a disdainful puff. Must be Parisian magic." instead;

if the player has the plate, say "The plate of cheeses twitches uncomfortably, aware that it should be doing something, but not sure what." instead.

Carry out casting xyzzy:

move the plate to the player.

Report casting xyzzy:

say "Under the influence of the Amulet of Elocution, you pronounce this as Xhi-zee. And lo, from nowhere, a [plate] appears!"

The amulet of elocution is a wearable thing. It is carried by the player. The description is "A heavy gold ring on a chain. If heated in an ordinary house fire, it glows with the words, 'Moses Supposes His Toeses Are Roses.'"

The plate is a portable supporter. On the plate is a very ripe ooze. Instead of smelling the ooze, say "It smells like socks. This is going to be wonderful." The ooze is edible. The printed name of the plate is "plate[if the plate supports the ooze] of cheese[end if]". The description of the ooze is "Definitely genuinely cheese." Understand "cheese" as the ooze.

Instead of eating the ooze: now the ooze is nowhere; say "You are transported..."; move the player to Paradise.

The Cheez Factory is a room. "All around you are squares of pressed orange polymer, or possibly cheez. Your only hope is the magic word your uncle taught you: XYZZY." The squares of pressed orange polymer are scenery in the Factory. The description is "You see nothing special about the squares of pressed orange polymer. Nothing special at all." Understand "square" or "cheez" as the squares.

Paradise is a room. The description is "Well, it might just be one of the posh upper rings of purgatory, if you're entirely honest with yourself."

Test me with "x squares / x amulet / x cheese / xyzy / wear amulet / xyzy / x ooze / smell ooze / eat ooze".

XYZZY is a magic word from the original Adventure, and many other games respond to it with some sort of amusing message.

88

Example Fine Laid

WI

Making writing that can be separately examined from the paper on which it appears, but which directs all other actions to the paper.

Sometimes it is useful to direct all -- or almost all -- actions from one object to another. For the sake of argument, say we have a sheet of paper with writing on it, and (because we're very meticulous) we want to let the player examine the writing and get a customized response, different from when he just examines the sheet of paper. But for all other purposes -- say, TAKE or TASTE -- we want the two objects to be treated as one.

Here, we approach the problem by changing the noun and/or the second noun of the current action, then issuing a new command to "try the current action". Because we've changed the noun and second noun, the "current action" at this point is different from the one generated originally by the player's command.

"Fine Laid"

High Street Stationer is a room.

The sheet of paper is a thing in High Street Stationer. The writing is part of the sheet of paper.

The description of the sheet of paper is "A beautiful sheet of heavy cream paper." The description of the writing is "Delicate and spidery."

Instead of tasting the sheet of paper, say "You might need more fiber in your diet, but this isn't the way."

Before doing something other than examining when the current action involves the writing:

- if the writing is the noun, now the noun is the sheet of paper;
- if the writing is the second noun, now the second noun is the sheet of paper;
- try the current action instead.

Test me with "examine sheet of paper / examine writing / get writing / taste writing".

★ Example Lucy

Redirecting a question about one topic to ask about another.

Occasionally we will want to replace the player's question topic with another of our own devising. We can do this in the simplest possible case like so:

"Lucy"

The International Boardgame Championship is a room. Lucy is a woman in the Championship.

Instead of asking Lucy about "checkers":
try asking Lucy about "games".

Instead of asking Lucy about "games",
say "'I don't like games,' she sniffs."

Test me with "ask lucy about checkers / ask lucy about games".

Note that this syntax did not work in older versions of Inform; it is now safe.

★ Example Cactus Will Outlive Us All

For every character besides the player, there is an action that will cause that character to wither right up and die.

"Cactus Will Outlive Us All"

Death Valley is a room. Luckless Luke and Dead-Eye Pete are men in the Valley. A cactus is in the Valley. Persuasion rule: persuasion succeeds.

A person has an action called death knell. The death knell of Luckless Luke is pulling the cactus. The death knell of Dead-Eye Pete is Luke trying dropping the cactus.

Before an actor doing something:

repeat with the victim running through people in the location:
let the DK be the death knell of the victim;
if the DK is not waiting and the current action is the DK:
say "It looks as if [the DK] was the death knell for [the victim], who looks startled, then nonexistent.";
now the victim is nowhere.

If we leave it at that, then pulling the cactus will kill Luckless Luke but then say "Nothing obvious happens.", which seems like a bit of an anti-climax. So we add a special case response for that one:

After pulling the cactus when Luckless Luke was in the location:
say "That's a real shame."

Test me with "get cactus / drop cactus / luke, get cactus / luke, drop cactus / pull cactus / look".

430

Example Leopard-skin

WI

A maze that the player can escape if he performs an exact sequence of actions.

Suppose (as in Infocom's *Leather Goddesses of Phobos*) that we have a maze that the player can escape only by performing the correct sequence of actions in the correct order. One way to do this would be to keep a list of the player's most recent actions, and see whether these match up with the combination we have established as the maze's solution.

For instance:

"Leopard-skin"

The Fur-Lined Maze is a room. "This seemingly endless sequence of rooms is decorated in a tasteful selection of exotic furs and gilded fixtures."

Clapping is an action applying to nothing. Understand "clap" as clapping.
Kweepaing is an action applying to nothing. Understand "kweepa" as kweepaing.

Carry out clapping:
say "You clap."

Carry out kweepaing:
say "You holler 'KWEIPA!' triumphantly."

The maze-sequence is a list of stored actions that varies.

When play begins:
add jumping to the maze-sequence;
add clapping to the maze-sequence;
add kweepaing to the maze-sequence.

The attempted-sequence is a list of stored actions that varies.

Every turn when the player is in the Fur-Lined Maze:
truncate the attempted-sequence to the last two entries;
add the current action to the attempted-sequence;
if the attempted-sequence is the maze-sequence:
say "That does it! You are instantly transported from the maze!";
end the story finally.

Test me with "hop / clap / clap / hop / kweepa / hop / clap / kweepa".

Example I Didn't Come All The Way From Great Portland Street

In this fiendishly difficult puzzle, which may perhaps owe some inspiration to a certain BBC Radio panel game (1967-), a list is used as a set of actions to help enforce the rule that the player must keep going for ten turns without hesitation, repetition, or deviating from the subject on the card.

There is very little to this, in fact. The tricky rule to enforce is Repetition: the player is forbidden to repeat any previously tried action. We keep track of this by keeping a set of past actions, which for want of a better term is called the "tally". All we need to do is:

if the current action is listed in the tally, challenge for "Repetition of [the current action]!";
otherwise add the current action to the tally.

Note that the tally can never contain duplicates, and that when, at the end of the round, we print it out, we sort it first - this makes a more natural-looking sentence. (Sorting a list of actions uses the natural order for actions: compare the sequence on the Actions page of the Index.) The full text, then, is:

"I Didn't Come All The Way From Great Portland Street"

The Paris Theatre is a room. An instrument is a kind of thing. The violin, the tuba, the xylophone and the triangle are instruments. The violin is inside the case. The tuba, the xylophone, the radish, the case, the bust of Nicholas Parsons, the purple felt hat and the triangle are in the Paris Theatre.

The Round is a scene. The Round begins when play begins. The Round ends when the turn count is 10.

The tally is a list of stored actions that varies.

When the Round begins:

say "'And the subject on the card is... musical instruments. Will you carry out for us something to do with that, please, for ten turns starting - now!'"

When the Round ends:

sort the tally;
say "Phweeep![paragraph break]'So, when the whistle goes ten turns are up, you get a point for acting when the whistle blows, and in that round you entertained us by [the tally], and you also get a bonus point for keeping going until the whistle went.'";
end the story finally.

To challenge for (infracation - text):

say "Bzzzzt! 'And [one of]Clement Freud[or]Derek Nimmo[or]Kenneth Williams[or]Peter Jones[at random] has challenged.'[paragraph break]'[infracation]'[paragraph break]'Well, as it's your first time playing the game, and the audience was enjoying your contribution so much, I will disallow the

challenge, you have [10 minus the turn count] turn[s] left on musical instruments, starting... now!"

Before doing something:

if the current action is listed in the tally, challenge for "Repetition of [the current action]!" instead;

otherwise add the current action to the tally;

if waiting, challenge for "Hesitation!" instead;

if not looking and not waiting and the noun is not an instrument and the second noun is not an instrument, challenge for "Deviation!" instead.

Test me with "look / wait / examine bust / take tuba / get triangle / hit xylophone / get tuba / examine tuba / get violin".

(The Paris Theatre in Lower Regent Street, London, was for many years the home of BBC radio panel games.)

222



Example Anteaters

WI

The player carries a gizmo that is able to record actions performed by the player, then force him to repeat them when the gizmo is dropped. This includes storing actions that apply to topics, as in "look up anteater colonies in the guide".

"Anteaters"

A book is a kind of thing. Understand "book" as a book. A book has a table name called the contents.

Report consulting a book about:

say "You flip through [the noun], but find no reference to [the topic understood]." instead.

Instead of consulting a book about a topic listed in the contents of the noun:

say "[reply entry][paragraph break]".

The Guide to Desert Fauna is a book. The contents of the Guide is the Table of Critters.

Table of Critters

topic reply

"spines" "You flip through the Guide for a while and eventually realise that spines are flora, not fauna."

"anteater" "The giant anteater, which grows to six feet in size and can kill a jaguar, is a solitary animal, found in colonies" many habitats, including grasslands, deciduous forests and rainforests. It does not form colonies. That's ants. They're actually quite easy to tell apart."

Death Valley is a room. The Guide is in the Valley.

The gizmo is in Death Valley. The gizmo has an action called idea. The description of the gizmo is "The gizmo is hard to describe, but it projects an idea of [idea]."

Before when the player carries the gizmo and the idea of the gizmo is waiting:
say "[The gizmo] eagerly soaks up the whole idea of [the current action].";
now the idea of the gizmo is the current action.

After dropping the gizmo:
say "The percussion of the fall seems to have shaken the gizmo's idea loose!
There's nothing for it now but [idea of the gizmo].";
try the idea of the gizmo;
now the idea of the gizmo is waiting.

Test me with "get guide / look up spines in guide / x gizmo / get gizmo / i / x
gizmo / drop gizmo / get gizmo / look up anteater colonies in guide / x gizmo /
drop gizmo".

294

Example Shawn's Bad Day

WI

Allowing the player to EXAMINE ALL.

We can add the handling of multiple objects to an existing action simply by adding in a line of grammar using "[things]". In response, Inform will consider every object accepted by the token, and perform the action once for each of those objects. Thus:

"Shawn's Bad Day"

The Treasury is a room. The vault is a lockable locked closed openable container in the Treasury. It is fixed in place. "A massive vault fills up one wall." The description is "The vault's system includes [a list of things which are part of the vault]."

A little green light, a little blue light, a little red light, a thin black pane of glass, a laser beam, a retinal scanner, a thumbprint ID plate, a dial, and a large lever are part of the vault.

The security guard is a man in the Treasury. The description is "His name is Shawn, and he doesn't look happy."

The description of the green light is "Off." The description of the blue light is "Tranquilly on." The description of the red light is "Angrily flashing."

Understand "examine [things]" as examining.

Test me with "examine all".

327

Example The Best Till Last

WI

Reordering multiple objects for dramatic effect.

If a single command asks to do many things, some dull and some exciting, we may want to save the good ones for the end.

"The Best Till Last"

The Funky Ignition Lounge is a room. "This is where all evenings end." The stick of gelignite, the solid magnesium footstool, the vetiver candle, and the vodka bottle are here.

The burn description of the vetiver candle is "It burns right down, expensively but gothically."

The player carries an inexpensive firework. The description of the firework is "It is a cardboard tube with red and green stripes along the outside, and a fuse sticking out of the end." The burn description of the firework is "It ignites gloriously! You take a few hasty steps back in order to avoid burning yourself, and not a moment too soon. Red and green sparks fly out of the tube, and there's a whistling noise punctuated by several loud cracks."

The player carries a lighter. The description of the lighter is "You don't smoke, but you like to have access to flame now and then anyway."

Burning it with is an action applying to one thing and one carried thing.

Understand "burn [things] with [something preferably held]" as burning it with.

The block burning rule is not listed in any rulebook.

A thing has some text called the burn description.

Check burning something:
if the player carries the lighter:
 try burning the noun with the lighter;
else:
 try burning the noun with the noun.

Check burning something with something when the second noun is not the lighter:
 say "Your trusty lighter is the best flame source available to you." instead.

Check burning something with something:
if the burn description of the noun is "":
 say "Best not." instead.

Carry out burning something with something:
 remove the noun from play.

Report burning something with something:
 say "[burn description of the noun][line break]".

A multiple action processing rule when the action name part of the current action is the burning it with action (this is the orderly burn rule):
 let L be the multiple object list;
 let dull list be a list of objects;
 let fun list be a list of objects;
 repeat with item running through L:
 if the burn description of the item is "":

```
        add item to dull list;
    else:
        add item to fun list;
    let F be the dull list;
    add fun list to F;
    alter the multiple object list to F.
```

Test me with "burn all with lighter".

328



Example Western Art History 305

WI

Allowing EXAMINE to see multiple objects with a single command.

In a gallery, there are many individual things to look at, but you can also get a general impression by just examining them as a collection.

First, we'll make a kind for the paintings exhibited in the gallery, and then we'll also make a special object to represent all of them as a mass:

```
"Western Art History 305"
```

```
A painting is a kind of thing. A painting is usually fixed in place. Understand
"painting" as a painting. Understand "paintings" as the plural of painting.
```

```
The painting-collective is a thing. The printed name of the painting-collective is
"paintings". The description of the painting-collective is "There's [a list of visible
paintings]."
```

We could if we wanted tweak the description to be different in style in different rooms of the gallery, but this will do for now. Next we need to make it possible to type something like EXAMINE PAINTINGS, which normally wouldn't work because the Standard Rules don't tell Inform to recognise multiple objects with the EXAMINE command (unlike, say, DROP or TAKE). This is easy:

```
Understand "examine [things]" as examining.
```

Now to make use of the special object. If the player types EXAMINE PAINTINGS, the multiple object list will become a list of the visible paintings. The following rule looks at this list: if it contains more than one painting, it replaces them with the painting-collective instead. Now there's only one examining action, so we get a reply like "There's an abstract painting, a pointilist painting and a French academic painting." instead of a list of descriptions of each in turn.

```
A multiple action processing rule when the current action is examining (this is the
examine kinds rule):
```

```
    let L be the multiple object list;
    let F be L;
    let the painting count be 0;
    repeat with item running through L:
        if the item is a painting:
            increment the painting count;
```

remove the item from F;
if the painting count is greater than one:
add the painting-collective to F;
alter the multiple object list to F.

And now some art to try this out on:

Gallery is a room. "Various paintings hang on the walls of this gallery, awaiting critical attention. A side chamber to the north contains smaller works."

The abstract painting, the pointilist painting, and the French academic painting are paintings in the Gallery.

North of the Gallery is the Side Chamber. A handsome miniature is a painting in the Side Chamber. The description of the handsome miniature is "The miniature depicts a uniformed soldier of the late 18th century, with braid on his shoulders and a curl in his beard."

The player carries a small notebook. The description of the notebook is "It contains the notes you've taken so far towards a paper for Western Art History 305. So far you're still feeling a bit uninspired."

Test me with "x paintings / x all / n / x paintings / x all".

411



Example Escape from the Seraglio

WI

Replacing the usual response to TAKE ALL so that instead of output such as "grapes: Taken. orange: Taken.", Inform produces variable responses in place of "grapes:".

"Escape from the Seraglio"

Section 1 - Special Announcement Rules

The number of takes this turn is a number that varies. Every turn: now the number of takes this turn is 0.

The friskily announce items from multiple object lists rule is listed instead of the announce items from multiple object lists rule in the action-processing rules.

This is the friskily announce items from multiple object lists rule:

```
if taking:
  if the current item from the multiple object list is not nothing:
    increment the number of takes this turn;
    say "[if number of takes this turn is 1]First [otherwise if the number of
takes this turn is 2]And then [otherwise if the number of takes this turn is 3]And I
suppose also [otherwise if the number of takes this turn is 7]And on we wearily
go with [otherwise if the number of takes this turn is 9]Oh, and not forgetting
[otherwise]And [end if][the current item from the multiple object list]: [run
paragraph on]";
  otherwise:
```

if the current item from the multiple object list is not nothing, say "[current item from the multiple object list]: [run paragraph on]".

Rule for deciding whether all includes the person asked: it does not.

Rule for deciding whether all includes a person when taking: it does not.

Section 2 - The Scenario

The Palm Chamber is a room. Sarissa is a woman in the Palm Chamber.

The Palm Chamber contains a bottle of ink, a quill pen, a tangerine, a bunch of grapes, a length of silken rope, some perfume, a cake of incense, a fitted leather bodice, a sapphire anklet, an illustrated novel, a whip, and a heavy iron key.

A persuasion rule for asking Sarissa to try taking the key:

say "Sarissa nervously demurs, knowing that it is forbidden.";
persuasion fails.

A persuasion rule: persuasion succeeds.

Test me with "take all / drop all / sarissa, take all".

428



Example Formicidae

WI

Manipulating the order in which items are handled after TAKE ALL.

Suppose we have an item that produces an interesting result the first time the player lifts it -- a rock with dangerous ants revealed underneath. The effect of the surprise is a little weakened, though, if the player sees that response as the result of a TAKE ALL, when it might be printed like this:

```
>[3] get all  
tent peg: Taken.  
water flask: Taken.  
trading permit: Taken.  
innocent-looking rock: You reach for the rock and turn it over to reveal a thriving  
colony of flesh-eating ants. Needless to say, you drop the rock and jump back  
with a decidedly effeminate scream. They can probably hear you all the way  
back in the base camp.  
rusty nail: Taken.
```

[Your score has just gone down by two points.]

The calm response to "rusty nail" looks odd now, and the score change is disconnected from the event that caused it.

To manage this, we might institute a system so that interesting objects are handled last in their list, like so:

"Formicidae"

Use scoring.

Section 1 - Procedure

The magic rule is listed before the generate action rule in the turn sequence rules.

A thing has a number called dramatic potential.

This is the magic rule:

```
let L be the multiple object list;  
if the number of entries in L is greater than 1:  
    sort L in dramatic potential order;  
    alter the multiple object list to L.
```

Section 2 - Scenario

The Foothills is a room. "The land has become hilly; though the soil is still mostly coarse yellow sand, clumps of grass are able to grow in the shadier places. Deep wagon ruts running from the southwest towards the mountains in the northeast show where generations of caravans have already passed."

The water flask, the tent peg, and the trading permit are things in Foothills.

The rock is a thing in Foothills. Before printing the name of the rock when the rock is not handled: say "innocent-looking ". The dramatic potential of the rock is 10.

The rusty nail is a thing in Foothills.

The ant colony is a fixed in place thing. "A busy group of ants are crawling to and fro in the unaccustomed sun." Rule for deciding whether all includes the ant colony while taking: it does not.

Instead of taking the rock when the rock is handled:
say "It might still have a stray ant or two on it."

After taking the rock:

```
now the rock is handled;  
move ant colony to the location;  
move the rock to the location;  
say "You reach for the rock and turn it over to reveal a thriving colony of flesh-eating ants. Needless to say, you drop the rock and jump back with a decidedly effeminate scream. They can probably hear you all the way back in the base camp.";  
decrease score by 2.
```

Test me with "get peg / drop peg / get all / get rock".

Note that while one could also manipulate the object list to add or remove items at this stage, there's a simpler way to control what Inform considers "ALL" to mean in commands: see the activity "Deciding whether all includes" in the activities chapter.



Example The Facts Were These

Creating a variant GIVE action that lets the player give multiple objects simultaneously with commands like GIVE ALL TO ATTENDANT or GIVE THREE DOLLARS TO ATTENDANT or GIVE PIE AND HAT TO ATTENDANT. The attendant accepts the gifts only if their total combined value matches some minimum amount.

Occasionally it happens that we want to process an action on multiple items differently than we would if the player had just typed each of the individual actions separately. In this example, the reason is that we can only successfully GIVE items when their combined value passes a certain threshold amount; otherwise the recipient will reject them.

This works as an implementation of money, if we give value only to cash objects (though several other implementations of cash are available, most of which are simpler and more efficient). We could also imagine a mechanic like this being used for a bargaining or auction game as well, given a society that deals in objects rather than credits.

In order to consider all the items in the gift at once, we create an action that applies to multiple objects, but will in fact test the whole object collection during the first pass and print a definitive answer to whether the action succeeded. All subsequent times the game consults the rulebook will be stopped at the very beginning. No further processing will occur or output be printed.

"The Facts Were These"

Section 1 - Procedure

We start by creating the idea that everything in the game has a monetary value:

A price is a kind of value. \$10 specifies a price. A thing has a price.

Understand "give [things preferably held] to [someone]" as multiply-giving it to.
Understand "give [things] to [someone]" as multiply-giving it to. Multiply-giving it to is an action applying to two things.

A subtlety here: we say "things preferably held" to prefer items that the player is holding (so if the player has two dollars in hand and a third lies on the ground, he will use just the two he has).

The second grammar line allows Inform to match things that aren't held if it can't make up the list from things that are. If all three dollars are on the ground, the player can pick them up before spending them.

We do not, however, make multiply-giving apply to a "carried" item, because that will generate implicit takes of those items in a way that will mess up our action reporting. Instead, we're going to build the implicit takes into the system in a different way, one that permits us to collate the reports more attractively and print a short, one-sentence list of anything that the player had to pick up.

A thing can be given or ungiven. A thing is usually ungiven.

This is for record-keeping purposes so that we can print an attractive list of what was given at the end of the turn.

First check multiply-giving it to:
if already gave at the office is true:
stop the action.

Already gave at the office is a truth state that varies.

"Already gave at the office" is the perhaps-excessively-named flag that keeps track of whether we've already done this action once.

Check multiply-giving something to the player:
now already gave at the office is true;
say "You can hardly bribe yourself.[paragraph break]" instead;

The following rule is longish because it processes the entire list at once, generating implicit takes if necessary (but processing those implicit takes silently according to its own special rule, so that the output can be managed attractively). We are also, at the same time, calculating the total value of the player's offer.

Check multiply-giving it to:
let L be the multiple object list;
let bribe-price be \$0;
repeat with item running through L:
if the player does not carry the item:
abide by the ungivability rules for the item;
carry out the implicitly taking activity with the item;
if the player does not carry the item:
now already gave at the office is true;
say "You can't include [the item] in your bribe, since you're not holding [them]![paragraph break]" instead;
increase bribe-price by the price of item;
if the number of entries in the recently-collected list is greater than 0:
repeat with item running through the recently-collected list:
now item is marked for listing;
say "You pick up [the list of marked for listing things] and make your offer. [run paragraph on]";
now everything is unmarked for listing;
if the bribe-price is less than the price of the second noun:
now already gave at the office is true;
say "[The second noun] angrily rejects your piffling bribe.[paragraph break]" instead.

The bit about making some items "marked for listing", above, rather than printing the list directly, is that using the "[the list of....]" syntax guarantees that Inform will respect grouping rules in writing its description. For instance, if the player has automatically taken all three dollars, the output will say "the three dollars" instead of "the dollar, the dollar, and the dollar."

Carry out multiply-giving it to:
let L be the multiple object list;

repeat with item running through L:
now the second noun carries the item;
now the item is given;
now already gave at the office is true;

Report multiply-giving it to:
say "[The second noun] rather shamefacedly tucks [the list of given things]
away into a pocket.[paragraph break]".

Now we create our own variation of implicitly taking in order to customize the output for the multiply-giving action. The "ungivability rules" should disallow any object that the player absolutely cannot take, because we want "carry out the implicitly taking activity" to succeed every time -- and therefore not print out any less-attractive results from implicit takes that don't succeed. Otherwise, the player's GIVE TREE AND DOG TO ATTENDANT might produce the reply "That's fixed in place" -- without specifying which object is fixed in place.

Because of the way this works, we will want to be careful: if we have any "instead of taking..." rules for special objects in the game, we should be sure to mirror those with an ungivability rule to print something more suitable in the case that the player tries taking that object as part of the multiple giving action.

The ungivability rules are an object-based rulebook.

An ungivability rule for a person:
now already gave at the office is true;
say "Slavery is illegal.[paragraph break]" instead.

An ungivability rule for something (called the item) which is enclosed by someone who is not the player:
now already gave at the office is true;
say "[The item] [aren't] yours to give.[paragraph break]" instead.

An ungivability rule for something which encloses the player:
now already gave at the office is true;
say "You don't want to end up as part of the gift.[paragraph break]" instead;

An ungivability rule for something (called the item) which is part of something:
now already gave at the office is true;
say "[The item] [are] attached to [a random thing which incorporates the item]
[paragraph break]" instead.

An ungivability rule for something (called the item) which is scenery:
now already gave at the office is true;
say "[The item] [are] unremovable.[paragraph break]" instead.

An ungivability rule for something (called the item) which is fixed in place:
now already gave at the office is true;
say "[The item] [are] fixed in place.[paragraph break]" instead.

An ungivability rule for a direction (called the item):
now already gave at the office is true;
say "[The item] [are] not susceptible to giving.[paragraph break]" instead.

Rule for implicitly taking something (called target) while multiply-giving:
silently try taking the target;
if the player carries the target:
add the target to the recently-collected list.

The recently-collected list is a list of objects that varies.

And since we don't want to list the individual objects separately:

The selectively announce items from multiple object lists rule is listed instead of the announce items from multiple object lists rule in the action-processing rules.

This is the selectively announce items from multiple object lists rule:
if multiply-giving:
do nothing;
otherwise:
if the current item from the multiple object list is not nothing:
say "[current item from the multiple object list]: [run paragraph on]".

And now, since this ought to work symmetrically if the player provides just one high-value item:

Check giving something to someone:
if the price of the noun is less than the price of the second noun:
say "[The second noun] angrily rejects your piffling bribe." instead.

As we've seen elsewhere, the giving action by default returns a refusal, but is also written to start working if we remove the blockage. So we do that here, and revise the report rule to match the report rule we have for multiple giving.

The block giving rule is not listed in any rulebook.

The new report giving rule is listed instead of the standard report giving rule in the report giving it to rules.

This is the new report giving rule:
say "[The second noun] rather shamefacedly tucks [the noun] away into a pocket."

After each instance of the multiply-giving action, we need to clear the variables we used to track its state. We could do this in "Before reading a command", but that's unsafe because the player might type GIVE PIE AND CAP TO ATTENDANT. GIVE DOLLARS TO ATTENDANT. all on a single line, and we would like to be able to clear the variables between one action and the next. The correct place to attach this behavior is immediately before the generate action rule, thus:

The before-generation rule is listed before the generate action rule in the turn sequence rules.

This is the before-generation rule:
now every thing is ungiven;
now already gave at the office is false;
truncate the recently-collected list to 0 entries.

Section 2 - Scenario

The Morgue Office is a room. "This is not the Morgue itself; this is only its outer office. The familiar room full of silver drawers and cold air lies beyond."

The Morgue Attendant is a man in the Morgue Office. "The Attendant has seen you come through a number of times, and is becoming suspicious of your abiding interest in dead people." The description is "The Morgue Attendant is fifty-four years, six months, five days, and three minutes old." The price of the Morgue Attendant is \$3.

A dollar is a kind of thing. The player carries three dollars. The price of a dollar is always \$1.

The player carries a miniature rhubarb pie. The price of the miniature rhubarb pie is \$5.

The player carries a knitted cap. The price of the knitted cap is \$2.

Test me with "test dollars / purloin three dollars / test multi-line / purloin three dollars / purloin pie / purloin cap / test specificity / purloin three dollars / test largesse / test mixed-gift".

Test multi-line with "give dollar and pie to attendant. give dollars and cap to attendant".

Test dollars with "drop all / give dollar to Morgue Attendant / give dollars to Morgue Attendant / get dollars / give dollars to morgue attendant / purloin three dollars / drop dollars / give dollars to Morgue Attendant".

Test specificity with "give three dollars to Morgue Attendant".

Test largesse with "give pie to Morgue Attendant".

Test mixed-gift with "give dollar and cap to Morgue Attendant / get cap / give dollar and cap to morgue attendant / give me and dollar to attendant".

PURLOIN, used in the tests here, is a special debugging command that allows the player to acquire objects that wouldn't otherwise be possible to take. It is only active in non-release versions of the story. For more about debugging commands, see the chapter on Testing and Debugging.

289

Example Alpaca Farm

WI

A generic USE action which behaves sensibly with a range of different objects.

This example takes the ordering of grammar lines to its logical extreme, sorting the player's input into different categories depending on the kind and condition of the objects mentioned.

"Alpaca Farm"

Understand "use [an edible thing]" as eating.

Understand "use [a wearable thing]" as wearing.

Understand "use [a closed openable container]" as opening. Understand "use [an open openable container]" as closing.

Understand "use [something preferably held] on [a locked lockable thing]" as unlocking it with (with nouns reversed). Understand "use [something preferably held] on [an unlocked lockable thing]" as locking it with (with nouns reversed).

Understand "use [a switched off device]" as switching on.

Understand "use [something]" as using. Using is an action applying to one thing. Carry out using: say "You will have to be more specific about your intentions."

Understand "use [a door]" as opening. Understand "use [an open door]" as entering.

The Llama Pen is a room. North of the Pen is the gate. The gate is a door. North of the gate is the Rocky Path. The brown llama is an animal in the Llama Pen.

Appearance is a kind of value. The appearances are muddy, scruffy, fluffy, and dapper. The brown llama has an appearance. The brown llama is muddy. Before printing the name of the brown llama, say "[appearance] ". Before printing the name of the brown llama while grooming: say "now-[if appearance of the brown llama is less than dapper]merely-[end if]".

A grooming tool is a kind of thing. Understand "use [a grooming tool] on [something]" as grooming it with (with nouns reversed). Grooming it with is an action applying to two things. Understand "groom [something] with [something]" as grooming it with.

Carry out grooming it with:

if the appearance of the noun is less than dapper, now the appearance of the noun is the appearance after the appearance of the noun.

Report grooming it with:

say "You attend diligently to the appearance and hygiene of [the noun]."

Instead of using a grooming tool in the presence of the brown llama:

try grooming the brown llama with the noun.

The player carries some nail nippers, a slicker brush, and an apple. The apple is edible. The brush and the nippers are grooming tools. The player wears a sombrero.

The description of the nail nippers is "Ten inches long, to give you the necessary leverage to cut tough llama toenails. It still helps to soften them up by making the llama stand in a bucket of water first, though."

The description of the slicker brush is "Fine, angled soft bristles set into a broad back, perfect for removing mud from the coat of a long-woolled llama."

The industrial-strength blower is a fixed in place device in the Llama Pen. "Attached to the nearest wall, on its own movable boom, is an industrial-strength blower for doing llama hair."

Understand "use [switched off blower]" as switching on. Understand "use [switched on blower] on [brown llama]" as grooming it with (with nouns reversed). Instead of using the blower in the presence of the brown llama, try grooming the brown llama with the blower.

Test me with "use gate / use blower / use nippers / use brush / use apple / remove sombrero / use sombrero".

Whether we actually want a USE action is a subject of some theoretical debate in the IF community. On the one hand, it helps avoid guess-the-verb problems where the player cannot figure out what term to use in order to express a fairly simple idea. On the other, it encourages the player to think that all items have one and exactly one use, rather than getting him to consider the range of possibilities that arise from having a complex vocabulary.

368

Example Apples

WI

Prompting the player on how to disambiguate otherwise similar objects.

Inform by default detects whether two objects can be disambiguated by any vocabulary available to the player. If so, it asks a question; if not, it picks one of the identical objects at random.

Generally this produces good behavior. Occasionally, though, two objects have some distinguishing characteristic that doesn't appear in the object name. For instance, suppose we've created a class of apples that can be told apart depending on whether they've been bitten or not:

An apple is a kind of thing. Consumption is a kind of value. The consumptions are pristine and bitten. An apple has a consumption. The description of an apple is "It is [consumption]."

Understand the consumption property as describing an apple.

The player can meaningfully type

>EAT BITTEN APPLE

or

>EAT PRISTINE APPLE

but if he types

>EAT APPLE

Inform will, annoyingly, ask

Which do you mean, an apple or the apple?

This gives the player no indication of why Inform is making a distinction. So here we add a special "printing the name" rule to get around that situation:

"Apples"

Orchard is a room.

An apple is a kind of thing. Consumption is a kind of value. The consumptions are pristine and bitten. An apple has a consumption. The description of an apple is "It is [consumption]."

Understand the consumption property as describing an apple.

Before printing the name of an apple while asking which do you mean: say "[consumption] ". Before printing the plural name of an apple while asking which do you mean: say "[consumption] ".

The player carries three apples.

Instead of eating a pristine apple (called the fruit):
say "You take a satisfying bite."
now the fruit is bitten.

Instead of eating a bitten apple (called the fruit):
say "You consume the apple entirely."
now the fruit is nowhere.

Inform will also separate the bitten from the pristine apples in inventory listings and room descriptions, even though it's not clear why; we can improve on that behavior thus:

Before listing contents: group apples together.

Rule for grouping together an apple (called target):
let source be the holder of the target;
say "[number of apples held by the source in words] apple[s], some bitten".

Before printing the plural name of an apple (called target):
let source be the holder of the target;
if every apple held by the source is bitten, say "bitten ";
if every apple held by the source is pristine, say "pristine ".

Test me with "i / eat apple / i / eat apple / pristine / i / eat apple / pristine / i".

★ Example WXPQ

Creating a more sensible parser error than "that noun did not make sense in this context".

The parser error "That noun did not make sense in this context" arises instead of "You can't see any such thing" when the player uses a command that could apply to any item in the game -- that is, a command such as

Understand "go to [any room]" as going directly to.
Understand "talk about [any subject]" as discussing.

...and so on. The idea here is that "You can't see any such thing" isn't a sensible rejoinder when the player doesn't really need to be able to see the object.

Nonetheless, "That noun did not make sense..." is itself a fairly dry and uninformative response, and we may want to override it to something more appropriate for the specific kind of context in which it might appear. For instance:

"WXPQ"

WXPQ Studio is a room. "After about 2 AM, no one is listening anyway, so you can more or less make up whatever you like to fill the airwaves."

John F Kennedy, Elvis, Ralph Nader, Tony Blair, and single-origin chocolate are things.

Understand "talk about [any thing]" or "discuss [any thing]" as discussing.
Discussing is an action applying to one visible thing.

Carry out discussing:

say "You babble for a while about your [one of]interest in[or]hatred of[or]passionate devotion to[or]conspiracy theory concerning[or]mother's secret love affair with[as decreasingly likely outcomes] [the noun]."

Rule for printing a parser error when the latest parser error is the noun did not make sense in that context error:

say "For once, you're at a loss for anything to say."

Test me with "discuss Elvis / discuss Kennedy / discuss chocolate / discuss narratology vs ludology debate".

Note that this solution works as simply as it does because we only have one command in the game that can apply to an "[any]" token. If we had several, we'd need to distinguish between the parser error attached to "discuss" and the parser error attached to "go to" (for instance). In that case, we might instead write something like

Rule for printing a parser error when the latest parser error is the noun did not make sense in that context error:

if the player's command includes "go":

say "There's no such place you know how to get to.";

otherwise:

say "For once, you're at a loss for anything to say."

☆☆☆ Example Walls and Noses

Responding to "EXAMINE WALL" with "In which direction?", and to "EXAMINE NOSE" with "Whose nose do you mean, Frederica's, Betty's, Wilma's or your own?"

Suppose we want our game to respond to "EXAMINE WALL" with "In which direction?", and to "EXAMINE NOSE" with "Whose nose do you mean, Frederica's, Betty's, Wilma's or your own?"

For the case of EXAMINE WALL, we need a way to determine whether every item being disambiguated is a direction. We'll start by making a "matched" adjective which will identify items being disambiguated:

"Walls and Noses"

Eight-Walled Chamber is a room. "A perfectly octagonal room whose walls are tinted in various hues."

Understand "wall" as a direction.

Definition: a direction is matched if it fits the parse list.

Definition: a room is matched if it fits the parse list.

Definition: a thing is matched if it fits the parse list.

Rule for asking which do you mean when everything matched is direction:
say "In which direction?"

Checking the parse list requires a bit of behind-the-scenes work with Inform 6. Fortunately, you don't have to understand this entirely in order to use the rest of the example:

To decide whether (N - an object) fits the parse list:

```
(- (FindInParseList({N})) -)

Include (-
[ FindInParseList obj i k marker;
  marker = 0;
  for (i=1 : i<=number_of_classes : i++) {
  while (((match_classes-->marker) ~= i) && ((match_classes-->marker) ~= -i))
  marker++;
  k = match_list-->marker;
  if (k==obj) rtrue;
  }
  rfalse;
];
-)
```

Now that we've defined our "matched" adjective, we can use it for other purposes as well -- even generating our own lists. Our second challenge was to respond to

EXAMINE NOSE with "Whose nose do you mean, Frederica's, Betty's, Wilma's or your own?"

Here we need to change the way the question is worded (not "which do you mean" but "whose nose do you mean"). We also have to the names of the noses as they're printed in this particular context, so that they don't repeat the word "nose" over and over. And -- as a point of good English style -- we also want "your own" nose always to be last on the list.

For this purpose we may want to use the built-in "Complex Listing" extension, which allows us to print specially ordered lists. So:

Include Complex Listing by Emily Short.

Wilma, Betty, and Frederica are women in the Eight-Walled Chamber.
Understand "lady" or "woman" as a woman. A nose is a kind of thing. A nose is part of every person.

Rule for asking which do you mean when everything matched is a nose:
prepare a list of matched things;
if your nose is an output listed in the Table of Scored Listing:
choose row with an output of your nose in the Table of Scored Listing;
now the assigned score entry is -1;
say "Whose nose do you mean, [the prepared list delimited in disjunctive style]?"

Rule for printing the name of a nose (called target) while asking which do you mean :
if everything matched is a nose:
if the target is part of a person (called owner):
if the owner is the player, say "your own";
otherwise say "[the owner][apostrophe]s";
otherwise:
make no decision.

Understand "own" or "mine" as your nose.

Test me with "x wall / north / x nose / mine".

424



Example Cave-troll

WI

Determining that the command the player typed is invalid, editing it, and re-examining it to see whether it now reads correctly.

Novice players of interactive fiction, unfamiliar with its conventions, will often try to add extra phrases to a command that the game cannot properly parse: HIT DOOR WITH FIST, for instance, instead of HIT DOOR.

While we can deal with some of these instances by expanding our range of actions, at some point it becomes impossible to account for all the possible prepositional

phrases that the player might want to tack on. So what do we do if we want to handle those appended bits of text sensibly?

We could go through and remove any piece of text containing "with ..." from the end of a player's command; the problem with that is that it overzealously lops off the ends of valid commands like UNLOCK DOOR WITH KEY, as well. So clearly we don't want to do this as part of the "After reading a command..." stage.

A better time to cut off the offending text is right before issuing a parser error. At that point, Inform has already determined that it definitely cannot parse the instruction as given, so we know that there's something wrong with it.

The next problem, though, is that after we've edited the player's text we want to feed the corrected version back to Inform and try once more to interpret it.

This is where we have a valid reason to write a new "rule for reading a command". We will tell Inform that when we have just corrected the player's input to something new, it should not ask for a new command (by printing a prompt and waiting for another line of input); it should instead paste our stored corrected command back into "the player's command" and proceed as though that new text had just been typed.

Thanks to John Clemens for the specifics of the implementation.

"Cave-troll" by JDC

Section 1 - The Mechanism

The last command is a text that varies.

The parser error flag is a truth state that varies. The parser error flag is false.

Rule for printing a parser error when the latest parser error is the only understood as far as error and the player's command matches the text "with":

```
now the last command is the player's command;
now the parser error flag is true;
let n be "[the player's command]";
replace the regular expression ".*" with "(.*)" in n with "with \1";
say "(ignoring the unnecessary words '[n]')[line break]";
replace the regular expression "with .*" in the last command with "".
```

Rule for reading a command when the parser error flag is true:

```
now the parser error flag is false;
change the text of the player's command to the last command.
```

Section 2 - The Scenario

The Cave is a room.

The troll is a man in the cave.

The player carries a sword.

The chest is a locked lockable container in the cave.

Test me with "attack troll with sword / unlock chest with sword / attack troll as a test".

A caveat about using this method in a larger game: "parser error flag" will not automatically control the behavior of any rules we might have written for Before reading a command... or After reading a command..., so they may now fire at inappropriate times. It is a good idea to check for parser error flag in those rules as well.

304



Example Down in Oodville

WI

Offering the player a choice of numbered options at certain times, without otherwise interfering with his ability to give regular commands.

Now and then in IF there is a situation where we need to ask the player for a numbered choice rather than an ordinary action command. What's more, that numbered choice might change during the game, so we don't want to just hard-wire the meanings of "1", "2", and "3" whenever the player types them.

A better trick is to keep a list or table (we'll use a table here because it involves slightly less overhead) recording what the player's numerical choices currently mean. Then every time the player selects a number, the table is consulted, and if the number corresponds to something, the player's choice is acted on.

In our example, we'll have a transporter pad that can take the player to any room in the game that he's already visited. (Just for the sake of example, we'll start him off with a few pre-visited rooms.)

"Down in Oodville"

Section 1 - Method

Understand "[number]" as selecting.

Selecting is an action applying to one number.

Check selecting: [assuming we don't want to be able to transport from just anywhere]

if the player is not on the transporter pad:

say "You can transport only from the transporter pad. From other places than the transporter room, you can HOME to your base ship, but not leap sideways to other locations.";

empty the transport options instead.

Check selecting:

if the number understood is greater than the number of filled rows in the Table of Transport Options or the number understood is less than one:

say "[The number understood] is not a valid option. ";

list the transport options instead.

Carry out selecting:

let N be the number understood; [not actually a necessary step, but it makes the next line easier to understand]

choose row N in the Table of Transport Options;

if the transport entry is a room:

move the player to the transport entry;

otherwise:

say "*** BUG: Improperly filled table of transport options ***" [It should not be possible for this to occur, but we add an error message for it so that, if it ever does, we will know what is causing the programming error in our code]

To list the transport options:

let N be 1;

say "From here you could choose to go to: [line break]";

repeat through the Table of Transport Options:

say " [N]: [transport entry][line break]";

increment N.

To empty the transport options:

repeat through the Table of Transport Options:

blank out the whole row; [first we empty the table]

To load the transport options:

repeat with interesting room running through visited rooms which are not the Transporter Room:

choose a blank row in the Table of Transport Options;

now the transport entry is the interesting room.

Table of Transport Options

transport

an object

with 3 blank rows. [In the current scenario, the number of blank rows need never be greater than the number of rooms in the game, minus the transport room itself.]

Understand "home" as homing. Homing is an action applying to nothing.

Check homing:

if the player is in the Transporter Room:

say "You're already here!" instead.

Carry out homing:

move the player to the transporter room.

Section 2 - Scenario

The Transporter Room is a room.

Oodville is a visited room.

Midnight is a visited room. The Diamond City is west of Midnight.

The transporter pad is an enterable supporter in the Transporter Room. "The transporter pad in the middle of the floor is currently dull blue: powered but unoccupied."

After entering the transporter pad:

say "The transporter beeps and glows amber as you step onto its surface. A moment later a hologram displays your options. [run paragraph on]";
empty the transport options;
load the transport options;
list the transport options.

Test me with "get on pad / 0 / -1 / 8 / 2 / look / w / home / get on pad / get off pad / 3".

If we wanted to replace the regular command structure entirely with numbered menus, or use menus to hold conversation options, we could: several Inform extensions provide these functions.

373



Example Cloves

WI

Accepting adverbs anywhere in a command, registering what the player typed but then cutting them out before interpreting the command.

It has sometimes been suggested that IF should allow for the player to use adverbs, so that doing something "carefully" will have a different effect from doing it "quickly". There are several inherent challenges here: it's a good idea to make very sure the player knows all his adverb options, and the list of possibilities should probably not be too long.

Another trick is that adverbs complicate understanding commands, because they can occur anywhere: one might type >GO WEST CAREFULLY or >CAREFULLY GO WEST, and ideally the game should understand both. After reading a command is the best point to do this sort of thing, because we can find adverbs, interpret them, and remove them from the command stream. So:

"Cloves"

Manner is a kind of value. The manners are insouciantly, sheepishly, and defiantly.

Now we have, automatically, a value called manner understood to be used whenever parsing manners, and we can use this even during the "after reading a command" stage, so:

After reading a command:

if the player's command includes "[manner]":
cut the matched text;
otherwise:
say "But how, my dear boy, how? You simply can't do something without a pose. Thus far you have mastered doing things [list of manners].";
reject the player's command.

When play begins:

now the left hand status line is "Behaving [manner understood]";

now the right hand status line is "[location]";
now the manner understood is insouciantly.

The Poseur Club is a room. "Lady Mary is laid out on a sofa, her wrists bandaged importantly[if the manner understood is insouciantly] -- and she looks all the more depressed by your indifference to her state[end if]; Salvatore is at the gaming table, clutching his hair with both hands[if the manner understood is defiantly] -- though he looks up long enough to snarl in response to that expression of yours[end if]; Frackenbush is muttering lines from another of his works in progress, as though poetry has nearly made him mad[if the manner understood is sheepishly]. But he spares you a reassuring smile. He's not a bad fellow, Frackenbush[end if].

The usual people, in short."

Instead of doing something other than waiting or looking:
say "Dear. No. That would smack of effort."

Instead of waiting when the manner understood is sheepishly:
say "You scuff your foot against the ground for a moment, and allow a seemly blush to creep over your cheek. It's quite effective, you are sure, though you can't look up and see how it is going."

Instead of waiting when the manner understood is insouciantly:
say "Thrusting your hands into your pockets, you whistle a jaunty tune.

'Do shut up,' says a Melancholy Poseur from over by the window."

Instead of waiting when the manner understood is defiantly:
say "You raise your chin and give a pointed glance around the room as though to say that you are waiting for someone; you are unembarrassed about waiting for her; you have by no means been stood up; and the first person to comment will receive a poke in the eye."

Before looking when the manner understood is sheepishly:
say "You gaze up from under your brows..."

Before looking when the manner understood is defiantly:
say "You cast a withering gaze over the room."

Before looking when the manner understood is insouciantly:
if turn count > 1,
say "You turn an eye to your surroundings, looking faintly-- just faintly-- amused."

Test me with "wait / wait insouciantly / sheepishly look / defiantly look / look insouciantly".

The qualification about turn count is to prevent this before message from occurring when the player first looks around the room (automatically) at the start of play.

Note that to test this example, one must type INSOUCIANTLY TEST ME, and not simply TEST ME: a poseur's work is never done.



Example Fragment of a Greek Tragedy

Responding to the player's input based on keywords only, and overriding the original parser entirely.

Apologies to the shade of A. E. Housman.

"Fragment of a Greek Tragedy"

Understand "restart/restore/save/quit" as "[meta-command]".

After reading a command:

if the player's command matches "[meta-command]", make no decision;
 say line break;
 repeat through Table of Current Topics:
 if the player's command includes topic entry:
 say "CHORUS: [reply entry][paragraph break]";
 follow the advance time rule;
 rule succeeds;
 say "[italic type] Pause.[roman type][line break]";
 follow the advance time rule;
 rule succeeds.

Table of Current Topics

topic	reply
"journey/trip/travel/came/arrived"	"Sailing on horseback, or with feet for oars?"
"horseback/legs/feet/oars"	"Beneath a shining or a rainy Zeus?"
"shining/rainy/weather/zeus"	"Mud's sister, not herself, adorns thy boots."

This would be a bit bare if we didn't provide the player with some sort of context at the outset, so let's put some remarks before the first command prompt:

Before reading a command while the turn count is 1:

say "CHORUS: O suitably-attired-in-leather-boots
 Head of a traveller, wherefore seeking whom
 Whence by what way how purposed art thou come
 To this well-nightingaled vicinity?
 My object in inquiring is to know.
 But if you happen to be deaf and dumb
 And do not understand a word I say,
 Then wave your hand, to signify as much."

This "turn count" condition is why it was useful to follow the advance time rule in "after reading a command": the game (or drama, if you like) will continue to count moves elapsed even though the rest of Inform's command parsing and world model is being ignored. In a longer and more ambitious implementation of this idea, we might want to allow scenes to govern the behavior and responses of the Chorus.

And then to give the whole exchange a play's format:

The Stage is a room.

The room description heading rule is not listed in the carry out looking rules.

When play begins:

now the command prompt is "YOU: ";

now left hand status line is "Fragment of a Greek Tragedy";

now right hand status line is "A. E. Housman".

(Because this example manipulates commands outside of the normal parser, the mechanism for TEST will not work here. Try typing commands such as: TELL CHORUS ABOUT JOURNEY / TELL CHORUS ABOUT FEET / TELL CHORUS ABOUT SHROPSHIRE / TELL CHORUS ABOUT ZEUS)

Chapter 7: Other Characters

*§7.1. Getting Acquainted; §7.2. Liveliness; §7.3. Reactive Characters;
§7.4. Barter and Exchange; §7.5. Combat and Death;
§7.6. Getting Started with Conversation; §7.7. Saying Simple Things;
§7.8. Saying Complicated Things; §7.9. The Flow of Conversation;
§7.10. Character Emotion; §7.11. Character Knowledge and Reasoning;
§7.12. Characters Following a Script; §7.13. Traveling Characters;
§7.14. Obedient Characters; §7.15. Goal-Seeking Characters; §7.16. Social Groups*

-  Contents of *The Inform Recipe Book*
-  Chapter 6: Commands
-  Chapter 8: Vehicles, Animals and Furniture
-  Indexes of the examples

§7.1. Getting Acquainted

Talking about characters presents some special challenges. For one thing, some characters are referred to by a proper name, but others are not: so the story might want to talk about "Jack" but also about "the drunk pedestrian". In the absence of other information, Inform attempts to divine our intentions based on the words with which we defined a new character: but we can always override its guess with an explicit statement, such as

[The Great Malefactor is proper-named.](#)

Belfry demonstrates further how titles are set at the start of play.

The relation between the player and the other characters is not always static, however. Sometimes we want the player to learn a character's name part-way through play, and start referring to "the drunk pedestrian" as "Fernando". Similarly, the status of another character may change due to some twist of the plot. **Gopher-wood** shows how to change the name of a character mid-story, and **Peers** handles changing the character's rank.

Alternatively, of course, the player character may already know some of the other characters when the story begins, even if the player does not. In that case, we may want to add a tag-line or so of identification to a character's name when he first appears in the story. **A Humble Wayside Flower** shows one way of doing this.

Another occasional challenge is dealing with such commands as EXAMINE DR. THISBY. The problem here is that Inform by default will understand the full stop after "Dr" to be the end of one command and the beginning of another, and will try to interpret "Thisby" as a verb. If we do have a story populated by such formally-addressed characters, we may turn to Punctuation Removal, which provides a phrase to remove the full stops in standard titles before attempting to interpret the command.

Other characters have physical characteristics as well as names, of course, and **Meet Market** demonstrates one way of implementing people with notable features.

Finally, in some IF, the roles of characters may change from playing to playing. If we are writing a replayable murder mystery, we might want to select a new culprit each time the story starts; for this, see **Clueless**.

★ See **The Human Body** for more on body parts and physical description

★ See **Memory and Knowledge** for a way to refer to characters whom the player knows about but who aren't currently in the room

↑ Start of Chapter 7: Other Characters

← Back to Chapter 6: Commands: §6.18. Alternatives To Standard Parsing

→ Onward to §7.2. Liveliness

↓ Example 31: ★ **Belfry** You can see a bat, a bell, some woodworm, William Snelson, the sexton's wife, a bellringer and your local vicar here.

↓ Example 135: ★ **Clueless** A murderer for the mystery is selected randomly at the beginning of the game.

↓ Example 238: ★ **Meet Market** A case in which relations give characters multiple values of the same kind.

↓ Example 32: ★★ **Gopher-wood** Changing the name of a character in the middle of play, removing the article.

↓ Example 315: ★★★ **Peers** The peers of the English realm come in six flavours - Baron, Viscount, Earl, Marquess, Duke and Prince - and must always be addressed properly. While a peerage is for life, it may at the royal pleasure be promoted.

↓ Example 237: ★★★★ **A Humble Wayside Flower** Relations track the relationships between one character and another. Whenever the player meets a relative of someone he already knows, he receives a brief introduction.

§7.2. Liveliness

A character who sits still in a chair and does nothing is much less convincingly alive than one who seems to be pursuing some sort of personal agenda. There are all sorts of ways to achieve this, but the least challenging is by introducing some random change to descriptions, and by giving a character some very simple routine behavior to carry out.

For instance, we'll often want the characters in a room to be described doing different things every time we look at them. **Camp Bethel** shows how this may be done.

Every turn rules lend some sprightliness to otherwise-silent characters, as well:

Every turn when the player can see Mrs MacGillicuddy:

say "Mrs. MacGillicuddy vacuums around [a random fixed in place thing which is in the location]."

We might expand on this by providing a whole table of things for Mrs MacG. to cycle through, or a text variation effect that gives her different activities every turn, as in

Every turn when the player can see Mrs MacGillicuddy:

say "Mrs. MacGillicuddy [one of]vacuums around the furniture[or]tries to remove gum

from the underside of the desks[or]causes a racket by testing the smoke alarm[or]makes a pointed comment or two about them as sit by idly while someone works her fingers to the bone[as decreasingly likely outcomes]."

This is no great innovation in characterization by itself, but it does at least remind the player that the character is alive and moving about, even when he isn't paying attention to her.

Annoyotron Jr demonstrates a character who actively tries to get our attention, and whose routine of behavior changes just slightly if we show signs of having reacted to him.

Lean and Hungry implements a classic thief, a character who doesn't interact with the player much except to pick up valuable objects he finds left around the map. Later we will see rather more advanced ways to make characters act on their own goals and plans, but this kind of simple behavior is easily rigged as part of an every turn rule.

Finally, **Text Foosball** extends the every-turn-rule idea to create an opponent who joins us in a randomized game of table soccer.

With animal characters, this kind of repetitive, semi-random behavior is often adequate: we don't expect animals to talk, or pursue steady goals, or to take an interest in what we do in their presence (unless it involves a food they like to eat).

For people, we are likely to need an assortment of additional techniques.

★ See **Animals** for a domestic cat and dog

- ↑ Start of Chapter 7: Other Characters
 - ← Back to §7.1. Getting Acquainted
 - Onward to §7.3. Reactive Characters
 - ↓ Example 153: ★ **Annoyotron Jr** A child who after a certain period in the car starts asking annoying questions.
 - ↓ Example 67: ★★ **Camp Bethel** Creating characters who change their behavior from turn to turn, and a survey of other common uses for alternative texts.
 - ↓ Example 76: ★★★ **Lean and Hungry** A thief who will identify and take any valuable thing lying around that he is able to touch.
 - ↓ Example 140: ★★★★ **Text Foosball** A game of foosball which relies heavily on every-turn rules.
-

§7.3. Reactive Characters

As we observe characters, so they observe us. Those who seem to have no awareness of what the player is doing often come across more like waxworks than like people. **Zodiac** demonstrates a scenario where the watchful presence of a dangerous criminal keeps the player from doing what he otherwise might, while **Police State** expands on this idea with a policeman who reacts to entire types of behavior in his presence, regardless of whether the culprit is the player or a third party. **Noisemaking** has a crow who will fly away in response to any loud noises the player makes.

And, of course, we definitely want to have characters react to being looked at or otherwise interfered with. **Search and Seizure** implements a smuggler who reacts when we try to confiscate his possessions. **Pine 1** gives us a sleeping princess who can be woken by a variety of methods.

We wrap up this section with two complete puzzle scenarios that demonstrate what can be achieved by giving characters reactions to the player's behavior. **A Day For Fresh Sushi** has a fish who watches the player's actions and comments on them, while the live furnishings in **Revenge of the Fussy Table** instead comment every turn on the current state of the world, until the player has successfully sorted out all their complaints.

-
- ↑ Start of Chapter 7: Other Characters
 - ← Back to §7.2. Liveliness
 - Onward to §7.4. Barter and Exchange
 - ↓ Example 96: ★ **Zodiac** Several variations on "doing something other than...", demonstrating different degrees of restriction.
 - ↓ Example 154: ★ **Pine 1** Pine: Using a scene to watch for the solution of a puzzle, however arrived-at by the player.
 - ↓ Example 187: ★ **Police State** Several friends who obey you; a policeman who doesn't (but who takes a dim view of certain kinds of antics).
 - ↓ Example 38: ★★ **Search and Seizure** A smuggler who has items, some of which are hidden.
 - ↓ Example 197: ★★★ **Noisemaking** Creating a stage after the report stage of an action, during which other characters may observe and react.
 - ↓ Example 82: ★★★★ **Revenge of the Fussy Table** A small game about resentful furniture and inconvenient objects.
 - ↓ Example 112: ★★★★★ **A Day For Fresh Sushi** A complete story by Emily Short, called "A Day for Fresh Sushi", rewritten using Inform 7. Noteworthy is the snarky commenter who remarks on everything the player does, but only the first time each action is performed.
-

§7.4. Barter and Exchange

By default, Inform characters are a bit grudging about giving and sharing objects: they react with disinterest when they're shown things and refuse everything they're offered.

If we'd like to change this, we can simply remove the default "block giving" rule, as in

[The block giving rule is not listed in the check giving it to rules.](#)

If we do this, giving items to characters will have the result of moving our possessions to the other person's inventory. Of course, without more customization, the player may not ever be able to persuade the other character to return his possessions. **Bribery** demonstrates a scenario in which a character will accept gifts if they are interesting to him, and respond with a changed attitude to the player.

Barter Barter expands further on this by allowing other characters to trade things with one another.

★ See **Modifying Existing Commands** for ways to allow the player to give or show things that he isn't currently holding

★ See **Actions on Multiple Objects** for an implementation of giving that allows the player to offer multiple objects at once, where their combined value determines whether they are accepted

★ See **Money** for ways to keep track of cash flow, physical money objects, and price negotiations

↑ Start of Chapter 7: Other Characters

← Back to §7.3. Reactive Characters

→ Onward to §7.5. Combat and Death

↓ Example 392: ★★ **Bribery** A GIVE command that gets rid of Inform's default refusal message in favor of something a bit more sophisticated.

↓ Example 207: ★★★ **Barter Barter** Allowing characters other than the player to give objects to one another, accounting for the possibility that some items may not be desired by the intended recipients.

§7.5. Combat and Death

Not all characters are friendly, and there are times when we may want to include a fight sequence. There are a number of ways to approach this, depending on whether we want to offer the player a random outcome, a predetermined one, or a combat sequence that depends partly on strategy or on having the proper equipment.

Lanista 1 demonstrates randomized combat in the style of a role-playing game. The player has a partially random chance of doing any given amount of damage; both the player and his opponent have hit points, and whichever one runs out first dies. **Lanista 2** continues this idea, but includes weapons that affect the amount of damage done. **Red Cross** by itself implements a command that we might use to find out how strong characters are at the moment.

A word of warning about designing such sequences: a player who gets a roll he doesn't like always has the option of UNDOing a turn and re-rolling. This means that he can always win a random battle sooner or later; bad luck only means that it takes him longer (so he gets more bored and irritated as he plays through). It is possible to turn off UNDO implementation with

Use UNDO prevention.

...but there is a good chance that this will irritate players in itself. Role-playing-style combat scenarios need careful design, lest they actively make a story less fun.

In a slightly more realistic setting, combat leaves physical remains behind, unless we're wielding some kind of futuristic weapon that evaporates our opponents entirely: **Puff of Orange Smoke** demonstrates characters who leave corpses behind when they die, while **Technological Terror** more tamely explodes robots into numerous component parts.

Finally, we can imagine some scenarios in which, instead of allowing characters to strike at each other for random damage, we want to introduce an element of strategy. **Don Pedro's Revenge** shows the rudiments of a system in which the characters can make different kinds of attack depending on where they are in a room filled with perches, barrels, and other swashbuckler props.

★ See **Saving and Undoing** for more discussion of handling random behavior in games

- ⬆ Start of Chapter 7: Other Characters
 - ⬅ Back to §7.4. Barter and Exchange
 - ➡ Onward to §7.6. Getting Started with Conversation
 - ⬇ Example 130: ★ **Lanista 1** Very simple randomized combat in which characters hit one another for a randomized amount of damage.
 - ⬇ Example 191: ★ **Red Cross** A DIAGNOSE command which allows the player to check on the health of someone.
 - ⬇ Example 201: ★★ **Puff of Orange Smoke** A system in which every character has a body, which is left behind when the person dies; attempts to do something to the body are redirected to the person while the person is alive.
 - ⬇ Example 293: ★★★ **Lanista 2** Randomized combat in which the damage done depends on what weapons the characters are wielding, and in which an ATTACK IT WITH action is created to replace regular attacking. Also folds a new DIAGNOSE command into the system.
 - ⬇ Example 113: ★★★★ **Don Pedro's Revenge** Combat scenario in which the player's footing and position changes from move to move, and the command prompt also changes to reflect that.
 - ⬇ Example 127: ★★★★ **Technological Terror** A ray gun which destroys objects, leaving their component parts behind.
-

§7.6. Getting Started with Conversation

Traditionally, conversation is one of the most difficult things to program in interactive fiction, because of the number of factors affecting the outcome of everything the player does. While it's acceptable for >EXAMINE POT to produce the same response every time the player types it, it's a bit less acceptable for ASK JOE ABOUT HIS ADULTERY to make Joe react the same way every time.

Conversation implementations often need to keep track of a lot of information: what else is going on in the model world, what the character knows, what plot phase we've reached, what mood the character is in, what else we've recently been talking about, whether we've said the same thing before (and how many times); and so on. Later in this chapter we will look at ways to model character knowledge and mood.

Then, too, we have the problem of how the player communicates his conversational intentions to the story. Technology has not yet advanced to the point where a player can simply type in remarks in full natural English and have the character detect the significance, emotional tone, and subtext, if any: so we can't have RACHEL, THIS DESSERT TASTES LIKE FEET or WILL, LOOK! OUR SINISTER METAL FOES ARE APPROACHING! or BOSS, I WOULD BE DELIGHTED TO FILE ANOTHER TPB REPORT.

The challenge is to create an interface that is both easy for the player to use and expressive enough to be interesting. We will look at some of the common solutions in "Saying Complicated Things".

The examples in the following sections point out ways to approach common conversation problems. None of them will offer an adequate system if we want to write a very conversationally rich story, however. This is partly because a thorough conversation system requires quite a lot of code in its own right. It's also partly because there is no one right solution to the problem of conversation design. Different games will have quite different requirements. When making decisions about a new story we have planned, it may be useful to glance through the conversation extensions available for Inform: there are quite a few, offering a range of different interfaces. Even if none is exactly suited for our needs, they may suggest ways to solve particular implementation challenges.

At the other end of the scale, though, there are times when Inform's default implementation is too complicated for what we want to do: so we will start with ways to simplify conversation, before moving to all the exotic complexities.

Before we get into these details, though, we have a couple of examples that are literally about getting started with a conversation: **Mimicry** introduces the feature that we must greet other characters before beginning to speak to them; **The Gorge at George** corrects the player's attempts to use a TALK TO command where a different mode of interaction is appropriate instead.

-
-  Start of Chapter 7: Other Characters
 -  Back to §7.5. Combat and Death
 -  Onward to §7.7. Saying Simple Things
 -  Example 330:  **The Gorge at George** If the player tries to TALK TO a character, suggest alternative modes of conversation.
 -  Example 110:    **Mimicry** People who must be greeted before conversation can begin.
-

§7.7. Saying Simple Things

There are times when even the commands ASK and TELL are overkill: sometimes the player doesn't have much information to offer, so TELL is never useful, for instance. If we don't want to make any distinction between modes of conversation, we can conflate the actions so that ASK LUCIUS ABOUT OLLIVANDER, TELL LUCIUS ABOUT OLLIVANDER and LUCIUS, OLLIVANDER all do the same thing: see **Sybil 1**.

If we are frequently permitting the player to say things like LUCIUS, OLLIVANDER as shorthand for "talk to Lucius about Ollivander", then we may also want to allow LUCIUS, OLLIVANDER? This makes the player character seem a bit slow (or at least Laconic), but it is an effective interface in some cases. The trick is that the question mark at the end of the command may prevent Inform from recognizing the keyword; should that problem arise, we may want to use Punctuation Removal to erase question marks from the player's command before attempting to interpret it.

Along the same lines, there are situations in conversation where similar commands do not correspond to the same actions within Inform; if we're careless about this, we may force the player to guess which vocabulary we want him to use, which is always vexing. Some cases to look out for:

Inform has actions for "saying yes" and "saying no". Sometimes this is useful, but sometimes we want YES and SAY YES TO FRED to do the same thing. **Sybil 2** shows how to roll these responses into one; **Proposal** expands on the idea to show more ways in which a player could reasonably answer a question put by another character.

Again, if we want ASK SYBIL ABOUT CAKE to do the same thing as SHOW CAKE TO SYBIL, we might use the technique in **Nameless** to make objects into valid topics of conversation, and to make ASK and SHOW behave the same way.

Finally, if we want to be able to ASK and TELL an inanimate object -- say, a computer -- about something, we may use the extension Inanimate Listeners to add this capability.

★ See **Remembering, Converting and Combining Actions** for ways to redirect one conversation command to another conversation topic

★ See **Varying What Is Read** for a way of asking the player trivia questions that he can answer only on the next turn

-
- ↑ Start of Chapter 7: Other Characters
 - ← Back to §7.6. Getting Started with Conversation
 - Onward to §7.8. Saying Complicated Things
 - ↓ Example 91: ★ **Sybil 1** Direct all ASK, TELL, and ANSWER commands to ASK, and accept multiple words for certain cases.
 - ↓ Example 93: ★★ **Sybil 2** Making the character understand YES, SAY YES TO CHARACTER, TELL CHARACTER YES, ANSWER YES, and CHARACTER, YES.
 - ↓ Example 171: ★★ **Proposal** Asking the player a yes/no question which he must answer, and another which he may answer or not as he chooses.
 - ↓ Example 297: ★★ **Nameless** ASKING someone about an object rather than about a topic.
-

§7.8. Saying Complicated Things

As we saw in the overview, there are challenges in choosing the commands with which the player will communicate to the story. Two common approaches are ASK/TELL conversation, where the player can ask or tell characters about keywords, as in ASK JILL ABOUT JACK or TELL FARMER ABOUT CHICKEN COOP, and menu-based conversation, where the player is offered a list of things to say and must pick one (often by number), as in

- 1) Ask Jill where Jack went.
- 2) Tell Jill that the chicken coop was robbed.

or, sometimes,

- 1) "Jill, have you seen your no-good layabout brother Jack anywhere?"
- 2) "Look, Farmer Jill, I think a fox got into the chickens."

The problem with ASK/TELL conversation is that it can feel undirected - if the player doesn't know which keywords to ask or tell about next, he gets stuck. It also doesn't always provide much sense of ongoing context or conversational flow, since the player can ask lots of unrelated questions and jump around a lot. What's more, sometimes the thing the player character asks isn't quite the question the player had in mind. If we type ASK JILL ABOUT JACK, Jill could wind up answering any of a number of questions - where Jack is, how old Jack is, whether Jack committed the recent murder, and so on. The player doesn't have much fine control over the conversation. Nonetheless, this is sometimes just what we want: **Farewell** implements a moderately sophisticated system along these lines, which keeps track of what the player has already said and allows him to review past conversation.

Menu-based conversation solves most of these problems: a branching tree of conversation choices maintains a consistent flow of discussion, it's hard for the player to run out of things to say, and the player always knows what his character is about to say. But there are compensating flaws. For one thing, a menu doesn't allow for many surprises. The player can see all the conversation the story has to offer by working methodically through all the menu branches. (This problem is sometimes referred to as the "lawnmower effect", since the process of seeing all the conversation is like the process of running a lawnmower over every inch of the lawn. It becomes a chore rather than an entertainment.) Menu systems can be long-winded to set up and therefore none are exemplified here, but several have been released as extensions for Inform.

Since about 2001, more and more IF has used a sort of compromise method: the player is allowed to ask or tell about keywords, but he's sometimes given prompts about things to say that follow naturally on the conversation he was just having, as in

You could ask where Jack is.

Moreover, when he asks about a topic where many comments are possible, he'll be allowed to clarify, either using a menu or through a disambiguation question such as

>ask Jill about Jack

Do you want to ask where Jack is, how old Jack is, or whether Jack committed the recent murder?

Sweeney implements one such hybrid type of conversation.

A third option is to take away almost all the player's expressiveness and give him just one command, TALK TO. The player can TALK TO characters whenever he wants, and the story will pick the most appropriate thing for him to talk about. This works best in works with few or simple puzzles and a fast-moving, constrained plot, where the player will keep having new things to talk about. **Cheese-makers** demonstrates this.

Finally, a few extreme games try to fake natural language understanding by looking for keywords in the player's input, rather than an exact grammar. This is perilous, because it is all too easy for the story to completely misunderstand what the player meant to type. Nonetheless, for the sake of example, see **Complimentary Peanuts**, in which the incomprehension is partly excused by the fact that the player is talking to someone a bit hard of hearing.

-
-  Start of Chapter 7: Other Characters
 -  Back to §7.7. Saying Simple Things
 -  Onward to §7.9. The Flow of Conversation
 -  Example 281:  **Farewell** People who respond to conversational gambits, summarize what they said before if asked again, and provide recap of conversation that is past.
 -  Example 282:  **Sweeney** A conversation where each topic may have multiple questions and answers associated with it, and where a given exchange can lead to new additions to the list.
 -  Example 168:  **Cheese-makers** Scenes used to control the way a character reacts to conversation and comments, using a TALK TO command.
 -  Example 376:  **Complimentary Peanuts** A character who responds to keywords in the player's instructions and remarks, even if there are other words included.
-

§7.9. The Flow of Conversation

All this discussion of conversation commands and ways to model dialogue doesn't address the higher-level design issue: how do we approach writing this material so that it has a rhythm and flow? How do we know when we've created enough conversation? How can we avoid sounding hopelessly stilted when the nature of IF implementation requires us to break our text into small snippets?

While most authors develop their own approaches, there is some general advice that may help, especially for works that have a strong narrative progression.

It helps to have the plot of the story, with all its component scenes, planned in advance. That doesn't mean there can't be any changes later, but having a list of the different scenes can help us remember the different contexts in which information can appear. If we're using Inform's scenes feature, we may even want to restrict some dialogue to be available only during a given scene.

The next step is to go through scene by scene and create the "spine" of the scene. What *must* be said during this section? Is there anything the player can't leave without knowing? If the player isn't moving the scene forward fast enough, will the other character or characters volunteer information in order to keep the pace going?

It often helps to draft a transcript showing what we imagine as the ideal playthrough of the scene - writing straight through can create a natural flow of dialogue - before dividing the dialogue into pieces for implementation.

Once the scene is complete enough for the player to get through from beginning to end, we can start filling it out. At this point, it sometimes helps to play through the scene a number of times and add new dialogue elements as we think of things that our character might reasonably want to say. Sometimes these additions will turn out to be short tangents from the main flow of a very directed scene; sometimes they might be important branches that lead the scene to an entirely alternate outcome. The main thing is to make sure that, if the scene needs to hit certain points before ending, none of our branches keep the player from returning to the subject at hand.

-  Start of Chapter 7: Other Characters
 -  Back to §7.8. Saying Complicated Things
 -  Onward to §7.10. Character Emotion
-

§7.10. Character Emotion

In a complex story, characters may evolve strong feelings about the player. Often we want to hint at the character's feelings through gesture and tone of voice - little things woven into dialogue and action sequences that might otherwise be unchanged. **Ferragamo Again** demonstrates creating phrases to give all our characters different ways to express their irritation at the player.

Then again, sometimes a discussion might produce quite spectacular results if a character is in the wrong mood. **Being Peter** shows the bare bones of an implementation in which a character's attitude rulebook is consulted to determine what her response will be - allowing for arbitrarily complicated outcomes.

-  Start of Chapter 7: Other Characters
 -  Back to §7.9. The Flow of Conversation
 -  Onward to §7.11. Character Knowledge and Reasoning
 -  Example 405:  **Being Peter** A set of rules determining the attitude a character will take when asked about certain topics.
 -  Example 170:   **Ferragamo Again** Using the same phrase to produce different results with different characters.
-

§7.11. Character Knowledge and Reasoning

A character may be endowed with knowledge and even reasoning skills. Relations form quite a good way of keeping track of such problems: for instance, we can allow characters to be acquainted with one another with a relation such as

[Lucy knows Lady Cardew.](#)

Or we might keep track of more complicated attitudes between characters, as in **Murder on the Orient Express**, in which some characters suspect others of the crime.

Alternatively, we might have a list of salient facts that are important in our story. We might declare these as values, and then characters could know, learn, and forget entries as appropriate:

[A fact is a kind of value. Some facts are defined by the Table of All Known Facts.](#)

[Knowledge relates various people to various facts. The verb to know \(he knows, they know, he knew, it is known\) implies the knowledge relation.](#)

Table of All Known Facts
fact summary
shoe-size "Lucy wears a size 9 shoe."
sunset-time "Sunset is at 8:22 PM this evening."

Lucy knows shoe-size.
Bob knows sunset-time and shoe-size.

Or again we might keep a whole database of information in a table: the characters in **Questionable Revolutions** know dates, countries, and a short description for each of several rebellions and popular uprisings, while in **Queen of Sheba**, Solomon is able to answer who, what, where, when, and why questions about a range of topics. This kind of approach is most useful when the characters need to display a deep knowledge of a particular field. The facts stored in the Table of All Known Facts, above, are comparatively sparse, because there we are designing a story in which not all data about the world is equally valuable: Lucy doesn't know the shoe size of every person in the story, because for some reason it is only her own shoe size that matters. On the other hand, the Table of All Known Facts can store different kinds of information, whereas the revolutions table has no way of storing shoe sizes or sunset times. And **Murder on the Orient Express** works differently again, because it is storing knowledge that concerns people and things that already exist in the world model, rather than abstract ideas. Our way of modeling character knowledge, in other words, will depend quite a lot on what kind of knowledge it is.

The possibilities of character reasoning are similarly broad, but **The Problem of Edith** introduces one kind: the character has a concept of how different conversation topics relate to one another, so that when she is asked about a new keyword, she picks a response that makes the question most relevant to the conversation already in progress.

We end with a longer scenario, in which we track what the character knows about the player and the conversational state: in **Chronic Hinting Syndrome**, the main character guides conversation in the direction he intends it to go, with the player's sometimes-reluctant participation.

★ See **Obedient Characters** for a character who needs to be taught how to perform actions before doing them

★ See **Characters Following a Script** for a programmable robot who can be given whole sequences of actions to perform

- ↑ Start of Chapter 7: Other Characters
 - ← Back to §7.10. Character Emotion
 - Onward to §7.12. Characters Following a Script
 - ↓ Example 241: **★☆☆ Murder on the Orient Express** A number of sleuths (the player among them) find themselves aboard the Orient Express, where a murder has taken place, and one of them is apparently the culprit. Naturally they do not agree on whom, but there is physical evidence which may change their minds...
 - ↓ Example 235: **★☆☆ The Problem of Edith** A conversation in which the main character tries to build logical connections between what the player is saying now and what went immediately before.
 - ↓ Example 278: **★★★★ Questionable Revolutions** An expansion on the previous idea, only this time we store information and let characters answer depending on their expertise in a given area.
 - ↓ Example 279: **★★★★ The Queen of Sheba** Allowing the player to use question words, and using that information to modify the response given by the other character.
 - ↓ Example 341: **★★★★★ Chronic Hinting Syndrome** Using name-printing rules to keep track of whether the player knows about objects, and also to highlight things he might want to follow up.
-

§7.12. Characters Following a Script

So far we've seen characters who will answer questions whenever the player feels like asking, and characters who will use some reasoning procedure to direct the conversation. There is a third option, often useful in IF with a fast-paced narrative: the character follows a conversational script, making sure to cover a series of points before the scene ends.

There are more and less tedious ways to implement this kind of scene. The worst case is one in which the player is not allowed to interrupt or ask any questions; he must merely wait until the character runs out of things to say. This can be useful and plausible in very small doses - say, two or three turns - but if the character has more information than that to impart, we may want to make the scene more interactive.

Pine 2 partly addresses this challenge: the character has a line of conversation that she wants to follow to its conclusion; we may ask questions along the way, but if we're silent, she'll take up the slack, and the scene won't end until she's done with what she has to say.

Another kind of script is a series of actions for the character to perform. **Robo** demonstrates a programmable robot that will observe what the player does, then try to emulate the actions later when switched into play-back mode. **Robo 2** extends this capacity to allow the robot to contain fifteen different scripts which the player can store, list, run, and erase.

Your Mother Doesn't Work Here offers a character with a list of tasks but whose plans can be interrupted by more urgent demands. This verges on not being a simple script any more: if we carry the idea to its natural conclusion, we get characters capable of planning scripts for themselves to accomplish their aims. This is conventionally called "goal-seeking".

★ See **Goal-Seeking Characters** for characters that work out plans for themselves in order to accomplish various outcomes

- ↑ Start of Chapter 7: Other Characters
 - ← Back to §7.11. Character Knowledge and Reasoning
 - Onward to §7.13. Traveling Characters
 - ↓ Example 426: ★ **Robo 1** A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.
 - ↓ Example 438: ★ **Your Mother Doesn't Work Here** Your hard-working mother uses a list as a stack: urgent tasks are added to the end of the list, interrupting longer-term plans.
 - ↓ Example 161: ★★★ **Pine 2** Pine: Adding a conversation with the princess, in which a basic set of facts must be covered before the scene is allowed to end.
 - ↓ Example 429: ★★★ **Robo 2** A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.
-

§7.13. Traveling Characters

There are a number of ways we can make characters navigate our map. We might reasonably want them to approach and follow the player (as in **Van Helsing**); or to allow the player to follow characters who have left the room (as in **Actaeon**).

Characters who are less interested in the player will more likely follow their own courses around the available geography, however. A character may move randomly from room to room, as demonstrated in **Mistress of Animals**; he may follow a path that we have specifically written in advance, as **Odyssey** shows; or, most elegantly, he may use the "best route" calculation to find the best possible way to a given target room, as seen in **Latris Theon**.

This final method is arguably the neatest solution to character movement, allowing for characters to act in sophisticated ways; if we incorporate the Locksmith extension, other characters will even unlock and open doors that are in their way. The chief catch is that it should not be used too profligately with large numbers of characters, since on slow machines the processing power required to plan all their travel will make a noticeable difference to the running speed of the story.

All the same, the constraints are not so severe as to preclude having a moderate number of route-finding characters all wandering around at once. This does introduce a new problem, however: movement descriptions can become hard to follow if every turn produces long reams of reports such as

Joe enters the room from the south.
Lawrence opens the gate.
Lawrence departs to the west.
Lucy comes in from above.
Ted enters the room from the south.
Bill departs to the west.

Patient Zero tackles this problem by calculating all of the character movement without printing any text; it then combines similar or related events into coherent paragraphs, as in

Rhoda and Antony walk into the Post Office. Rhoda could have been rolling in chocolate and Antony looks as though dipped in french vanilla.

or

Antony opens the iron gate. He goes through.

★ See **Doors, Staircases, and Bridges** for some technical details of allowing other characters to interact with doors when they're in rooms that don't contain the player

- ⬆ Start of Chapter 7: Other Characters
 - ⬅ Back to §7.12. Characters Following a Script
 - ➡ Onward to §7.14. Obedient Characters
 - ⬇ Example 77: ★ **Mistress of Animals** A person who moves randomly between rooms of the map.
 - ⬇ Example 39: ★★ **Van Helsing** A character who approaches the player, then follows him from room to room.
 - ⬇ Example 274: ★★★ **Odyssey** A person who follows a path predetermined and stored in a table, and who can be delayed if the player tries to interact with her.
 - ⬇ Example 302: ★★★ **Actaeon** A FOLLOW command allowing the player to pursue a person who has just left the room.
 - ⬇ Example 185: ★★★★ **Latris Theon** A person who can accept instructions to go to new destinations and move towards them according to the most reasonable path.
 - ⬇ Example 402: ★★★★★ **Patient Zero** People who wander around the map performing various errands, and in the process spread a disease which only the player can eradicate.
-

§7.14. Obedient Characters

Other characters can perform all the same activities that the player can; this does not always mean that they're willing to obey the player's instructions. By default, characters will refuse to obey commands of the form JULIA, WEST or ANTONY, TAKE THE PINCUSHION. Their objections can be overridden, however, and **The Hypnotist of Blois** implements a hypnotist who can make characters obedient at will.

In For Demonstration Purposes, the character is only capable of a few actions at the outset, but can be taught new ones if the player performs them first.

Often we want characters' obedience to be more selective. Just as the viewpoint character may be characterized in terms of what he will and will not do, so may others: **Generation X** demonstrates a character who will do what she's told, but who will comment unfavorably when the player asks for a nonsensical or repeated action, and who may eventually get fed up and leave.

Characters can be given moral objections to certain commands, as well: **Virtue** defines a few kinds of actions as bad, so that the character commanded will refuse to perform them.

Under Contract, more subtly, has the character object if the player's commands implicitly require any behavior he considers inappropriate: for instance, if the player commands him to put his pants in a container, he will work out that this requires the removal of the pants as a preliminary. If we want to implement a similar character, we may want to simply copy his

unsuccessful attempt rule and the table of his retorts, then replace his banter with lines of our choosing.

The little example **Latin Lessons** allows us to make characters clever about vague commands: we can, for instance, write rules so that CLARK, EAT will have Clark sensibly pick something edible, rather than having the parser ask what we want Clark to eat.

Finally, **Northstar** demonstrates how we might make Inform understand commands of the form ASK JOSH TO TAKE INVENTORY or ORDER JOAN TO WEAR THE ARMOR.

★ See **Characters Following a Script** for a programmable robot who can be given whole sequences of actions to perform

- ⬆ Start of Chapter 7: Other Characters
 - ⬅ Back to §7.13. Traveling Characters
 - ➡ Onward to §7.15. Goal-Seeking Characters
 - ⬇ Example 184: ★ **Virtue** Defining certain kinds of behavior as inappropriate, so that other characters will refuse indignantly to do any such thing.
 - ⬇ Example 186: ★ **The Hypnotist of Blois** A hypnotist who can make people obedient and then set them free again.
 - ⬇ Example 371: ★ **Latin Lessons** Supplying missing nouns and second nouns for other characters besides the player.
 - ⬇ Example 188: ★★ **Generation X** A person who goes along with the player's instructions, but reluctantly, and will get annoyed after too many repetitions of the same kind of unsuccessful command.
 - ⬇ Example 423: ★★★ **Northstar** Making Inform understand ASK JOSH TO TAKE INVENTORY as JOSH, TAKE INVENTORY. This requires us to use a regular expression on the player's command, replacing some of the content.
 - ⬇ Example 239: ★★★★ **For Demonstration Purposes** A character who learns new actions by watching the player performing them.
 - ⬇ Example 205: ★★★★★ **Under Contract** Creating a person who accepts most instructions and reacts correctly when a request leads implicitly to inappropriate behavior.
-

§7.15. Goal-Seeking Characters

Goal-seeking characters are the most advanced IF life-form: they want to achieve specific outcomes, and they are able to work out plans of approach in order to bring these things about. They walk to rooms, open containers to search for things, use keys and tools, and ask leading questions in conversation.

A really advanced implementation of goal-seeking behavior is beyond the scope of our examples (though extensions exist that treat the problem more thoroughly). We can accomplish a surprising amount without heavy customization, though, if we keep in mind three points of technique:

First: it helps to think abstractly and to create broadly-defined actions as a first step to more specific tasks. For instance, a character's goal might be to eat some dinner. He'd be equally

satisfied with spaghetti carbonara or with braised lamb shanks, but he needs to figure out which is available. So we might have our every turn rule (or whatever we're using to activate the character) say something like

Every turn when Clark is hungry:
try Clark dining.

Dining would then be an action we've defined specially, which looks around Clark's environment for suitable food; if it finds food, it issues a

try Clark eating the suitable food;

command; but if not, it sends Clark off to look for something likely. **The Man of Steel** demonstrates the use of this.

Second: though it doesn't actually contribute to the goal-seeking per se, lively reporting brings characters' generated behavior to life.

Clark eats a donut.

doesn't characterize Clark very much, even though the eating may be part of a subtle, intelligent plan to seduce Lois Lane. We'll do better if we replace a lot of the character reporting rules: to that end, see the example **The Man of Steel Excuses Himself**.

Third: goal-seeking characters notice when something is in the way of the action they want to perform. When that happens, they form a plan about how to remove the obstacle. We've already seen this kind of implementation on the player's behalf: the player will pick up items before eating them, say. We can use Before rules to do similar things for other characters, as in

Before Clark eating the wrapped candy:
try Clark unwrapping the candy;
if the candy is wrapped, stop the action.

Here we've set things up so that if Clark tries to eat the wrapped candy, he'll be interrupted by this other command; and if his unwrapping-the-candy attempt fails, he won't go on and eat the thing. **IQ Test** demonstrates a character who shows this kind of planning intelligence.

Because before-rules chain neatly, we can trigger whole plans of behavior if we have a sensible set, as in

Before someone entering a closed container: try the person asked opening the noun.
Before someone opening a locked container: try the person asked unlocking the noun.
Before someone unlocking a locked container: ...

We must exercise a little bit of care if it is possible for the chain of actions to produce an endless loop - e.g., the character trying to take a key that is inside the transparent, locked box that it opens might repeatedly try to open the box, first unlocking the box, first taking the key, first opening the box, ... **Boston Cream** is a fully-worked scenario that deals with such a set of conundra.

★ See **Traveling Characters** for characters who plan routes to locations and travel towards them

★ See **Event Scheduling** for characters who follow a pre-written schedule of activities

★ See **Plot Management** for having a central function direct all the characters in order to further the plot

↑ Start of Chapter 7: Other Characters

← Back to §7.14. Obedient Characters

→ Onward to §7.16. Social Groups

↓ Example 189: ★ **IQ Test** Introducing Ogg, a person who will unlock and open a container when the player tells him to get something inside.

↓ Example 203: ★★ **The Man of Steel** An escaping action which means "go to any room you can reach from here", and is only useful to non-player characters.

↓ Example 208: ★★★ **The Man of Steel Excuses Himself** Elaborating the report rules to be more interesting than "Clark goes west."

↓ Example 190: ★★★★★ **Boston Cream** A fuller implementation of Ogg, giving him a motivation of his own and allowing him to react to the situation created by the player.

§7.16. Social Groups

Crowds of characters introduce new challenges, because we often want to show them interacting with one another, or to describe individuals in less detail when a whole group is present.

Strictly Ballroom gives us a set of characters who pair off each turn, making sure to mention each one once, and leaving one unfortunate person behind as a wallflower: this exemplifies how we might use a behavioral rule not to dictate the behavior of each individual separately but rather to model a whole group together. **Happy Hour** does calculate movements for characters individually, but then collates the descriptions, creating a single paragraph to describe whatever group is currently in the room.

Characters can also have complicated attitudes to one another, and it can be helpful to use relations to track these. **Unthinkable Alliances** demonstrates the grouping of characters into alliance factions, while **The Abolition of Love** provides a host of relations to track love affairs, marriages, memberships in families, and mere mutual respect.

Emma combines these two effects: its characters move between social groups depending on how they feel about the others in their particular talking circle, and descriptions change depending on who is where in the room.

Lugubrious Pete's Delicatessen simulates a queue at a deli, in which the customers who most impress Pete get served first.

★ See **Traveling Characters** for groups of characters who move around and have their movements collated into a joint description

- ⬆ Start of Chapter 7: Other Characters
 - ⬅ Back to §7.15. Goal-Seeking Characters
 - ➡ Onward to Chapter 8: Vehicles, Animals and Furniture: §8.1. Bicycles, Cars and Boats
 - ⬇ Example 229: **★ Unthinkable Alliances** People are to be grouped into alliances. To kiss someone is to join his or her faction, which may make a grand alliance; to strike them is to give notice of quitting, and to become a lone wolf.
 - ⬇ Example 231: **★ The Abolition of Love** A thorough exploration of all the kinds of relations established so far, with the syntax to set and unset them.
 - ⬇ Example 436: **★ Lugubrious Pete's Delicatessen** In this evocation of supermarket deli counter life, a list is used as a queue to keep track of who is waiting to be served.
 - ⬇ Example 176: **★★ Strictly Ballroom** People who select partners for dance lessons each turn.
 - ⬇ Example 353: **★★ Emma** Social dynamics in which groups of people form and circulate during a party.
 - ⬇ Example 356: **★★ Happy Hour** Listing visible characters as a group, then giving some followup details in the same paragraph about specific ones.
-

Examples from Chapter 7: Other Characters

- ⬆ Start of this chapter
- ➡ Chapter 8: Vehicles, Animals and Furniture
- ⬇ Indexes of the examples

31

★ Example Belfry

WI

You can see a bat, a bell, some woodworm, William Snelson, the sexton's wife, a bellringer and your local vicar here.

"Belfry"

The Belfry is a room. A bat is in the Belfry. The bell is in the Belfry. Some woodworm are in the Belfry. A man called William Snelson is in the Belfry. A woman called the sexton's wife is in the Belfry. A man called a bellringer is in the Belfry.

In the Belfry is a man called the vicar. The indefinite article of the vicar is "your local".

Test me with "look".

135

★ Example Clueless

WI

A murderer for the mystery is selected randomly at the beginning of the game.

"When play begins" is the best point to initialize any aspects of the game that are meant to change between playings. For instance, in this scenario, we would randomly select one of the other characters to be guilty of murder:

"Clueless"

The murderer is a person that varies.

When play begins:

now the murderer is a random person who is not the player.

The Billiards Room is a room. Colonel Mustard and Professor Plum are men in the Billiards Room. Miss Scarlet and Mrs White are women in the Billiards Room.

Instead of examining the murderer:

say "[The noun] certainly looks fiendish!"

Test me with "x mustard / x plum / x scarlet / x white".

238

★ Example Meet Market

WI

A case in which relations give characters multiple values of the same kind.

We have already seen that we can give things value properties -- a lamp has a brightness, for instance. Relations give us additional flexibility: since we may relate various things to various values, it is possible to describe a thing as having more than one value at the same time.

To demonstrate:

"Meet Market" by "K M and Eric Rossing"

Feature is a kind of value. The features are snub-nosed, gangly, comely, bright-eyed, and sulky.

Appearance relates various persons to various features. The verb to appear means the appearance relation.

Meet Market is a room.

Bob, Carol, Ted, and Alice are people in the Meet Market.

Bob appears snub-nosed and gangly.

Ted appears sulky and snub-nosed.

Carol appears comely and bright-eyed.

Alice appears bright-eyed and comely.

Yourself appears sulky and gangly.

Instead of looking:

```
say "The snub-nosed ones: [list of people who appear snub-nosed][line break]";
say "The gangly ones: [list of people who appear gangly][line break]";
say "The comely ones: [list of people who appear comely][line break]";
say "The bright-eyed ones: [list of people who appear bright-eyed][line break]";
say "The sulky ones: [list of people who appear sulky][paragraph break]".
```

Test me with "look".

The same logic might be used to provide characters who have complex mood states: a person might be angry and sad, not merely one or the other -- feelings being what they are.

32



Example Gopher-wood

WI

Changing the name of a character in the middle of play, removing the article.

Suppose that we want a character who starts out with a general epithet ("the bearded man") but is later introduced to the player properly ("Japheth"). In that case, we want to be able to tell Inform to stop using an article once the character has been given a proper name. We can do this like so:

```
"Gopher-wood"
```

The Ark is a room. A bearded man is in the Ark.

Instead of examining the bearded man for the first time:

```
now the printed name of the bearded man is "Japheth";
now the bearded man is proper-named;
say "You peer at him a bit more closely and realize that it's Japheth."
```

Finally, we need to tell Inform to understand the man's name, but only when he's been introduced. For this purpose, we borrow from the chapter on Understanding:

```
Understand "Japheth" as the bearded man when the bearded man is proper-named.
```

Test me with "x japheth / x man / look / x japheth".

315



Example Peers

WI

The peers of the English realm come in six flavours - Baron, Viscount,

Earl, Marquess, Duke and Prince - and must always be addressed properly. While a peerage is for life, it may at the royal pleasure be promoted.

Almost all of this example is the flummery of pomp and circumstance: only the first two paragraphs really do anything.

"Peers" by Elizabeth II R

A title is a kind of value. The titles are Baron, Viscount, Earl, Marquess, Duke and Prince.

A peer is a kind of man. A peer has a title. A peer is usually a Baron. Before printing the name of a peer, say "[title]". Understand the title property as describing a peer.

The House of Lords is a room. Maltravers, Pollifax, Omnium and St Vincent are peers in the House of Lords. Omnium is a Duke. St Vincent is an Earl.

Ennobling is an action applying to one thing and one title.

Check ennobling:

if the noun is the player, say "The Sovereign is the fountain of honour, and may not be ennobled." instead;

if the noun is not a peer, say "Commoners should remain so." instead;

if the title of the noun is the title understood, say "But that is his title already." instead;

if the title of the noun is greater than the title understood, say "As he is already of the rank of [title of the noun], any such letters patent are liable to be deemed invalid, following the precedent of the Buckhurst Peerage Case (1876). Best not." instead.

Carry out ennobling:

now the title of the noun is the title understood.

Report ennobling:

say "'Whereas Our Parliament for arduous and urgent affairs concerning Us the state and defence of Our United Kingdom and the Church is now met at Our City of Westminster We strictly enjoining Command you upon the faith and allegiance by which you are bound to Us that the weightness of the said affairs and imminent perils considered (waiving all excuses) you be at the said day and place personally present with Us and with the said Prelates Great Men and Peers to treat and give your counsel upon the affairs aforesaid And this as you regard Us and Our honour and the safety and defence of the said Kingdom and Church and dispatch of the said affairs in nowise do you omit Witness Ourselves at Westminster the Fifth day of November in the 43rd year of Our Reign,' you say, with unpunctuated serenity. The new [noun] bows stiffly."

Understand "dub [someone] a/an [title]" as ennobling.

Test me with "dub st vincent a baron / dub maltravers a marquess / look / examine marquess".



Example A Humble Wayside Flower

Relations track the relationships between one character and another. Whenever the player meets a relative of someone he already knows, he receives a brief introduction.

First we define the relationships we choose to acknowledge:

"A Humble Wayside Flower"

Marriage relates one person to another (called the spouse). The verb to be married to means the marriage relation.

Fatherhood relates one person (called father) to various people. The verb to engender means the fatherhood relation.

For brevity, we will ignore the existence of mothers. It is a sad world.

Siblinghood relates a person (called A) to a person (called B) when a person who engenders A engenders B. The verb to be sibling to means the siblinghood relation.

Family relates a person (called A) to a person (called B) when A is married to B or A engenders B or B engenders A or A is sibling to B. The verb to be related to means the family relation.

A person can be known or unknown. After printing the name of an unknown person (called the alien):

```
if a known person (called the contact) is related to the alien:
  say " ([relation between alien and contact] of [the contact])";
  now the alien is known;
  rule succeeds.
```

To say relation between (first party - a person) and (second party - a person):

```
if the first party is married to the second party:
  if the first party is female, say "wife";
  otherwise say "husband";
  rule succeeds;
if the first party is sibling to the second party:
  if the first party is female, say "sister";
  otherwise say "brother";
  rule succeeds;
if the first party engenders the second party:
  say "father";
  rule succeeds;
if the second party is the father of the first party:
  if the first party is female, say "daughter";
  otherwise say "son";
  rule succeeds.
```

Pere Blanchard's Hut is a room. Percival Blakeney is a known man in the Hut. Marguerite is a woman in the Hut. Percival is married to Marguerite. Outside

from the Hut is the Garden. Louise is a woman in the Garden. The Road to Paris is west of the Garden. Armand St Just is a man in the Road. Louise is married to Armand. Monsieur St Just is a man. He engenders Armand and Marguerite.

Test me with "out / west / east / west".

Monsieur St Just never appears on the scene in this piece, but if we did put him somewhere the player could find him, he, too, would be properly introduced.

153

Example Annoyotron Jr

WI

A child who after a certain period in the car starts asking annoying questions.

"Annoyotron Jr"

The Minivan is a room. The Open Road is outside from the Minivan. Pete is a man in the Minivan. "Pete [if the player has been in the Minivan for 3 turns]is starting to look bored[otherwise]is playing with his travel activity book[end if]."

Every turn:

if the player has been in the Minivan for 5 turns, say "'Are we there [if saying no]now?'[otherwise]yet?' asks Pete.[end if]"

Instead of saying no:

say "'Oh,' says Pete. There is a blessed, momentary silence."

Instead of going to the Open Road:

say "You leap to your death.";
end the story.

Test me with "z / z / look / g / g / g / no / z / z / z / no / z / out".

67

Example Camp Bethel

WI

Creating characters who change their behavior from turn to turn, and a survey of other common uses for alternative texts.

"Camp Bethel"

Camp Bethel Kitchen is a room.

One use for text alternatives is to change the description of a room after first visiting. We've already seen, in the example "Slightly Wrong", how to do this with "[if visited] ... [otherwise] ... [end if]". But since the first description is printed once and the second description on all subsequent occasions, we could equally well write

The description of Camp Kitchen is "[one of]You've never been into the kitchen before, though you've spent many an hour in the dining lodge. The place is larger than you would have expected, and it has none of the fake rustic touches of the rest of the camp[or]A tidy, efficient industrial kitchen, without any of the kitsch rusticity found elsewhere[stopping]."

We might also want to liven up the behavior of people and animals, who are probably not doing the exact same thing every time we glance in their direction. There are more complex techniques for modeling the behavior of characters, as we will see in the chapters on Advanced Actions and Activities; but if we just want some textual variety, we might write something like:

Jeremy is a man in the Camp Bethel Kitchen. "Jeremy stands at his station, [one of]peeling white onions[or]briskly dicing onions[or]chopping celery[or]peeling carrots[or]tying fresh herbs together with string[or]putting all the vegetables into a large stock pot[or]watching over his boiling vegetable stock[cycling]."

And since (textual variation or not) we do want the player to be able to see all these objects:

Jeremy carries white onions, celery, carrots, and herbs. Jeremy's station is scenery in the kitchen. It is a supporter.

Jeremy is following a sequence of actions to do an implied task (still somewhat robotically, but it will do for now). Animals might be a bit more capricious, though:

Fluffy is an animal in the Camp Bethel Kitchen. "[one of]Fluffy is chasing its tail[or]Fluffy is staring out the window[or]Fluffy is rubbing itself against your leg[purely at random]."

A housefly is an animal in the Camp Bethel Kitchen. "A large housefly [one of]lands on a countertop[or]flies around noisily[or]circles Jeremy's chef hat[at random]."

The housefly's description is merely "at random" rather than "purely at random" because we want to show it doing a different thing each turn, whereas Fluffy could plausibly stare out the window for five turns in a row.

There are more complex ways to change and override the initial descriptions of people and things; if text alternatives do not get us far enough, we can turn to the "rule for writing a paragraph about," documented in the Activities chapter.

Another frequent use of text alternatives is to give characters a bit of variety in things they're likely to say many times in the course of a game:

Instead of telling Jeremy about something:

say "Jeremy looks [one of]surprised[or]intrigued[or]nonplussed[at random].
[one of]You don't say[or]That's very interesting[or]Do go on[or]I wish I'd known that sooner[at random]!".

Or, with somewhat more complexity:

Instead of asking Jeremy about something:

say "[one of]Sorry,[or]I'm afraid[or]Hm,[at random] [one of]I don't know much about that[or]you've got me there[or]I haven't the faintest[at random],' Jeremy [one of]drawls[or]replies[or]comments[or]exclaims[at random]";

say "[one of][or] huskily[or] throatily[or] silkily[or] in a deep manly voice[as decreasingly likely outcomes]."

Notice that, in that last line, our first option is entirely blank. If we put nothing as an element of the text alternatives list, this means that printing nothing at all is a viable alternative. In fact, we've made this the most common probability out of the decreasingly likely outcomes, so that five times in fifteen, or a third of the time the text is run, there will be no modifier printed at all.

Test me with "look / g / g / g / ask Jeremy about his feelings for me / ask jeremy about his amnesia / tell Jeremy about my unborn child".

As this example (alas) reveals, text alternatives will not go all the way toward making our characters into compelling conversationalists; we will have to wait until we know more about Actions. But at least we have abolished the default responses, and given Jeremy a touch of personality, however witless.

76



Example **Lean and Hungry**

WI

A thief who will identify and take any valuable thing lying around that he is able to touch.

"Lean and Hungry"

Substance is a kind of value. The substances are silver, gold, and lead. Everything has a substance. A thing is usually lead.

Definition: a thing is valuable if it is not lead.

The Limestone Cave is a room. "Not very big, and it doesn't go back far, but you'd hoped to find some shelter here. Outside it is raining exceptionally hard."

The sinister gentleman is a man in the Cave. "Leaning against the wall is a sinister gentleman in a threadbare waistcoat." The description is "He looks as though he might once have been quite well off."

After examining the gentleman:

say "[The noun] smiles back at you in an unnerving fashion."

Now we make the rule that governs the gentleman's behavior. Here we're going to invoke the rules that allow characters besides the player to do actions. More about this can be found in the Advanced Actions chapter:

Every turn:

if the sinister gentleman can touch something valuable (called the treasure)

which is not carried by a person:
try the gentleman taking the treasure.

Report the gentleman taking something:
say "[The gentleman] slyly acquires [the noun] and tucks it into his pocket."
instead.

That "not carried by a person" prevents the gentleman from stealing from the player (or, less plausibly, from himself). If we did want him to pick pockets, we could just have said "which is not carried by the gentleman".

The player is carrying a coin, a bust of Abraham Lincoln, a bottle of sherry, a small pistol, and a wad of Confederate cash. The coin is gold. The pistol is silver.

A locket is in the Cave. The locket is gold. "A broken gold locket lies on the ground, a token of your late Mama."

Test me with "get locket / i / drop locket / z / drop cash / z / x gentleman / drop all / z / z / look".

140



Example Text Foosball

WI

A game of foosball which relies heavily on every-turn rules.

Suppose we want a game of foosball in which our opponent acts every turn, but does different things depending on where the ball currently lies. We can put together a sequence of every-turn rules to account for this, as follows:

"Text Foosball"

Use scoring.

The Lounge is a room. "The Lounge is appointed with everything necessary to rest and relaxation: a vending machine, a potted palm, a stack of Entertainment Weekly issues from 1993, and -- your pride and joy -- a foosball game."

The foosball game is scenery in the Lounge. Understand "table" or "football" or "foozball" or "fussball" or "soccer" as the foosball game. The game is a supporter. On the game is a small white ball. The ball can be still, approaching, receding, or unreachable. The description of the ball is "Currently [small white ball condition]."

After printing the name of the small white ball, say " ([small white ball condition])".

When play begins:

now left hand status line is "You: [score]";
now right hand status line is "Joey: [Joey's score]".

Some tiny men on sticks are part of the game. Understand "handles" as the tiny men. The description is "Okay, a couple of the tiny men have had their feet

broken off, and the table surface itself is a bit warped, and the ball resembles a quail egg in respect of shape and color. This makes for a game of unusual randomness, but skill is overrated."

Instead of attacking or pulling or pushing the game when the ball is unreachable:
say "You give the table a good shove, and the ball moves ever-so-slightly";
now the ball is still.

Instead of taking the white ball:
say "You'd forfeit the game if you did that."

Instead of turning the tiny men when the ball is unreachable:
say "The ball has somehow gotten to a mystical point on the table where it cannot be reached, no matter what. Close inspection reveals that this point has been marked in chalk with a tiny X. Not that that does any good."

Instead of turning the tiny men when the ball is approaching:
if a random chance of 2 in 3 succeeds:
if a random chance of 1 in 2 succeeds, now the ball is receding;
otherwise now the ball is still;
say "[if the ball is still]Thunk. [otherwise]Thwack! [end if]You keep the ball from reaching its goal! Now it is [small white ball condition].";
otherwise:
let Joey score.

To let Joey score:
now the ball is still;
now Joey's score is Joey's score + 1;
say "The ball rolls neatly into your goal, despite your efforts. ";
if Joey's score < score, say "You put the ball back in the center with a snap.
No reason to worry yet; you're still ahead. Joey looks determined, though.";
otherwise say "After allowing a moment or two for Joey's gloating to pass, you replace it at the center."

Instead of turning the tiny men when a random chance of 1 in 13 succeeds:
if the ball is unreachable, continue the action;
now the ball is unreachable;
say "You hit the ball off-center and it rolls sluggishly into a little dip in the surface of the foosball table. ";
if Joey's score > 7, say ""You did that on purpose! Joey exclaims indignantly.";
otherwise say "You and Joey exchange glances. This is never good."

Instead of turning the tiny men:
say "You madly rotate the tiny men on sticks! ";
if a random chance of 1 in 2 succeeds:
say "Hoorah! You hit the ball!";
now the ball is receding;
otherwise:
say "Somehow you fail to bring your monopodal player into contact with the ball."

Joey is a man in the Lounge. "Joey is hunkered over the foosball handles on his side of the table." Joey can be active or inactive.

Joey's score is a number that varies.

Every turn: now Joey is active.

Every turn when the ball is approaching and Joey is active:

- let total be Joey's score + score;
- if total > 9, make no decision;
- now Joey is inactive;
- let Joey score;
- rule succeeds.

Every turn when the ball is unreachable and Joey is active:

- let total be Joey's score + score;
- if total > 9, make no decision;
- now Joey is inactive;
- say "Joey glares angrily at the stuck ball."

Every turn when the ball is receding and Joey is active:

- let total be Joey's score + score;
- if total > 9, make no decision;
- if the ball is unreachable, make no decision;
- now Joey is inactive;
- if a random chance of 1 in 2 succeeds:
 - if a random chance of 1 in 2 succeeds, now the ball is still;
 - otherwise now the ball is approaching;
 - say "Joey connects with your shot. Now the ball is [small white ball condition]!";
- otherwise:
 - now the ball is still;
 - say "Joey tries to block, but misses! Back it goes in the center, where it is [small white ball condition].";
 - increment the score.

Every turn when the ball is still and Joey is active:

- let total be Joey's score + score;
- if total > 9, make no decision;
- if the ball is unreachable, make no decision;
- now Joey is inactive;
- if a random chance of 1 in 2 succeeds:
 - now the ball is approaching;
 - say "Joey hits the ball solidly down towards your goal. Now it is [small white ball condition].";
- otherwise:
 - say "Joey fails to hit the ball in your direction. It remains [small white ball condition]."

Every turn:

- let total be Joey's score + score;
- if total > 9:
 - if Joey's score > score, end the story saying "Rats! Joey wins!";
 - if Joey's score < score, end the story finally saying "Victory is yours!";
 - if Joey's score is score, end the story saying "A perfect tie."

different degrees of restriction.

Notice that the following two scenarios do not have the same effect:

"Zodiac"

The Secluded Alley is a room. The Capricorn Killer is a man in the Secluded Alley.

The player carries a can of mace and a roll of duct tape.

Instead of doing something other than examining with the Capricorn Killer:
say "You wouldn't dare!"

Test me with "x me / x killer / touch killer / smell mace".

...will prevent the player from doing anything else to the killer, but allow him free range of action with other objects. By contrast, the following will prevent him doing anything other than examining to any item:

"Zodiac"

The Secluded Alley is a room. The Capricorn Killer is a man in the Secluded Alley.

The player carries a can of mace and a roll of duct tape.

Instead of doing something other than examining something in the presence of the Capricorn Killer:
say "You dare not attempt it!"

Test me with "x me / x killer / touch killer / smell mace".

But notice that because we specified "examining something", actions that take no object (like sing) are still not affected.

The most strict statement would be

"Zodiac"

The Secluded Alley is a room. The Capricorn Killer is a man in the Secluded Alley.

The player carries a can of mace and a roll of duct tape.

Instead of doing something other than looking or examining in the presence of the Capricorn Killer:
say "You dare not attempt it!"

Test me with "x me / x killer / touch killer / smell mace".

Because we left "something" out, now the pattern does not have to match an activity with an object; it will match any activity at all. Singing too will be deemed too risky in the presence of our sinister foe. But we do need to make an exception for "look", or else we won't get even the room description.

154

★ Example Pine 1

WI

Pine: Using a scene to watch for the solution of a puzzle, however arrived-at by the player.

Because scene rules are checked every turn, they can be useful for designing puzzles which have multiple solutions. Instead of deciding the puzzle is "solved" when the player does a certain action, we set up a scene that checks to see whether the player has achieved a certain outcome -- however he accomplished it.

For instance, in this scenario, we're waiting for Sleeping Beauty to wake up, and it doesn't much matter how...

"Pine"

A person can be asleep or awake. A person can be active or passive.

The Spinning Tower is a room. "A remote corner of the old castle, reserved for spinning and weaving tasks."

Sleeping Beauty is an asleep woman in the Spinning Tower. "[if asleep]Sleeping Beauty lies here, oblivious to your presence[otherwise]Sleeping Beauty stands beside you, looking a little confused[end if]." The description is "She is even more magnificent than the rumors suggested." Understand "woman" or "girl" or "princess" or "lady" as Sleeping Beauty.

Discovery is a scene. Discovery begins when play begins. Discovery ends when Sleeping Beauty is awake. Marriage Proposal is a scene. Marriage Proposal begins when Discovery ends.

When Discovery ends: say "Throughout the palace you can hear the other sounds of stirring and movement as the spell of centuries is broken."

Instead of waking an awake person: say "Redundant."

Instead of waking an asleep person: say "Yes, but how?"

Instead of attacking an asleep person:

now the noun is awake;
say "[The noun] sits bolt upright. 'Hey! Ow!' So much for that true love's kiss nonsense."

Instead of kissing an asleep person:

now the noun is awake;
say "[The noun] slowly stirs to wakefulness!"

Instead of throwing water at an asleep person:
now the second noun is awake;
now the noun is nowhere;
say "You pour out [the noun] on [the second noun]."

[The second noun] wakes, shuddering. 'Agh! I had a terrible dream about drowning and then-- Hey!'"

The player carries a jug of water. Understand "pour [something] on [something]" or "splash [something] at/on [something]" as throwing it at.

Test me with "x beauty / wake beauty / pour water on beauty".

187

★ Example Police State

WI

Several friends who obey you; a policeman who doesn't (but who takes a dim view of certain kinds of antics).

"Police State"

Seventh Avenue is a room. "The bars are all closed now, and there aren't any good clubs to go to, so you're on your own for open-air entertainment."

Persuasion rule for asking the policeman to try doing something: persuasion fails.

Persuasion rule for asking someone to try doing something: persuasion succeeds.

Note that the policeman will never get to the second persuasion rule, so he will always refuse to do the player's nefarious bidding.

Charles, Thomas, and Larry are men in Seventh Avenue. Patricia is a woman in Seventh Avenue.

And here's an unnecessary aesthetic touch from a later chapter, which will round up the descriptions of your friends into a single paragraph:

Rule for writing a paragraph about someone who is not the policeman:
let X be the number of visible people who are not the policeman;
say "It's just [X in words] of you now: [a list of visible people who are not the policeman]. But it sure has been a rip-roaring evening."

The policeman is a man in Seventh Avenue. "A policeman with a very guarded expression is watching you."

Singing is an action applying to nothing. Understand "sing" as singing.

Report singing:
say "A little the worse for wear, you sing."

Smelling a person is disorderly conduct. Tasting a person is disorderly conduct. Jumping is disorderly conduct. Singing is disorderly conduct.

Instead of someone trying disorderly conduct in the presence of the policeman:
say "The policeman arrests [the person asked]!";
now the person asked is nowhere;
the rule succeeds.

Instead of disorderly conduct in the presence of the policeman:
end the story saying "The policeman arrests you!"

Test me with "charles, look / charles, jump / look / policeman, sing / thomas, taste policeman / patricia, sing / look / jump".

Notice the difference between the two rules about disorderly conduct: the one for other people says 'the rule succeeds' to make sure that the action is counted as a success and not (as normally happens with instead rules) a failure. Most of the time we don't care whether actions are judged successes or failures, but it matters here, because if we type CHARLES, JUMP and the result fails, then text such as 'Charles is unable to.' will be printed - which would get in the way. So we declare the action a success.

38



Example Search and Seizure

WI

A smuggler who has items, some of which are hidden.

Suppose we want a character who carries hidden objects, but only while he is wearing his jacket. If we deprive him of this, his other possessions become known. Furthermore, if we ever search him, his possessions also become known, and can thereafter be mentioned by us.

"Search and Seizure"

Size is a kind of value. The sizes are small, medium, and large. A thing has a size. A thing is usually small.

A thing can be licit or contraband. A thing is usually licit.

A thing can be discovered or secret. A thing is usually secret.

Now, we want the character to be able to hide small things if he has some sort of concealing garment on. We also want to be able to see anything that the player has already found once, perhaps by using the >SEARCH PERSON command. So:

Rule for deciding the concealed possessions of someone (called the suspect):
if the particular possession is discovered, no;
if the suspect wears something and the particular possession is small, yes;
otherwise no.

The following rule, borrowed from a later chapter, assures that any items that are ever mentioned to the player will be treated as known from then on:

Before printing the name of something (called discovery): now the discovery is discovered.

The Customs House is a room. The smuggler is a man in the Customs House. The smuggler wears a leather jacket. He carries a bottle of Altairan rum, some raspberries, and a laser pistol. The pistol is large. The jacket is large. The rum and the raspberries are contraband. The description of the smuggler is "He has a bestubbed chin and a sinister eye."

The tourist is a woman in the customs house. The description is "The type who walks off home with a dozen contraband items in her pocket not because she means to steal things but because she's too stupid to understand that the law applies to herself." She wears a tight-fitting dress. The dress is large. The tourist carries a grapevine and an archaeological artifact. The grapevine is large and contraband. The artifact is contraband.

Report examining someone: say "[The noun] is [if the noun is wearing something]wearing [a list of unconcealed things worn by the noun] and [end if]carrying [a list of unconcealed things carried by the noun]."

Report examining someone who is concealing something contraband:
say "[The noun] looks nervous. You can just tell."

Report examining someone who is carrying an unconcealed contraband thing:
say "Your eye goes at once to [the list of unconcealed contraband things carried by the noun] which [the noun] is unable to hide";
if an unconcealed licit thing is had by the noun, say ", though [if the noun is female]s[end if]he also has [a list of unconcealed licit things had by the noun]";
say "." instead.

Notice that we can talk about what the smuggler wears, what he carries, and what he "has": things the smuggler has can be either worn or carried, so the phrase is useful if we don't care to make that distinction.

Instead of searching someone:
say "[The noun] is revealed to be carrying [a list of things carried by the noun]."

Instead of confiscating the dress:
say "You are not allowed to perform strip-searches in the public customs area."

Understand "confiscate [something]" as confiscating.

Confiscating is an action applying to one thing.

Check confiscating:
unless the noun is had by someone who is not the player:
say "You can only confiscate other people's possessions." instead.

Carry out confiscating:
now the noun is carried by the player.

Report confiscating:
say "Through the authority vested in you by the power of the Sovereign of Centauri Proxima, you make [the noun] your own."

Test me with "x smuggler / search smuggler / x smuggler / confiscate jacket / x smuggler / confiscate rum / confiscate pistol / x smuggler / confiscate raspberries / x smuggler".

Test more with "x tourist / confiscate dress / confiscate grapevine / x tourist / search tourist / confiscate artifact / x tourist".

197

☆☆☆ Example Noisemaking

WI

Creating a stage after the report stage of an action, during which other characters may observe and react.

Suppose the current sequence of action handling is not quite enough for us: we'd also like to have a stage after reporting, where other characters can react to the player character's behavior after it has already happened and been reported on screen. Having such a stage is unlike using "after", because after occurs before reports and prevents them from being printed. So, for instance, we could allow the player to do any of a range of different actions that make loud noises, and have a nervous bird that reacts to all of them by flying away afterward.

To do this, we can add a new rule into the specific action-processing rules. (For a list of these, see the Rules index.) Moving rules around and adding new ones requires syntax that we will learn in the chapter on Rulebooks, but the present example is fairly straightforward:

"Noisemaking"

Section 1 - Procedure

The other-player response rule is listed after the report stage rule in the specific action-processing rules.

This is the other-player response rule:
follow the observation rules.

The observation rules is a rulebook.

Section 2 - Scenario

Country Lane is a room. West of Country Lane is Outside the Farmhouse. East of Country Lane is Village Center. North of Country Lane is Open Field.

The player carries a drum.

The black crow is an animal in Country Lane.

Singing is an action applying to nothing. Understand "sing" as singing.

Report singing:
say "You hum a little ditty."

Singing is a loud action.

Attacking the drum is a loud action.

The block attacking rule is not listed in any rulebook.

Report attacking something:
say "THWACK!"

An observation rule for loud action in the presence of the black crow:
let N be a random adjacent room;
if N is a room, move the black crow to N;
say "The crow, startled, flies off to [N]."

Test me with "sing / g / n / hit drum".

82



Example **Revenge of the Fussy Table**

WI

A small game about resentful furniture and inconvenient objects.

This example does include a number of features that we haven't met yet, particularly rulebooks. Nonetheless, the basic idea should be relatively clear. Our premise is that the player is faced with a series of disgruntled furnishings, none of which want to be responsible for supporting the cold, hot, drippy, or spiky objects lying around the room.

"Revenge of the Fussy Table"

Use scoring.

Section 1 - The Supporters

The Dining Room is a room. "A room of handsome proportions, with an octagonal dining table on a fine parquet floor. The original set of chairs has been taken away, leaving only a red velvet dining chair and a blue suede armchair."

The fussy table is a supporter in the Dining Room. It is scenery. The description is "It's hypochondriac: the result of a sojourn in a superior antique shop, where it picked up the scent of almond oil, words like provenance, and a secretive shame about being resurfaced."

Chairs don't get along with it.

In its mellowest moods, after a really good dusting, it will mention, casually, a cabinet it once knew, which was twelve feet tall, came from the Winter Palace, and had an asking price of \$350K."

A chair is a kind of supporter with carrying capacity 1. A chair is enterable. A chair is usually scenery. Understand "chair" as a chair.

The blue suede armchair and the red velvet dining chair are chairs in the Dining Room. The description of the blue suede is "Quiet, mellow, and with a lingering aroma of clove cigarettes, and possibly something less legal. It doesn't sit up straight, and doesn't let anyone else do so either." The red velvet has the description "In temperament, in bearing, in everything it is the opposite of the blue suede chair. It is concerned for its duty all the time, has a rigidly straight back, and pokes diners in the spine when it suspects them of overdoing things."

The carrying capacity of the player is 2.

Section 2 - Sources of Unpleasantness

A thing can be drippy or dry.

Heat is a kind of value. The heats are luke-warm, cold, and scalding. Everything has a heat. Understand the heat property as describing a thing.

A thing can be spiky or smooth. The cleated left shoe is a wearable spiky thing. It is worn by the player. The cleated right shoe is a wearable spiky thing. It is worn.

Definition: a thing is unpleasant if it is scalding or it is cold or it is spiky or it is drippy.

Definition: a supporter is occupied if something is on it.

Definition: something is contented if it is not concerned.

Definition: a chair is concerned if something unpleasant is on it.

Definition: the table is concerned if something drippy is on it or something scalding is on it or something spiky is on it.

Definition: the ceramic tile is concerned if something spiky is on it.

Definition: a room is concerned if something drippy is in it or something scalding is in it or something spiky is in it or someone which is in it is wearing a spiky thing.

Before printing the name of a drippy thing: say "drippy ". Before printing the name of a cold thing: say "cold ". Before printing the name of a scalding thing: say "scalding ".

Section 3 - What the Player Can Do About It

After putting a cold thing on a scalding thing:

say "[The noun] meets [the second noun]; both shriek in pain. But the

necessary heat transfer occurs.";
now the heat of the noun is luke-warm;
now the heat of the second noun is luke-warm.

After putting a scalding thing on a cold thing:
say "[The noun] meets [the second noun]; both shriek in pain. But the necessary heat transfer occurs.";
now the heat of the noun is luke-warm;
now the heat of the second noun is luke-warm.

After taking a scalding thing:
say "'Right,' you say. 'I'll just hold onto [the noun], shall I? Because that won't be in the least uncomfortable for me.'"

After taking a drippy thing:
say "With a sense of martyrdom, you retrieve [the noun], which dribbles water down your cuffs."

After entering a chair when the player is wearing a spiky thing (called the clog):
say "You sit on [the noun], lifting [the list of spiky things worn by the player] so that the parquet is no longer affronted."

Section 4 - In Which the Furnishings Complain

An every turn rule:
follow the complaint rules.

The complaint rules is a rulebook.

A complaint rule:
if something (called the offending item) on the table is drippy:
say "'Help! Get me a coaster!' screams the table[if the table is visible], its veneer squirming under [the offending item][otherwise] from the Dining Room[end if].";
rule succeeds;
if something (called the offending item) on the red chair is drippy:
say "'Oh dear,' murmurs the red chair, as [the offending item] drips into its velvety seat. 'Oh dear, I will have a damp spot. This is so very -- what will people think?'"";
rule succeeds;
if something (called the offending item) on the visible armchair is drippy:
say "[The offending item] visibly begins degrading the suede where it sits. The armchair is tactfully silent.";
rule succeeds;
if a drippy thing (called the offending item) is in the location and the player is in the Dining Room:
say "'Cripes,' says the parquet. 'No one mind me at all. Just leave that [offending item] right here. You know I'm the most valuable thing in the room?'"";
rule succeeds.

A complaint rule:
if a cold dry thing (called the offending item) is on the table:
say "The table shivers under [the offending item].";
rule succeeds.

A complaint rule:

if something (called the offending item) on the table is scalding:
say "'Hey!' protests the table, practically smoking at [the offending item].
'Heard of a trivet?";
rule succeeds;
if something (called the offending item) on the armchair is scalding:
say "The armchair doesn't complain about [the offending item]; it just
begins, quietly, to give off warm sweet clouds of blue steam, as though its inner
opium-nature had been released.";
rule succeeds;
if something (called the offending item) in the Dining Room is scalding:
say "'Yes indeed,' [if the location is the Dining
Room]says[otherwise]bellows[end if] the parquet. 'That [offending item] won't
leave any sort of mark what-so-ever.'";
rule succeeds.

A complaint rule:

if something (called the offending item) on the table is spiky:
say "The table holds very very very still lest [the offending item] mar its
finish. But its resentment is palpable.";
rule succeeds;
if something (called the offending item) on the velvet chair is spiky:
say "[The offending item] stabs the velvet of the red velvet chair, which
draws itself up even more sharply in a pose of dutiful martyrdom.";
rule succeeds;
if something (called the offending item) which is spiky is on the ceramic tile:
say "'[The offending item] itches!' cries the ceramic tile.";
rule succeeds;
if something (called the offending item) in the location is spiky:
say "'[The offending item] should not be on a parquet floor,' sings the
parquet floor sadly.";
rule succeeds;
if someone (called the offending person) in the location is wearing something
spiky and the location is the Dining Room:
say "The parquet floor mutters about the things people wear to walk on
parquet floors these days.";
rule succeeds.

A complaint rule:

if the table supports something which supports a drippy thing (called the
offending item),
say "'[The offending item] could still theoretically dribble down here,' says
the table nonchalantly. 'It's possible.'";
if the velvet chair supports a concerned thing (called the friend),
say "'[The friend] seems a little worried,' remarks the velvet chair helpfully."

A complaint rule:

if the concerned table can see a supporter (called the object of envy) which is
not concerned:
say "The table looks jealously at [the object of envy]. 'Some people have
such an easy life.'";
rule succeeds.

Instead of going to an adjacent room when the player is in the Dining Room:
say "'You're not leaving!?' whimpers the table at once."

Moreover, every time the player gets close to resolving this issue, his unhelpful companion Alison brings in something else inconvenient. We can use the counting of contented supporters to decide when the player is getting close to winning and it's time for her to bring something else...

Section 5 - Source of Further Complications

The Tidy Kitchen is south of the Dining Room. Alison is a woman in the Tidy Kitchen. The kettle is a scalding thing. The ceramic tile is a cold portable supporter. It has carrying capacity 1. The platter is a scalding portable supporter. It has carrying capacity 1. The glass of ice water is a cold, drippy thing.

Instead of drinking the glass of ice water, say "It's not yours, it's for Alison, once she finishes cooking."

Alison carries the kettle, the tile, the platter, and the glass.

An every turn rule:
follow the behavior rules.

The behavior rules is a rulebook.

A behavior rule:
if ice water is carried by Alison and the player is in the Dining Room and almost all of the supporters are not concerned:
let the next victim be a random fixed in place contented supporter;
move the ice water to the next victim;
say "Humming to herself, Alison brings in [an ice water] and sets it down smack on [the next victim], then goes back out.";
rule succeeds.

A behavior rule:
if platter is carried by Alison and the player is in the Dining Room and almost all of the supporters are not concerned:
let the next victim be the red velvet chair;
move the platter to the next victim;
say "Whistling a jaunty tune, Alison brings in a hot platter and deposits it on [the next victim], then returns to the kitchen.";
rule succeeds.

A behavior rule:
if the tile is carried by Alison and the player is in the Dining Room and almost all of the supporters are not concerned:
let the next victim be a random contented chair;
move the tile to the next victim;
say "Humming to herself, Alison brings in [the tile] and sets it down smack on [the next victim], then goes back out.";
rule succeeds.

A behavior rule:
if kettle is carried by Alison and the player is in the Dining Room and at least three fixed in place supporters are contented:
let the next victim be the fussy table;
move the kettle to the next victim;
say "Just then the kettle boils in the kitchen. Whistling chirpily, Alison brings

it in and sets it down smack on [the next victim], then goes back out."; rule succeeds.

And just so that the player knows where he stands at the end of each turn:

Section 6 - General Assessment

An every turn rule:

if some of the things are concerned, say "You sense some resentment from [the list of concerned things]."

An every turn rule:

if the player is carrying more than one scalding thing:
let the dropped item be a random scalding thing carried by the player;
say "You can't hang onto so many burning hot things, and lose your grip on [the dropped item].[line break]";
silently try dropping the dropped item.

An every turn rule:

if almost all of the supporters are concerned,
say "'I hope everything is going well,' bellows Alison from the other room."

Finally, we need some rules to decide when the player is, in fact, finished.

Section 7 - Scoring

An every turn rule:

now score is 5 minus the number of concerned things;
if the location is concerned, decrement the score;
if all of the supporters are concerned and the location is concerned, end the story;
if none of the supporters are concerned and the location is not concerned, end the story finally.

The maximum score is 5.

When play begins, now score is 4.

Test me with "i / take off left shoe / get the glass / put the glass on the platter / get the tile / drop the tile / get the kettle / put the kettle on tile / sit on the blue armchair".

112



Example A Day For Fresh Sushi

A complete story by Emily Short, called "A Day for Fresh Sushi", rewritten using Inform 7. Noteworthy is the snarky commenter who remarks on everything the player does, but only the first time each action is performed.

WI

The following is an almost-completely-faithful rewrite of Emily Short's "A Day for Fresh Sushi", which was originally written using the (very different) Inform 6

programming language. The -- let us be honest and call it a gimmick -- of this game is the evil fish, who has some unpleasant remark to offer on pretty much every action. But the effect would wear off fast if he repeated himself, so these comments need to be single-use only.

Inform 7's repeated action syntax makes it much tidier to write the same scenario, so:

"A Day For Fresh Sushi" by Emily Short.

Use scoring.

The story headline is "Your basic surreal gay fish romance".

The Studio is a room. "[if visited]Decorated with Britney's signature flair. It was her innate sense of style that first made you forgive her that ludicrous name. And here it is displayed to the fullest: deep-hued drapes on the walls, the windows flung open with their stunning view of old Vienna, the faint smell of coffee that clings to everything. Her easel stands over by the windows, where the light is brightest.[otherwise]This is Britney's studio. You haven't been around here for a while, because of how busy you've been with work, and she's made a few changes -- the aquarium in the corner, for instance. But it still brings back a certain emotional sweetness from the days when you had just met for the first time... when you used to spend hours on the sofa...

You shake your head. No time for fantasy. Must feed fish.[end if]"

Instead of smelling the Studio:

say "The evil fish notices you sniffing the air. 'Vanilla Raspberry Roast,' it remarks. 'You really miss her, don't you.'

You glance over, startled, but the fish's mouth is open in a piscine equivalent of a laugh. You stifle the urge to skewer the thing..."

Instead of jumping:

say "'Er,' says the fish. 'Does that, like, EVER help??'"

Instead of going nowhere:

say "You can't leave until you've fed the fish. Otherwise, he'll complain, and you will never hear the end of it."

The cabinet is an openable closed container in the Studio. It is fixed in place. "A huge cabinet, in the guise of an armoire, stands between the windows." The description is "Large, and with a bit of an Art Nouveau theme going on in the shape of the doors." Understand "armoire" as the cabinet.

Instead of looking under the cabinet for the first time:

say "'Dustbunnies,' predicts the fish, with telling accuracy. It executes what for all the world looks like a fishy shudder. 'Lemme tell you, one time I accidentally flopped outta the tank, and I was TWO HOURS on the floor with those things STARING ME IN THE NOSE. It was frightening.'"

After opening the cabinet for the first time:

say "'There ya go,' says the fish. 'The girl is getting WARMER.'"

After closing the cabinet for the first time:

if the fish food is not found, say "'Ooh, what do you think, Bob? I think we're going to have to dock the girl a few points. HAVE ANOTHER LOOK, sweetcakes, there's a doll.'"

The cabinet contains some paints and some cloths. The description of the paints is "A bunch of tubes of oil paint, most of them in some state of grunginess, some with the tops twisted partway off."

After taking the paints for the first time:

say "'Boy,' says the fish, apparently to himself, 'I sure hope that's some food she's finding for me in there. You know, the yummy food in the ORANGE CAN.'"

After examining the paints for the first time:

say "'Tons of useful stuff in there,' hollers in the fish, in a syncopated burble."

The description of the cloths is "Various colors of drapery that Britney uses to set up backgrounds and clothe her models. She does a lot of portraiture, so this comes in handy. It's all a big messy wad at the moment. Organized is not her middle name." Understand "drapery" or "cloth" as the cloths. The indefinite article of the cloths is "a heap of". [see 3.17]

Instead of searching or looking under the cloths for the first time:

now the player is carrying the fish food;
now the fish food is found;
say "Poking around the cloths reveals -- ha HA! -- a vehemently orange can of fish food."

Instead of showing the cloths to the fish:

say "'What are you, some kind of sadist? I don't want to see a bunch of cloths! What kind of f'ing good, 'scuse my French, is that supposed to do me? I don't even wear pants for God's sake!'

He really looks upset. You start wondering whether apoplexy is an ailment common to fish."

After examining cloths for the first time:

say "'Whatcha looking at? I can't see through the doors, you know.'"

There is a can of fish food. Understand "canister" as the can. The description is "A vehemently orange canister of fish food." The fish food can be found or hidden. The fish food is hidden.

Instead of giving the can to the fish:

say "'I don't want the whole can, GeniusChyk. Just feed me and we'll ALL be happy, 'kay?'"

Instead of showing the can to the fish:

say "'That's the ticket, sweetie! Bring it on.'"

Instead of opening the can:

say "'Oh, for--!' The evil fish breaks out in exasperation and hives. 'Screw the screwing around with the screwtop. SHE never has to do that.'

'Well, SHE is not here,' you reply. 'What do you suggest?'

'>FEED FISH<' says the fish promptly, making fishy faces and pointing at you with his fin. 'Simplicity. Try it.'

Instead of inserting the can into something:

say "'HeLLLLLOOO,' screams the fish. 'Whatever happened to FEEDING MEEE?'"

The easel is a supporter in the Studio. It is scenery. On the easel is a painting. Understand "portrait" or "image" as the painting.

The description of the painting is "Only partway finished, but you can tell what it is: Britney's mother. You only met the old woman once, before she faded out of existence in a little hospice in Salzburg.

In the picture, her hands are grasping tightly at a small grey bottle, the pills to which she became addicted in her old age, and strange, gargoyle-like forms clutch at her arms and whisper in her ears.

But the disturbing thing, the truly awful thing, is the small figure of Britney herself, down in the corner, unmistakable: she is walking away. Her back turned.

You thought she'd finally talked this out, but evidently not. Still feels guilty for leaving. You only barely stop yourself from tracing, with your finger, those tiny slumped shoulders..."

Instead of taking the painting, say "No, you'd better leave it. It'd freak her out if you moved it."

Before examining the painting for the first time:

say "A ferocious banging from the aquarium attracts your attention as you go to look at the painting. 'Hey!' screams the fish. 'She doesn't like strangers looking at her paintings before they're DOONNNE!'"

'Shut up, you,' you reply casually. 'I'm not a stranger.' But the fish puts you off a little bit, and your heart is already in your mouth before you see the painting itself..."

Instead of examining the painting more than once:

say "Once is really enough. It's pretty much embedded in your consciousness now."

After doing something to the painting:

say "'So what's it of?' asks the fish, as you turn away. 'She never asks if I want to see them, you know?'"

'Her mother,' you respond without thinking.

'Yeah? Man. I never knew my mother. Eggs, that's the way to go.'

The window is scenery in the Studio. The window can be openable. The window can be open. It is openable and closed. Understand "windows" as the window. The description of the window is "[if open]Through the windows you get a lovely view of the street outside. At the moment, the glass is thrown open, and a light breeze is blowing through.[otherwise]Through the windows, you get a lovely

view of the street outside -- the little fountain on the corner, the slightly dilapidated but nonetheless magnificent Jugendstil architecture of the facing building. The glass itself is shut, however.[end if]"

After opening the window for the first time:

say ""Thank god some air," says the fish. 'Man, it was getting hard to breathe in here.' Two beats pass. 'Oh wait.'"

The table is scenery in the Studio. On the table is a vase. The vase is an open container. It is not openable.

The description of the table is "A monstrosity of poor taste and bad design: made of some heavy, French-empire sort of wood, with a single pillar for a central leg, carved in the image of Poseidon surrounded by nymphs. It's all scaley, and whenever you sit down, the trident has a tendency to stab you in the knee. But Britney assures you it's worth a fortune." The description of the vase is "A huge vase -- what you saw once described in a Regency romance as an epergne, maybe -- something so big that it would block someone sitting at the table from seeing anyone else also sitting at the table. But it does function nicely as a receptacle for hugeass bouquets of flowers."

Instead of looking under the table for the first time:

say ""You're not going to find anything down there," whines the fish. 'I mean, c'mon. It's the fricking floor. Please tell me you can see that. I can see that. I'm a myopic fish in a tank ten feet away and I can tell you there is nothing there but floor.'"

After examining the table:

say ""That there is MY PA," says the fish, pointing at the scaley triton figure with one fin."

Instead of inserting something which is not the bouquet into the vase:

say ""Okay, so, what were you, raised in a barn? Normal folks like to use that for flowers. Or so I've observed.'"

After inserting the bouquet into the vase for the first time:

say "You settle the flowers into the vase and arrange them so that they look sprightly.

'Oooh,' says the fish. 'No one ever changes the plant life in HERE. It's the same seaw--'

'Cut me a break and cork it,' you reply tartly."

The player is carrying a telegram, a bouquet, and a lingerie bag. The player is wearing a chef hat.

The description of the telegram is "A telegram, apparently. And dated three days ago. [fixed letter spacing]TRIUMPH OURS STOP BACK SOON STOP BE SURE TO FEED FISH STOP[variable letter spacing]". [For printing options see 4.13.] Understand "yellow paper" as the telegram.

After examining the telegram for the first time:

say ""So," blubs the evil fish. 'How about it? Little food over here?'"

After examining the telegram:

choose a random row in the Table of Insulting Fish Comments;
say "[comment entry][paragraph break]".

Table of Insulting Fish Comments

comment

"'Yeah, yeah,' says the fish. 'You having some trouble with the message, there? Confused? Something I could clear up for you?'"

"'Oookay, genius kid has some troubles in the reading comprehension department.' The fish taps his head meaningfully against the side of the tank. 'I'm so hungry I could eat my way out, you get my meaning?'"

"'I'll translate for you,' screams the fish in toothy fury. 'It says GIVE FOOD TO FISH!! How much more HELP do you NEED???'"

The description of the chef hat is "A big white chef hat of the kind worn by chefs. In this case, you. Just goes to show what a hurry you were in on the way out of the restaurant." Understand "big" or "white" or "chefs" or "chef's" as the chef hat. [Inform knows that this is clothing because the player starts out wearing it, so there's no need to say so separately.]

The aquarium is a transparent open container in the Studio. It is not openable. "In one corner of the room, a large aquarium bubbles in menacing fashion." The description of the aquarium is "A very roomy aquarium, large enough to hold quite a variety of colorful sealife -- if any yet survived." Understand "tank" as the aquarium.

The aquarium contains some gravel and some seaweed. Understand "little rocks" as the gravel. Understand "weed" as the seaweed. The description of the gravel is "A lot of very small grey rocks." The description of the seaweed is "Fake plastic seaweed of the kind generally bought in stores for exactly this purpose."

The examine containers rule does nothing when examining the aquarium.

After examining the gravel for the first time:

say "The fish notices your gaze; makes a pathetic mime of trying to find little flakes of remaining food amongst the gravel."

After examining the seaweed for the first time:

say "'Nice, hunh?' blubs the fish, taking a stabbing bite out of one just by way of demonstration. 'Look so good I could eat it.'"

The aquarium contains an animal called an evil fish. The description of the fish is "Even if you had had no prior experience with him, you would be able to see at a glance that this is an evil fish. From his sharkish nose to his razor fins, every inch of his compact body exudes hatred and danger."

Instead of taking the evil fish:

say "The fish swims adroitly out of range of your bare hand. 'Hey,' he says, and the bubbles of his breath brush against your fingers. 'Count yourself lucky I don't bite you right now, you stinking mammal.'"

Instead of attacking the evil fish:

say "Oh, it's tempting. But it would get you in a world of hurt later on."

Instead of kissing the evil fish:

say "You're saving all your lovin for someone a lot cuddlier."

After examining the evil fish for the first time:

say "The fish glares at you, as though to underline this point."

After examining the evil fish for the second time:

say "'If you're looking for signs of malnutrition,' says the fish, 'LOOK NO FURTHER!!' And it sucks in its gills until you can see its ribcage."

An every turn rule:

choose a random row in the Table of Fish Banter;
say "[comment entry][paragraph break]".

Table of Fish Banter

comment	used
"'Hey, nice SKIN TONE,' shouts the evil fish. His words reach you in a spitting gurgle of aquarium water. 'You gone over to a pure eggplant diet these days?'"	0
"The evil fish is floating belly up! ...oh, curse. He was toying with you. As soon as he sees you looking, he goes back to swimming around."	0
"The evil fish darts to the bottom of the tank and moves the gravel around with his nose."	0
"The evil fish is swimming around the tank in lazy circles."	0
"The evil fish begins to butt his pointy nose against the glass walls of the tank."	0

The description of the bouquet is "Okay, so it's silly and sentimental and no doubt a waste of money, of which there is never really enough, but: you miss her. You've missed her since ten seconds after she stepped aboard the shuttle to Luna Prime, and when you saw these -- her favorites, pure golden tulips like springtime -- you had to have them." Understand "flowers" or "tulip" or "tulips" as the bouquet.

After examining the bouquet for the first time:

say "'Oh, you shouldn't have,' says the fish. 'For me?'"
You just respond with a livid glare."

Instead of smelling the bouquet for the first time:

say "'Mmm-mm,' says the fish. 'Damn, I sure wish I had olfactory abilities. Hey, if I did, I might be even better at noticing the presence or absence of FOOD.'"

The description of the lingerie bag is "You grant yourself the satisfaction of a little peek inside. You went with a pale, silky ivory this time -- it has that kind of sophisticated innocence, and it goes well with the purple of your skin. A small smirk of anticipation crosses your lips."

After examining the lingerie bag for the first time:

say "'What's in THERE?' asks the fish. 'Didja bring me take-out? I don't mind Chinese. They eat a lot of carp, but what do I care? I'm not a carp. Live and let live is what I s--'

'It's NOT take-out.' You stare the fish down and for once he actually backstrokes a stroke or two. 'It's PRIVATE.'"

After examining the lingerie bag for the second time:

say "'If it's not take-out, I don't see the relevance!' shouts the fish. 'Food is what you want in this situation. Food for MEEEE.'"

Understand the command "feed" as something new.

Understand "feed [something]" as feeding.

Feeding is an action applying to one visible thing.

Check feeding:

if the noun is not the evil fish, say "That doesn't make much sense." instead;
if the player is not carrying the fish food, say "You need the fish food first."
instead.

Carry out feeding:

increment the score;
say "Triumphantly, you dump the remaining contents of the canister of fish food into the tank. It floats on the surface like scum, but the fish for once stops jawing and starts eating. Like a normal fish. Blub, blub.[paragraph break]";
say "[bold type] *** TWO HOURS LATER ***[roman type][paragraph break]'So,' Britney says, tucking a strand of hair behind your ear, 'where shall we go for dinner? Since I made the big bucks on this trip, it's my treat. Anywhere you like.[paragraph break]'I've had a hankering all day,' you admit, as the two of you turn from the shuttle platform and head toward the bank of taxis. 'I could really go for some sashimi right now.'";
end the story finally.

Before feeding the fish food:

try feeding the evil fish instead.

When play begins:

say "You're on the run. You've got a million errands to do -- your apartment to get cleaned up, the fish to feed, lingerie to buy, Britney's shuttle to meet--
[paragraph break]The fish. You almost forgot. And it's in the studio, halfway across town from anywhere else you have to do. Oh well, you'll just zip over, take care of it, and hop back on the EI. This'll be over in no time.[paragraph break]Don't you just hate days where you wake up the wrong color?[paragraph break]".

The maximum score is 1.

Test me with "x fish / g / kiss fish / x aquarium / x gravel / x seaweed / i / x telegram / x bouquet / smell bouquet / x lingerie / g / x hat / x window / open window / x painting / g / x cabinet / open cabinet / x cloths / search cloths / open food / feed fish".

392

★★ Example Bribery

WI

A GIVE command that gets rid of Inform's default refusal message in favor of something a bit more sophisticated.

If we want to rewrite the functionality of a command that usually ends with a "block..." rule, we will have to begin by turning the blocking off.

"Bribery"

The block giving rule is not listed in the check giving it to rules.

As it happens, correct behavior is built into the GIVE command once "block giving" is turned off, so we do not have to write a replacement report or carry-out rule; the object will be transferred to the possession of the caterpillar. But we do want to adjust the action just a little so that our gift cheers up the recipient:

Carry out giving (this is the gratitude for gifts rule): improve the mood of the second noun.

Mood is a kind of value. The moods are hostile, suspicious, indifferent, friendly, and adoring. An animal has a mood. An animal is usually indifferent.

To improve the mood of (character - an animal):
if the mood of character is less than friendly, now the mood of the character is the mood after the mood of the character.

Now whenever we give something to an animal, the animal will be pleased about the present. Of course, we might also want to add a check rule to giving, to see whether the offering is something the recipient really wants:

Check giving (this is the polite refusal of unwanted objects rule):
unless the noun interests the second noun:
say "[The second noun] disdainfully refuses [the noun]." instead.

To decide whether (item - a thing) interests (character - a person):
if the character has the item, no;
if the item is edible, yes;
no.

Instead of showing something to someone:
try giving the noun to the second noun.

There is already a perfectly workable report rule that will describe what happens when we give something to someone, but let's say we want to report on the recipient's changed mood, too:

After giving something to someone:
say "You give [the noun] to [the second noun], who appears mollified and is now merely [mood of the second noun]."

And the rest is all scenario:

The Leafy Branch is a room. "You stand on smooth bark dappled by sunlight. The scent-trail runs forward to home.

The branch continues forward and backward from here, and a stem extends forward-up."

Instead of going south in Leafy Branch, say "You must not back down! The scent trail leads onward!"

The Very Hungry Caterpillar is a hostile animal in the Leafy Branch. "[The Caterpillar] looks [mood]." Instead of examining the Caterpillar, say "[The caterpillar] appears [mood]."

The player carries an edible thing called a peanut crumb. The carrying capacity of the player is 1. After taking something, say "You lift [the noun], though it is nearly your own size."

Instead of going north in the presence of a hostile caterpillar:

say "[The Caterpillar] moves to block your exit, glaring down at you with all the bristles on its skin extended to full size."

Instead of going north in the presence of a suspicious caterpillar:

say "[The Caterpillar] moves to block your exit, though it might allow you past if you offered further tribute."

The Leaf Face is above the branch. "The smooth and shiny surface of the leaf extends forward from here, but you have lost the scent-trail. This is not the way home." The pear fragment is an edible thing in Leaf Face. The dead aphid is a thing in Leaf Face.

The Twig is north of Leafy Branch. "The scent-trail is weak but not entirely gone, and you pursue it faithfully..."

After going to the Twig:

say "The scent-trail is weak but not entirely gone, and you pursue it faithfully...";

end the story finally.

Understand "forward-up" as up. Understand "forward" as north. Understand "backward" as south. Understand "backward-down" as down.

Test me with "forward / give crumb to caterpillar / forward / forward-up / get aphid / get fragment / down / give aphid to caterpillar / drop aphid / forward-up / get fragment / down / give fragment / forward".

207



Example Barter Barter

WI

Allowing characters other than the player to give objects to one another, accounting for the possibility that some items may not be desired by the intended recipients.

By default, if we make no modifications, telling one player to give something to another will fail, even if persuasion succeeds. This is because the default behavior of the GIVE command is interrupted by the "block giving rule" -- since in many cases we do not want people to exchange objects freely.

However, suppose that we do want characters to be able to exchange articles freely: we allow persuasion to succeed and turn off the "block giving rule".

"Barter Barter"

The block giving rule is not listed in the check giving it to rules.

A persuasion rule for asking people to try giving: persuasion succeeds.

The Trading Post is a room.

Meriwether Lewis is a man in the Trading Post. He carries a fluffy handmade quilt and a bag of beans. The beans are edible.

William Clark is a man in the Trading Post. He carries leather slippers, a journal, and a loaf of bread. The bread is edible. The slippers are wearable.

Instead of examining someone:

say "[The noun] is carrying [the list of things carried by the noun]."

And now we might want to implement a way to keep track of whether the recipient character wants what's being offered:

Check someone trying giving something to someone (this is the sneering refusal rule):

if the second noun dislikes the noun, stop the action.

Unsuccessful attempt by someone trying doing something:

if the reason the action failed is the sneering refusal rule, say "'Would you care for [the noun]?' [the person asked] asks solicitously of [the second noun]."

But [the second noun] refuses [the noun] disdainfully.;"

otherwise say "[The person asked] just appears bewildered by this improbable instruction."

Distaste relates one person to various things. The verb to dislike means the distaste relation.

Clark dislikes the beans. Lewis dislikes the bread.

Since we've defined this as a relation, we could change what the characters like and dislike during the course of the game, freely; for instance, characters might grow hungry and suddenly like all the edible articles.

Test me with "x lewis / x clark / clark, give the slippers to lewis / clark, give the bread to lewis".

The Arena is a room. "Sand, blood, iron. These festivals are normally held on hot days, but the sun has gone behind a cloud and fat drops of rain now and then spatter the arena floor." The gladiator is a man in the Arena. "A bare-chested Scythian gladiator faces you, wielding a trident."

We start by recording, for each person, a maximum number of points of damage the person can sustain when starting from health, and the current number of points remaining. In the tradition of role-playing games, these are referred to as hit points.

A person has a number called maximum hit points. A person has a number called current hit points.

The maximum hit points of the player is 35. The maximum hit points of the gladiator is 25.

The current hit points of the player is 35. The current hit points of the gladiator is 25.

Now our rule for the actual attack. We want first to calculate how much damage the player's attack does, inflict that damage, and remove the enemy if he's dead; then, if he doesn't die, the enemy counter-attacks, also for a randomized amount of damage, and if this kills the player, the game ends in defeat.

Instead of attacking someone:

```
let the damage be a random number between 2 and 10;
say "You attack [the noun], causing [damage] points of damage!";
decrease the current hit points of the noun by the damage;
if the current hit points of the noun is less than 0:
    say "[line break][The noun] expires, and is immediately carried away by the
Arena slaves!";
    now the noun is nowhere;
    end the story finally;
    stop the action;
let the enemy damage be a random number between 2 and 10;
say "[line break][The noun] attacks you, causing [enemy damage] points of
damage!";
decrease the current hit points of the player by the enemy damage;
if the current hit points of the player is less than 0:
    say "[line break]You expire!";
    end the story.
```

This last bit is a refinement to help the player keep track of how the contest is going:

When play begins:

```
now the left hand status line is "You: [current hit points of player]";
now the right hand status line is "Gladiator: [current hit points of gladiator]".
```

Test me with "hit gladiator / g / g / g".

health of someone.

"Red Cross"

Diagnosing is an action applying to one visible thing. Understand "diagnose [something]" as diagnosing.

Check diagnosing:

if the noun is not a person, say "Only people can have diagnoses." instead.

Carry out diagnosing:

say "You're not a medical expert."

Instead of diagnosing the player:

say "You feel fine!"

The Clinic is a room. Fred is a man in the Clinic. The player carries a tongue-depressor.

Test me with "diagnose me / diagnose fred / diagnose tongue-depressor".

201



Example Puff of Orange Smoke

WI

A system in which every character has a body, which is left behind when the person dies; attempts to do something to the body are redirected to the person while the person is alive.

Suppose we want to let the player kill characters, leaving behind corpses.

"Puff of Orange Smoke"

Paraguay is a room. Bolivia is north of Paraguay. Lydia is a woman in Paraguay. "Lydia is, as usual, here." The description of Lydia is "Long, long legs and a sarcastic attitude." Instead of touching Lydia: say "'Watch it, sailor,' she snaps."

A body is a kind of thing. A body is a part of every person. Instead of touching a body: say "[The noun] is grotesquely inert."

The description of Lydia's body is "Long, long legs and no attitude at all." The initial appearance of Lydia's body is "Lydia's corpse is sprawled at your feet."

Using our "part of every person..." line, we've conveniently assigned one body per person. Since we're going to separate people from their bodies when the bodies die, though, we also want a more permanent relation that will help us keep track of which bodies used to belong to which people:

Spirit-possession relates one person to one body. The verb to be owner of means the spirit-possession relation.

When play begins:

repeat with victim running through people:

let the corpse be a random body which is part of the victim;
now the victim is owner of the corpse.

When Lydia is alive, we want >TOUCH LYDIA'S BODY to mean the same thing as >TOUCH LYDIA, so we use the setting action variables rules as a convenient point at which to reassign the action:

Setting action variables when the noun is a body which is part of a person (called owner):

now the noun is the owner.

Setting action variables when the second noun is a body which is part of a person (called owner):

now the second noun is the owner.

This doesn't change Inform's idea about what action is being performed; just about the object it's being performed on. The rest of the action will now proceed as if the player had typed >TOUCH LYDIA.

Along similar lines, once Lydia is dead, we want >MOVE LYDIA to mean >MOVE LYDIA'S BODY if the body is in view:

Setting action variables when the noun is a dead person and the noun is owner of a visible body (called the mortal remains):

now the noun is the mortal remains.

The trick is, though, that >MOVE LYDIA will only be understood if there is something called Lydia that the player can see and refer to, even after she's dead. There are various ways to do this, but the least painful here will be to make the deceased Lydia permanently visible, by putting her in an always-accessible backdrop. The backdrop itself will never be mentioned in the game, and we should make its name something that the player is unlikely to type casually; we don't want the player to interact with it directly. So:

The worldview is a privately-named backdrop. It is everywhere. The spirit-world is a privately-named transparent closed unopenable container. It is part of the worldview.

Definition: a person is dead if he is in the spirit-world.

It's also possible that the player will type something like >X LYDIA when Lydia's corpse is not in view, so we should have an appropriate answer to that as well:

Before doing something to a dead person:

say "[The noun] is dead; or had you blocked that out?" instead.

Because the before rules happen after the setting action variables rules, this will only ever happen if the corpse is not visible.

Now we define the attack itself, which should discard the body, move the spirit to its eternal resting place, and describe the event to the player:

Instead of attacking someone:
let the corpse be a random body which is part of the noun;
move the corpse to the location;
move the noun to the spirit-world;
say "With a single blow, you rid the world of [the noun]."

And finally a trick borrowed from the chapter on understanding, so that we can refer to "Lydia's body" while Lydia is alive, but "Lydia's corpse" only after Lydia has died:

Understand "corpse" as a body when the item described is not part of a person.

Test me with "x body / x lydia's body / touch lydia's body / x corpse / kill lydia / look / x lydia's body / x lydia's corpse / x corpse / x lydia / touch lydia / lydia, hello / n / x lydia / touch lydia / lydia, hello".

293



Example Lanista 2

WI

Randomized combat in which the damage done depends on what weapons the characters are wielding, and in which an ATTACK IT WITH action is created to replace regular attacking. Also folds a new DIAGNOSE command into the system.

Back in the chapter on randomization, we explored a way to create a randomized combat system. That system didn't allow for multiple weapons, though. Here we explore how to create an ATTACK IT WITH action that will let the player choose between weapons with different maximum powers.

We're also going to rewrite that original "instead of attacking:" rule into an attacking it with action that can be performed equally by the player or by any of the player's enemies.

"Lanista, Part Two"

The Arena is a room. "Sand, blood, iron. These festivals are normally held on hot days, but the sun has gone behind a cloud and fat drops of rain now and then spatter the arena floor." The gladiator is a man in the Arena. "A bare-chested Scythian gladiator faces you, wielding [a list of weapons carried by the gladiator]."

Section 1 - Hit Points

A person has a number called maximum hit points. A person has a number called current hit points.

The maximum hit points of the player is 35. The maximum hit points of the gladiator is 25.

In our simpler version of this example we set the current hit points by hand, but in a game with many characters this would get dull and repetitive, so here we'll use a "when play begins" to set all current hit point values automatically to maximum:

When play begins:

repeat with victim running through people:
now the current hit points of the victim is the maximum hit points of the victim.

Definition: a person is dead if his current hit points are less than 0.

Section 2 - Diagnosis

Diagnosing is an action applying to one visible thing. Understand "diagnose [something]" as diagnosing.

Check diagnosing:

if the noun is not a person, say "Only people can have diagnoses." instead.

Carry out diagnosing:

say "[if the noun is the player]You have[otherwise][The noun] has[end if]
[current hit points of the noun] out of a possible [maximum hit points of the noun]
hit points remaining."

Section 3 - Weapons

A weapon is a kind of thing. A weapon has a number called the maximum damage. The maximum damage of a weapon is usually 4.

The gladiator carries a weapon called a trident. The maximum damage of the trident is 5. The gladiator carries a weapon called a net. The maximum damage of the net is 1.

The player carries a weapon called a mace. The maximum damage of the mace is 3.

Section 4 - Attacking it with

In our new system, we want to specify what is being used for an attack. This means that we need to create a new "attacking it with" action, and also that we should disable the existing "attacking..." command.

Here's why: If we leave the default attack command in place, Inform will continue to accept commands like >ATTACK GLADIATOR, but reply foolishly with the default "Violence is not the answer..." response.

A somewhat better approach would be to change the reply of >ATTACK GLADIATOR to say something like "You must specify a weapon to attack with." But this is still less than ideal, because it means that the player has to then rewrite his entire command. If, on the other hand, we take out "ATTACK GLADIATOR" entirely, the game will always prompt "What do you want to attack the gladiator with?" -- which teaches the player the correct command structure for this particular

game, and avoids pretending to understand any command that is not meaningful within this game.

This is a little bit of work because ATTACK has a lot of synonyms in the default library, but if we look through the actions index we can find them all:

Understand the commands "attack" and "punch" and "destroy" and "kill" and "murder" and "hit" and "thump" and "break" and "smash" and "torture" and "wreck" as something new.

Now we make our new command:

Attacking it with is an action applying to one visible thing and one carried thing. Understand "attack [someone] with [something preferably held]" as attacking it with.

Note that we've specified "one carried thing", because we want the player to pick up a weapon to use if necessary. And now we assign all the old attack vocabulary to apply to the new command:

Understand the commands "punch" and "destroy" and "kill" and "murder" and "hit" and "thump" and "break" and "smash" and "torture" and "wreck" as "attack".

This may seem counter-intuitive, but order of source code matters here: we first get rid of the old, default vocabulary, then define our new action, then make the vocabulary apply to that new action. Inform will now understand >HIT GLADIATOR WITH TRIDENT, >BREAK GLADIATOR WITH TRIDENT, and so on.

Our new action is also a perfect place to use an action variable: we're going to need to choose an amount of damage done and refer to that several times in our action rules. So let's set that up first:

The attacking it with action has a number called the damage inflicted.

Setting action variables for attacking something with something:

if the second noun is a weapon:
let the maximum attack be the maximum damage of the second noun;
now the damage inflicted is a random number between 1 and the maximum attack.

Check an actor attacking something with something (this is the can't attack with something that isn't a weapon rule):

if the second noun is not a weapon:
if the actor is the player, say "[The second noun] does not qualify as a weapon.";
stop the action.

Check an actor attacking something with something (this is the can't attack a non-person rule):

if the noun is not a person:
if the actor is the player, say "[The noun] has no life to lose.";
stop the action.

Carry out an actor attacking something with something (this is the standard attacking it with a weapon rule):

- decrease the current hit points of the noun by the damage inflicted;
- if the noun is dead:
 - now the noun is nowhere.

Though our checks and carry-out rules are similar regardless of who is acting, we're going to want actions to be described differently for different actors, so we'll use separate "report attacking" and "report someone attacking" rules. We'll also make some special cases for when the character has died as a result of the attack:

Report attacking a dead person with something (this is the death-report priority rule):

- say "You attack with [the second noun], killing [the noun]!" instead.

Report attacking someone with something (this is the normal attacking report rule):

- say "You attack [the noun] with [the second noun], causing [damage inflicted] point[s] of damage!" instead.

Report someone attacking the player with something when the player is dead (this is the player's-death priority rule):

- say "[The actor] attacks you with [the second noun], finishing you off!";
- end the story;
- stop the action

Report someone attacking the player with something (this is the standard report someone attacking the player with rule):

- say "[The actor] attacks you with [the second noun], causing [damage inflicted] point[s] of damage!" instead.

Report someone attacking something with something (this is the standard report attacking it with rule):

- say "[The actor] attacks [the noun] with [the second noun], causing [damage inflicted] point[s] of damage!" instead.

When play begins:

- now the left hand status line is "You: [current hit points of player]";
- now the right hand status line is "Gladiator: [current hit points of gladiator]".

Every turn (this is the gladiator-attack rule):

- if the gladiator is not dead, try the gladiator attacking the player with a random weapon which is carried by the gladiator.

Test me with "hit gladiator with mace / kill gladiator / drop mace / attack gladiator / attack gladiator with mace / g / g".

Those devoted to role-playing will note that our form of randomization is still pretty naive: most RPG systems use multiple dice in order to create more interesting probability curves. For a system that simulates actual dice-rolling, see the full "Reliques of Tolti-Aph" game.



Example Don Pedro's Revenge

Combat scenario in which the player's footing and position changes from move to move, and the command prompt also changes to reflect that.

Suppose our game features a detailed simulated combat between the player character and his opponent. He might have several weapons available, and several types of attack available; and at any given time he might be perched up in the rigging of his ship, standing on the open deck, or boxed in between some barrels. His options will vary depending on his position, and obviously it would detract from the pacing to make the player keep LOOKing in the middle of combat in order to remind himself where he is. Instead, we'll roll this information into the command prompt:

"Don Pedro's Revenge"

The Deck of the Helene Marie is a room. "The two crews are embattled all around you, but your attention is reserved for your particular enemy: Don Pedro."

Table of Random Prompts

position prompt	
boxed	"So securely boxed-in that you can really only parry or thrust, you try to "
boxed	"Trapped between your barrels, you decide to "
perched	"Able to slice at your attackers but not to advance or retreat, you choose to "
perched	"Perched up here with the advantage of height (but little mobility), you attempt to "
free	"Out on the open deck with no impediments, free to advance or retreat, you decide to "

When play begins: reset the prompt.

Every turn: reset the prompt.

To reset the prompt:

- sort the Table of Random Prompts in random order;
- repeat through the Table of Random Prompts:
 - if the position entry is the placement of the player:
 - now the command prompt is prompt entry;
 - stop.

After reading a command: say conditional paragraph break.

A placement is a kind of value. The placements are boxed, perched, free. The player has a placement. The player is free.

Understand "retreat" or "parry" as retreating. Retreating is an action applying to nothing.

Check retreating:

- if the player is perched, say "You can't move backward or parry very successfully from this position." instead.

Carry out retreating:

- now the player is boxed;

say "You protect yourself, but end up wedged in between two barrels."

Understand "thrust" or "advance" as advancing. Advancing is an action applying to nothing.

Check advancing:

if the player is perched, say "You can't move forward from here, only slash." instead.

Carry out advancing:

now the player is free;

say "You push forward aggressively, making your way to the open deck."

Instead of jumping:

now the player is perched;

say "You leap and swing yourself boldly up into the rigging, leaving your attackers beneath you."

Instead of jumping when the player is perched:

now the player is free;

say "You leap down from your position, into the middle of the deck."

Test me with "advance / jump / advance / retreat / jump / retreat / retreat / advance".

Of course, this won't be much fun until we also provide the player with a few weapons, some more fighting maneuvers, and, most of all, a Don Pedro to defeat.

127



Example Technological Terror

WI

A ray gun which destroys objects, leaving their component parts behind.

"Technological Terror"

The Decomposition Ray Gun is a thing carried by the player.

First we need to define our shooting action:

Shooting it with is an action applying to two things.

Check shooting something with something:

if the player is not carrying the Ray Gun, say "You are pathetically unarmed!" instead;

if the second noun is not the Ray Gun, say "[The second noun] does not fire." instead;

if the noun is the Ray Gun, say "Nice trick if you can do it!" instead;

if the noun is the player, say "That would be disastrous!" instead.

Next, some grammar to allow the player to use this action:

Understand "shoot [gun] at [something ungunlike]" as shooting it with (with nouns reversed).

Definition: a thing is ungunlike if it is not the gun.

Understand "shoot [something ungunlike] with [gun]" as shooting it with.
Understand "shoot [something] with [something]" as shooting it with.

Understand "shoot [something] at [something]" as shooting it with (with nouns reversed). Understand "fire [gun] at [something ungunlike]" as shooting it with (with nouns reversed). Understand "fire at [something ungunlike] with [gun]" as shooting it with. Understand "fire at [something] with [something]" as shooting it with.

Strictly speaking, we only need these last grammar lines (with "understand shoot something...") in order to define an action that the player can take. Adding more grammar lines means that Inform will try to match the most specific ones first, which is useful when the player types something ambiguous and there is one choice that obviously fits this action better than the others. See the chapter on Understanding for a further discussion.

Here we get to use "now..." to give it its destructive effect:

Carry out shooting something with something:

say "ZAP! [The noun] twinkles out of existence! [if something is part of the noun][The list of things which are part of the noun] clatter to the ground! [end if] [paragraph break]";

now every thing which is part of the noun is in the location;
now the noun is nowhere.

The Deathbot Assembly Line is a room. "Here is the heart of the whole operation, where your opponents are assembled fresh from scrap metal and bits of old car." The dangerous robot is a thing in the Assembly Line. "One dangerous robot looks ready to take you on!" A robotic head, a drill arm, a needle arm, a crushing leg and a kicking leg are parts of the dangerous robot.

Instead of examining something when something is part of the noun:

say "[The noun] consists of [a list of things which are part of the noun]."

Test me with "x robot / shoot robot / fire at kicking leg / shoot gun at drill arm / look".

If the player tries to TALK TO a character, suggest alternative modes of conversation.

"Gorge at George"

The Dusty Lot is a room. "A few miles up the road from the concert venue, but at least it's cheap to park here."

The motorcyclist is a man in the Dusty Lot. "A man clad in [a list of things worn by the motorcyclist] leans against his Harley and watches you without saying anything." The Harley is scenery in the Lot. The motorcyclist wears a black leather jacket and shades. Understand "man" or "guy" as the motorcyclist.

Understand "talk to [someone]" as a mistake ("To start a conversation, try to ASK [the noun] ABOUT something or TELL [the noun] ABOUT something.").

Instead of asking the motorcyclist about something:
say "He smirks cryptically."

Instead of telling the motorcyclist about something:
say "This does not seem to interest him much."

Test me with "talk to motorcyclist / ask motorcyclist about himself / tell motorcyclist about me".

110



Example Mimicry

WI

People who must be greeted before conversation can begin.

Suppose we want to add a sense of some conversational flow, so that the player is forced to acknowledge the presence of people before beginning detailed conversations with them. We collect all speech actions into a single category:

"Mimicry"

Asking someone about something is speech. Telling someone about something is speech. Answering someone that something is speech. Asking someone for something is speech.

And then write a general rule.

Before speech in the presence of an ungreeted person: try waving hands.

One complication is that "asking someone to try doing something", which describes commands such as FRED, GO SOUTH, cannot be made into a kind of action. This requires its own rule:

Before asking someone to try doing something in the presence of an ungreeted person: try waving hands.

Now we define what greetings are going to look like:

Check waving hands:

unless the player can see someone who is not the player, say "You are alone." instead.

Carry out waving hands:

say "You nod hello to [the list of ungreeted people who can be seen by the

player].";
now every ungreeted person who can be seen by the player is greeted.

The report waving hands rule is not listed in the report waving hands rulebook.

Because of the way we've defined the command, this will now also work if the player waves.

A person can be greeted or ungreeted. A person is usually ungreeted. The player is greeted.

And now the scenario:

The International Convention of Mimes is a room. Lester, Harold, Geoff, Kwame, and Peter are men in the Convention. Elouise is a woman in the Convention. The Invisible Box is an enterable container in the Convention. "You can detect, from the way people keep leaning on it, an invisible box in the middle of the room."

Lester carries a bowler hat.

Instead of speech in the presence of someone:
describe poor reception.

Definition: a person is other if it is not the player.

At 9:01 AM:
move Phineas to the location; say "A mime called Phineas appears from the non-existent bathroom."

Phineas is a man.

A persuasion rule:
describe poor reception;
persuasion fails.

To describe poor reception:
if the player is in the Invisible box,
say "Everyone convulses with silent laughter as you try to shout from within the invisible box.";
otherwise
say "You attempt to convey your meaning with gesture and interpretive dance, but [the list of visible other people] scorn[if the number of visible other people is 1]s[end if] your performance, refusing to respond."

Test me with "ask lester about work / lester, east / ask lester for bowler / lester, nice not talking to you / get in box / ask lester for hat / phineas, east".

Sometimes we do not particularly want to deal with all the variations on asking, telling, or answering someone something, but want to direct everything to a single conversational command:

"Consulting the Oracle"

The Grove is a room. In the Grove is a woman called the Sybil.

Instead of telling someone about something, try asking the noun about it.
Instead of answering the noun that something, try asking the noun about it.

Instead of asking the Sybil about "persians", say "She nods gravely."

And similarly, a difference between GIVE and SHOW is sometimes overkill:

Instead of showing something to someone, try giving the noun to the second noun.

The player carries a coin. Instead of giving the coin to the Sybil: move the coin to the Sybil; say "She accepts with a smile."

It is also often the case that we want to accept more than one form of a term. For instance

Instead of asking the Sybil about "Darius/king", say "Her smile unnerves you."

will match either "Darius" or "king". If necessary, we can go a step further and define our own token to match a variety of phrases, like this:

Understand "Athenians/Spartans/Greeks" or "hoplite army/forces" as "[Greeks]".
Instead of asking the Sybil about "[Greeks]", say "She looks encouraging."

The token "[Greeks]" will match all of "Athenians", "Spartans", "Greeks", "hoplite army", or "hoplite forces". It will not match "hoplite" or "forces" alone; it is important to note that the / divides individual words which are understood equivalently, but does not define entire phrases as equivalent. More about how Inform understands specific phrases can be found in the chapter on Understanding.

Test me with "test one / test two".

Test one with "ask sybil about persians / tell sybil about persians / sybil, persians / ask sybil about darius / ask sybil about king".

Test two with "ask sybil about greeks / ask sybil about athenians / ask sybil about hoplite army / ask sybil about hoplite forces / give the coin to the sybil".



TELL CHARACTER YES, ANSWER YES, and CHARACTER, YES.

Inform already understands YES, NO, and SORRY as commands in their own right, which can make things a little sticky when we want a character to ask a question of the player. The most important thing is not to cover some of the possible phrasings while ignoring others.

"Replies"

The Grove is a room. In the Grove is a woman called the Sybil.

Instead of asking the Sybil to try saying no: try saying no. Instead of asking the Sybil to try saying yes: try saying yes. Instead of asking the Sybil to try saying sorry: try saying sorry.

Instead of answering the Sybil that "yes", try saying yes. Instead of answering the Sybil that "no", try saying no. Instead of answering the Sybil that "sorry", try saying sorry.

Instead of saying yes in the presence of the Sybil:
say "She looks interested."

Instead of saying no in the presence of the Sybil:
say "She looks annoyed."

Instead of saying sorry in the presence of the Sybil:
say "She looks bored."

The complexity arises from the fact that we want to handle both YES and SYBIL, YES. If we only had the latter, 'yes' would be treated as a text given to the Sybil, just as in the commands SAY YES TO SYBIL or ANSWER YES. But because we have defined it as a command (so that the player can use it independently), SYBIL, YES is understood as an order to the Sybil to do the YES action.

Fortunately, we can redirect everything, as here, so that the results wind up the same.

And if we want yet another variation not covered by the Inform standard:

Understand "tell [someone] [text]" as answering it that. Understand "tell [someone] that [text]" as answering it that.

But that is a matter for a later chapter.

Test me with "yes / sybil, yes / say yes to sybil / answer yes / tell sybil yes / no / sybil, no / say no to sybil / answer no / tell sybil no / sorry / sybil, sorry / say sorry to sybil / answer sorry / tell sybil sorry".



which he may answer or not as he chooses.

Suppose we want to ask the player a question where he might say yes or no in response. There are two possible forms of this: the modal question where the player must pick one to proceed, and the non-modal question where he might also type other verbs.

"Proposal"

The story genre is "A Worked Example about Yes/No Questions".

Section 1 - Asking a Modal Yes/No Question

When play begins:

say "Do you like Mr Spruce? ";
if player consents, now Spruce is handsome;
otherwise now Spruce is ugly;
say paragraph break.

Section 2 - Mr Spruce's Non-Modal Question

Use full-length room descriptions.

The Conservatory is a room. "You are in a room full of plants."

Mr Spruce is a man in the Conservatory. Mr Spruce can be apprehensive or calm. Mr Spruce is calm. Mr Spruce can be handsome or ugly.

At 9:02 AM: say "Mr Spruce flings himself to his knees and implores you to become his lawfully wedded wife.";
now Mr Spruce is apprehensive;
Mr Spruce gives up in two minutes from now.

At the time when Mr Spruce gives up:

say "Mr Spruce sighs heavily, seeing that you don't intend to reply. 'Never mind, my dear, I'll ask later. Perhaps I should have spoken to your Papa first... yes, a gently-bred female... no wonder...';"
now Mr Spruce is calm;
Mr Spruce departs in one minute from now.

At the time when Mr Spruce departs:

if the player can see Mr Spruce, say "Mr Spruce takes his leave of you.";
otherwise say "Mr Spruce pokes his head in to say that he is leaving.";
end the story saying "Well, that is over..."

Instead of saying yes in the presence of an ugly apprehensive Mr Spruce:

now Mr Spruce is calm;
say "Remembering what your mother said to you about the stock exchange and Dear Papa, you close your eyes and accept Mr Spruce.";
end the story saying "Alas for your maiden hopes."

Instead of saying yes in the presence of a handsome apprehensive Mr Spruce:

now Mr Spruce is calm;

say "You are silent with delight for a moment before you say yes, yes!";
end the story saying "How Genevieve Stubbs will cry!"

Instead of saying no in the presence of an ugly apprehensive Mr Spruce:
now Mr Spruce is calm;
say "Gently you inform Mr Spruce that it is impossible. He seems less deflated
than you had expected.";
end the story saying "Odd, that..."

Instead of saying no in the presence of a handsome apprehensive Mr Spruce:
now Mr Spruce is calm;
say "You lower your eyes and refuse petulantly, hoping to stir him to a more
ardent repetition of these same requests. But then -- alack! -- he says 'I see how
it is!' in a strangled voice, and strides from the room!";
end the story saying "A fatal error!"

And since the player might SAY YES TO SPRUCE, we had better reroute the
relevant options:

Instead of answering Mr Spruce that "no", try saying no.

Instead of answering Mr Spruce that "yes", try saying yes.

Instead of asking Mr Spruce to try saying yes, try saying yes.

Instead of asking Mr Spruce to try saying no, try saying no.

Instead of saying sorry, try saying no.

Instead of asking Mr Spruce to try saying sorry, try saying no.

Instead of answering Mr Spruce that "sorry", try saying no.

Test me with "z / z / z / yes".

Test more with "z / z / z / no".

297



Example Nameless

WI

ASKing someone about an object rather than about a topic.

By default, ASK SOMEONE ABOUT... applies only to a text token. We might want also to offer the player the option of asking characters about pieces of physical evidence. This example implements an ASK PERSON ABOUT THING command that is mostly synonymous with SHOW, with the added nuance that the player can ask about things that are not currently visible, as long as he has encountered them at some time in the past.

"Nameless"

The Black Chamber is a room. "Despite its menacing name, it is quite an ordinary room, underlying the post office above. Here letters are brought each day, unsealed, transcribed, resealed, and sent again on their way; their contents then analyzed and recorded."

The Nameless Advisor is a woman in the Black Chamber. "A woman whose name has never been disclosed to you sits at the window, writing numbers on a sheet of paper." The Advisor carries a sheet of paper. Understand "woman" as the nameless advisor.

The player carries a letter from the emperor. The description of the letter is "Though its origin is obvious, its meaning is secret: the letters are an inexplicable jumble."

Now we create our new action, "interrogating it about". We write the grammar lines so that we can show any object in sight to someone, but also ask someone about any object that we have ever interacted with in the game, whether it is currently visible or not.

A thing can be known or unknown. The Nameless Advisor is known.

Understand "ask [someone] about [any known thing]" as interrogating it about. interrogating it about is an action applying to two visible things.

Now we replace and redirect the showing action. This gets rid of the requirement in the default library that the player be holding anything he shows to another character:

Understand the commands "show" and "display" and "present" as something new.

Understand "show [something] to [someone]" or "display [something] to [someone]" or "present [something] to [someone]" as interrogating it about (with nouns reversed). Understand "show [someone] [something]" as interrogating it about.

This bit keeps track of what the player has seen, for the purposes of "any known thing":

Before printing the name of something (called the target): now the target is known.

Here we define what happens by default when we interrogate someone about something; we use the same response we get to asking someone about something that isn't otherwise interesting:

Carry out interrogating someone about something:
say "There is no reply."

Now redirect all asking to a topic table, and all interrogating to an object table:

Instead of asking Nameless Advisor about a topic listed in the Table of Nameless Advisor Topics:
say "[reply entry][paragraph break]".

Instead of interrogating Nameless Advisor about an item listed in the Table of Nameless Advisor Items:

say "[reply entry][paragraph break]".

Table of Nameless Advisor Items

item	reply
letter	"'It is enciphered,' she remarks[if the advisor can see the letter], glancing over the contents[otherwise], after you have offered a detailed description[end if]. 'A substitution cipher of some complexity, I believe.'"
Advisor	"She listens to your inquiries about her identity and parentage with a placid smile, but does not answer."

Table of Nameless Advisor Topics

topic	reply
"cipher"	"'I know many dozens of ciphers,' she replies, smiling in a disquieting way."
"substitution cipher"	"'One letter is allowed to stand for another,' she explains, folding her hands together patiently. The backs of both hands are tattooed with silvery stars."

And just so that we can test what happens when asking someone about something out of sight:

The safe box is a container in the Chamber. It is fixed in place. It is openable and closed.

...and something unknown:

The poisonous apple is a thing.

Test me with "test sight / test knowledge".

Test sight with "i / x letter / ask Nameless Advisor about cipher / show cipher to Nameless Advisor / ask Nameless Advisor about the letter / show the letter to Nameless Advisor / show Nameless Advisor the letter".

Test knowledge with "open safe box / put letter in safe box / close safe box / ask Nameless Advisor about the letter / show the letter to Nameless Advisor / ask Nameless Advisor about the apple".

281



Example Farewell

WI

People who respond to conversational gambits, summarize what they said before if asked again, and provide recap of conversation that is past.

We begin with the idea that each person comes with his own table of things to say:

"Farewell"

A person has a table name called conversation.

Instead of asking someone about something:
let the source be the conversation of the noun;

```

if topic understood is a topic listed in source:
  if there is a turn stamp entry:
    say "[The noun] has already told you that [summary entry].";
  otherwise:
    now turn stamp entry is the turn count;
    say "[reply entry][paragraph break]";
otherwise:
  say "[The noun] stares at you blankly."

```

For the sake of simplicity, we'll conflate asking and telling here, though it would certainly be possible to have a more complex implementation if we want the characters to be told things as well.

```

Instead of telling someone about something:
  try asking the noun about it.

```

Now we might want to add a recap command to review conversation that has already occurred.

```

Definition: a person is other if it is not the player.

```

```

Understand "recap" or "recall" or "review" as recalling conversations.

```

```

Recalling conversations is an action applying to nothing.

```

Since we've been recording the turn count of each conversation bit, we can even present these in order by sorting the tables first.

```

Carry out recalling conversations:
  repeat with speaker running through other people:
    let source be the conversation of the speaker;
    sort source in turn stamp order;
    say "[The speaker] has so far told you: [line break]";
    let index be 0;
    repeat through source:
      if there is a turn stamp entry:
        let index be 1;
        say " [summary entry][line break]";
      if index is 0, say " absolutely nothing[line break]";
    say conditional paragraph break.

```

Now it remains only to create a couple of characters and provide them both with something to say:

```

The Farewell Bend Cafe is a room. "Beautiful Farewell Bend, Idaho -- or is it Oregon? An almost-abandoned truckstop, in any case, on one of those interminable American east-west highways."

```

```

Tina is a woman in the Farewell Bend Cafe. The conversation of Tina is the Table of Tina's Chatter. "Tina the waitress is slowly pouring coffee from the pot with a black neck into the pot with an orange neck."

```

```

George is a man in the Farewell Bend Cafe. The conversation of George is the Table of George's Chatter. "There is also a large man at table five. The tattoo on

```

his arm says George. For the moment we will assume that it is his own name and not someone else's."

Table of Tina's Chatter

topic	reply	summary	turn stamp
"aasvogel"	"Oh, it's a vulture."	"that an aasvogel is a vulture"	a number
"acaudate"	"She shrugs, mid-pour. 'Means something doesn't have a tail.'"	"that acaudate means 'tailless'"	--
"absorptiometer"	"It's a thing that measures the solubility of gases in a liquid,' she explains gently, as to a child."	"that an absorptiometer measures solubility of gasses in a liquid"	--

Table of George's Chatter

topic	reply	summary	turn stamp
"baccaceous"	"Something that has or bears berries,' says George, without looking up."	"that baccaceous means berry-bearing or berry-like"	a number
"bagheera"	"Oh, that'd be a velvet-like textile."	"that bagheera is a velvet-like textile"	--
"balistarius"	"That's a crossbow-man,' George replies instantly."	"that a balistarius is a crossbow-man"	--

A word of warning: this system does assume that every person in the game has a conversation table defined. If that were not the case, we would have to be a bit more careful.

As always, we can override specific words, too:

Instead of asking Tina about "advertisement" for the first time:
 say "Tina looks embarrassed. 'Of course! I almost forgot.' She hands you a brochure.";
 move the brochure to the player.

The encyclopedia sales brochure is a thing. The description is "A glossy flyer indicating that you can receive a free Volume A-Aalto of the New Idahoan Encyclopedia Set if you send back the business reply card, and then have the option of purchasing the remaining volumes at a very very reasonable price."

Test me with "recap / ask tina about aasvogel / recap / ask george about baccaceous / ask tina about absorptiometer / recap / ask tina about advertisement / read brochure".



Example Sweeney

A conversation where each topic may have multiple questions and answers associated with it, and where a given exchange can lead to new additions to the list.

"Sweeney"

A subject is a kind of thing. Some subjects are defined by the Table of Conversation Subjects.

Table of Conversation Subjects

subject	conversation
pies	Table of Pie Queries
employment	Table of Job Queries

Understand "job" as employment. Understand "meat" or "food" as pies.

Table of Job Queries

quip	discussion	label	subtopics
"whether there is a job available here"	"Say, are you hiring?' you ask, as casually as you can manage. [The interlocutor] looks you over dubiously. 'I might be hiring someone, but I can't say it would necessarily be you.'"	0	--
"what happened to that boy that worked here"	"Tell me, didn't you used to have a young assistant working here?' She shrugs. 'Young men these days are so unstable. He left-- who knows where he's gone? I haven't seen hair or fingernail of him for weeks.'"	0	--
with 3 blank rows.			

Table of Pie Queries

quip	discussion	label	subtopics
"what pie fillings are available"	"What pies do you have in today, Mrs Lovett?' you ask. She starts, then smiles. 'Meat pies, of course.'"	0	Table of Pie Flavor Queries
with 3 blank rows.			

Table of Pie Flavor Queries

quip	discussion	label	subtopics
"what kind of meat"	"What kind of meat goes into these pies, Mrs Lovett?' you ask pressingly. She looks shifty. 'Whatever the butcher brings this week,' she says. 'With the price of meat what it is, when you get it, you have to be glad of what you can get. If you get it.'"	0	--

To copy (first table - a table name) to (second table - a table name):

repeat through first table:

- let copied quip be "blank";
- if there is a quip entry, now the copied quip is the quip entry;
- let copied discussion be "blank";
- if there is a discussion entry, now the copied discussion is the discussion entry;
- let copied subtopics be second table;
- if there is a subtopics entry, now the copied subtopics are the subtopics entry;
- choose a blank row in the second table;
- if copied quip is not "blank", now quip entry is copied quip;
- if copied discussion is not "blank", now discussion entry is copied discussion;
- if copied subtopics is not second table, now subtopics entry is copied subtopics.

Current conversation table is a table name that varies. Current conversation table is Table of Job Queries.

Interlocutor is a person that varies.

Understand "ask [someone] about [any subject]" as asking it about the subject.

Asking it about the subject is an action applying to two visible things.

Carry out asking it about the subject:
say "You can't think of anything to say."

Instead of asking someone about the subject a subject listed in the Table of Conversation Subjects:

- now interlocutor is noun;
- now current conversation table is the conversation of the second noun;
- if the number of filled rows in the current conversation table is 1:
 - repeat through current conversation table:
 - now label entry is 1;
 - now number understood is 1;
 - try selecting 1 instead;
 - if the number of filled rows in the current conversation table is 0:
 - say "You can think of nothing further to say on that topic.";
 - stop the action;
 - otherwise:
 - let index be 0;
 - let total be the number of filled rows in the current conversation table;
 - say "Do you mean ";
 - repeat through current conversation table:
 - now index is index + 1;
 - now label entry is index;
 - say "([index]) [quip entry]";
 - if index is total, say "?";
 - if index is total - 1, say ", or ";
 - if index is less than total - 1, say ", ".

Understand "[number]" as selecting.

Selecting is an action applying to one number.

Carry out selecting:
say "No such option is available."

Instead of selecting a label listed in the current conversation table:
say "[discussion entry][paragraph break]";
if there is a subtopics entry:

- copy subtopics entry to current conversation table;
- choose row with label of number understood in the current conversation table;
- blank out the whole row.

Mrs Lovett's Meat Pies is a room. Mrs Lovett is a woman in Meat Pies.

Test me with "ask lovett about pies / ask lovett about employment / 1 / 2 / ask lovett about pies".



As we have seen, there are a number of different ways of controlling conversation in interactive fiction, and the best choice of way will depend quite a lot on what kind of work we're writing.

One common model is to replace Inform's default ASK and TELL commands with a TALK TO command. This gives the player less control than he would otherwise have: instead of asking a character about any topic under the sun, he's restricted to seeing (or not seeing) a single sequence of text that the author has written in advance. On the other hand, such a system is harder for the player to break (since he can never ask about a topic that the author hasn't implemented), and easier for the author to tie into plot developments. If we give TALK TO different output at each scene, we get conversation that is always tied to the current state of the plot.

This is a design approach that works best in a game with a large number of short, focused scenes. For other kinds of conversation system design, compare the other examples listed in the Recipe Book.

"The Cheese-makers" by Phrynichus.

Chapter 1 - Replacing old talk commands and making a new one

Here, using some techniques that will be discussed in the chapter on Understanding, we get rid of Inform's default handling of ASK and TELL, and create our own TALK TO action instead:

Understand the commands "ask" and "tell" and "say" and "answer" as something new.

Understand "ask [text]" or "tell [text]" or "answer [text]" or "say [text]" as a mistake ("[talk to instead]").

Instead of asking someone to try doing something:
say "[talk to instead][paragraph break]".

Instead of answering someone that something:
say "[talk to instead][paragraph break]".

To say talk to instead:
say "(To communicate in [story title], TALK TO a character.) "

Understand "talk to [someone]" as talking to. Understand "talk to [something]" as talking to. Talking to is an action applying to one visible thing.

Chapter 2 - Specific scenes and talking

Now, suppose we have a situation -- say, a stage play -- in which it is appropriate to talk to different characters at different times. During the prologue of the play, no one else is on-stage, and the player is to address the audience directly:

Section 1 - Prologue

When play begins:

now right hand status line is "416 BC";
now left hand status line is "[location]".

Prologue is a scene. Prologue begins when play begins.

The Theater of Dionysus is a room.

The audience is a person in the Theater. "The usual audience looks on: the priests and judges in the front row, and then Athenians, metics, and foreigners." The audience can be prepared or unprepared. The description is "Have you ever seen such a company of perjurers, pathics, and thieves?" Understand "priest" or "priests" or "priest of dionysus" or "judge" or "judges" or "athenians" or "metics" or "foreigners" as the audience.

Instead of talking to the player when the Prologue is happening:

say "There will be plenty of occasion for muttered asides later in the play, but for now you must prepare the audience for things to come."

Instead of talking to the audience when the Prologue is happening:

say "Drawing breath, you turn to the audience, and offer them a genial, witty, colorful, and of course crude synopsis of what they are about to see; describing all the characters in unmistakable terms and not omitting the most important of them all, your august self.";
now the audience is prepared.

Instead of talking to the audience when the Prologue has happened:

say "You may only direct monologues to the audience when the other actors are off-stage. Otherwise, their characters might have to notice."

Prologue ends when the audience is prepared.

But there might follow a scene in which the player shouldn't talk at all:

Section 2 - Parodos

Parodos is a scene. Parodos begins when Prologue ends.

When Parodos begins:

move the chorus to the theater.

Instead of talking to someone during Parodos:

say "Sssh: this moment belongs to the chorus. They've worked so hard on it, after all."

Parodos ends when the time since Parodos began is 4 minutes.

The chorus is a person. The description is "They are dressed in exaggerated rural costume and feminine masks, as they are meant to represent a company of female cheese-makers from the Spartan-occupied deme of Dekeleia." Understand "cheesewives" or "cheese-makers" or "chorus-leader" as the chorus.

Every turn during Parodos:

repeat through Table of Choral Events:

say "[output entry][paragraph break]";
blank out the whole row;
make no decision.

Table of Choral Events

output

"The chorus now begins its entry, accompanying with anapestic song its march up the eisodos."

"The chorus draws nearer, stomping and clomping and swinging their baskets of cheese."

"You stand aside as the chorus fills the orchestra and dances to and fro."

"The tune of the aulos-player grows more and more frenzied and then breaks off."

This last rule is a refinement borrowing from the Activities chapter, which gives characters different appearances in room descriptions depending on when we happen to look; because of the action of the play, we want to show the chorus and audience doing different things during different scenes.

Rule for writing a paragraph about the chorus during Parodos:

say "The chorus are dancing and singing their way[if the time since Parodos began is less than 3 minutes] up the long walkways onto the stage[otherwise] into position in the orchestra[end if]. [The audience] appear to be pricing their costumes to the nearest obol: woe to the producer who cheats them of their due share of spectacle."

And now a scene in which the player can talk several times to a character (Heracles) but has no useful dialogue with the chorus, the audience, or himself. The prohibition from talking to the audience after the Prologue is already written, but we'll supply some appropriate responses for talking to the player or the chorus during this scene:

Section 3 - Episode

Episode is a scene. Episode begins when Parodos ends.

When Episode begins:

move Heracles to the theater;
say "The chorus falls silent, which is the cue: Heracles bursts out of the scene building."

Heracles is a man. The description is "Hard to mistake in his lion skin and boots, and carrying a formidable club." Heracles wears a lion skin and boots. He carries a formidable club. Heracles can be placid or annoyed. Heracles is placid. Heracles can be satisfied, intrigued, or unsatisfied. Heracles is unsatisfied.

Instead of talking to the chorus during Episode:

say "Your improvised flirtation with the chorus raises no response but a crude gesture from the chorus-leader, who seems to be modeling the role on lambe."

Instead of talking to the player during Episode:

if Heracles is annoyed:
say "You mutter to yourself about men with more appetite than brain. The actor playing Heracles ignores you, but it's good odds he's scowling under his mask. He hates it when anyone but himself ad-libs for attention.";

otherwise:
now Heracles is annoyed;
say "'By the dog, he'll eat me if he gets a chance,' you mutter aside.
[paragraph break]'What's that you say, my ignoble friend?' demands Heracles,
hefting his club. He's not entirely joking: you've left the script just now."

Instead of talking to Heracles when Heracles is unsatisfied during Episode:
say "'Dear Heracles, friendly Heracles,' you begin, cringing out of the way as
he responds with one of his affectionate ox-killing punches to the shoulder.
[paragraph break]But Heracles falls still, and looks almost thoughtful, as tell him
you know how he may rout the Spartans, woo all twenty-four lactic ladies, and
tame his savage gut with a bathtubful of porridge. [paragraph break]'Speak on,
little man,' he says.";
now Heracles is intrigued.

Instead of talking to Heracles when Heracles is intrigued during Episode:
say "It takes several exchanges for him to wrap his one-inch brain around
your ten-inch plan; but in the end he embraces the scheme, the women, and
your humble self.";
now Heracles is satisfied.

Every turn when not talking to someone during Episode:
repeat through Table of Episodic Events:
say "[output entry][paragraph break]";
blank out the whole row;
make no decision.

Table of Episodic Events
output

"With a fart and a roar, Heracles asks the world at large, and you in particular,
where his dinner might be."

"In epic diction, Heracles invites the dairy-mistresses, whey-matrons, and
concubines of curd to supply him a supper from their ample baskets."

"Heracles and the chorus banter about the proclivities of cheese-wives. The
chorus suggest that Heracles, as a son of Zeus, must know something about the
appetites of which they speak."

"Heracles boasts that a man like himself can perform any feat, but only when his
belly is full. Coyly, the matrons prance and dance, skip and gambol out of his
grasp, singing mockingly about heads of garlic and loaves of sesame-crust
bread."

"The song of the feta fanciers now turns to pots of honey and new-made wine,
borrowing verses from last year's Lenaia winner, 'The Bees'. With a jolt, you
realize that you've missed your cue and the chorus are filling in for you."

"Playing for time, the chorus-leader elaborates a whole banquet: rabbit stew,
shanks of lamb, spitted quails, eels from lake Copais. Heracles looks as near
swooning as any girl fresh from Brauron."

"The chorus-leader extends the list of delicacies to include ox-brains, ham-
hocks, barley, mullet, carrots, pigeons, lentils, radishes, peas, and apples both
wine-dark and golden. The audience shifts on the benches. An expression of
gloom settles over the Priest of Dionysus in the front row."

"Inspired by Euripides' own Muse, the chorus-leader invents a mock-Alcaean
hymn on the merits of chervil. This is clearly his swan-song: if you don't speak at
last, the play will come to a halt."

"Silence descends."

Rule for writing a paragraph about Heracles during Episode:

say "[Heracles] stands at the center of the orchestra, with members of [the chorus] ranged on either side. [paragraph break][The audience] appear to be reserving their judgement, though they show signs of restiveness at the usual jokes: must there be a Heracles in [italic type]every[roman type] play?"

Episode ends successfully when Heracles is satisfied.

When Episode ends successfully:

say "That, of course, is your cue: you're to come back on as Pan thirty verses from now, and it takes time to put on the hooves and the woolly-legged trousers.";

end the story saying "You exit".

Episode ends disastrously when the number of filled rows in the Table of Episodic Events is 0.

When Episode ends disastrously:

end the story saying "The production has crashed to a halt".

Test me with "ask audience about me / tell audience about me / audience, hello / audience, jump / talk to me / talk to audience / g / talk to chorus / look / x heracles / talk to me / talk to audience / z / look / talk to heracles / g".

376



Example Complimentary Peanuts

WI

A character who responds to keywords in the player's instructions and remarks, even if there are other words included.

The "reading a command" activity is not the only point at which we can interact with snippets, as it happens; it is merely the most useful. "The player's command" can be consulted at other points, however, as in this example of your somewhat deaf (or distracted, or simply cussed) Aunt:

"Complimentary Peanuts"

Instead of asking Aunt Martha to try doing something:

repeat through Table of Aunt Martha's Commentary:

if player's command includes topic entry:

say "[commentary entry][paragraph break]";

rule succeeds;

say "'Hmmf,' says Aunt Martha."

The topic understood is also a snippet, so that whenever one has been generated, we can treat it in the same way as "the player's command":

Asking someone about something is speech.

Telling someone about something is speech.

Answering someone that something is speech.

Asking someone for something is speech.

Instead of speech when the noun is Aunt Martha:
 repeat through Table of Aunt Martha's commentary:
 if the topic understood includes topic entry:
 say "[commentary entry][paragraph break]";
 rule succeeds;
 say "'Hmmf,' says Aunt Martha."

This is superior to checking "the player's command" because we do not want ASK MARTHA ABOUT FRENCH FRIES to trigger the "Martha" keyword, only the "french fries" keywords.

The Empyrean Shuttle Bay is a room. "From here you have an excellent view of the colony world, which looks... well, it looks discouragingly orange. But terraforming is in progress."

Aunt Martha is a woman in the Empyrean Shuttle Bay. A gleaming shuttle and a stack of rations are in the Shuttle Bay. The shuttle is a vehicle. "Your shuttle awaits."

Table of Aunt Martha's Commentary

topic	commentary
"shuttle"	"Shuttles! I hate shuttles,' Aunt Martha grumbles. 'Give me an airplane! AIRPLANE.'"
"airplane/airport"	"Those were the days,' Aunt Martha agrees, plainly reliving the days when she wore a blue-and-white uniform and passed out packets of salted pretzels."
"rations"	"Do you think there are any peanuts in there?' she asks in a wistful tone."

Test me with "martha, get in the shuttle / martha, for pity's sake, do you see an airplane around here? / martha, pass me the rations".

This means that Martha will respond to keywords regardless of the setting in which they occur. For instance:

>martha, get in the shuttle
 "Shuttles! I hate shuttles," Aunt Martha grumbles. "Give me an airplane! AIRPLANE."

>martha, for pity's sake, do you see an airplane around here?
 "Those were the days," Aunt Martha agrees, plainly reliving the days when she wore a blue-and-white uniform and passed out packets of salted peanuts.

>martha, pass me the rations
 "Do you think there are any peanuts in there?" she asks in a wistful tone.

This is not the stuff of which Loebner-winning chatbots are made, admittedly, but it is occasionally a useful alternative to stricter modes of command-parsing.

Let's say that we're implementing a particularly irrational and volatile character. Some of the time she remains composed; some of the time she reacts with unexpected vehemence for reasons only partly related to what was said.

Moreover, her responses are divided between successful and failing outcomes, where success indicates that she's not too upset and failure means that she is distraught; we use this to determine how the rest of the room reacts.

"Being Peter"

The Drawing Room is a room. "The company is assembled here for champagne. Most of it, anyway: Mary is on the phone to her babysitter, Roger is keeping her anxious company, and Carol doesn't drink. But everyone else."

Maggie is a woman in the Drawing Room.

The player wears a top hat.

Quizzing it about is an action applying to one thing and one visible thing. Understand "ask [someone] about [any thing]" as quizzing it about.

Instead of quizzing Maggie about something:

- follow the attitude rules;
- say "Everyone waits to see what the reaction will be: [outcome of the rulebook].";
- if rule succeeded, say "There is general relief.";
- otherwise say "Everyone is pointedly silent."

The attitude rules are a rulebook. The attitude rules have outcomes she stays calm (no outcome - default), she gets angry (failure), she has a stroke (failure), she is only mildly annoyed (success), and she is elated (success).

Here we want Inform to consult every appropriate attitude rule until it gets to some answer; if an attitude rule does not provide a result, the default 'no outcome' will mean that we go on to the next rule, and so on.

A subject is a kind of thing. income, love life, and children are subjects.

An attitude rule for quizzing Maggie about love life:
she gets angry.

An attitude rule:
if the player wears the top hat, she gets angry.

Now, as we saw, the 'no outcome' result will never be returned and printed as Maggie's reaction, precisely because it is "no outcome". Therefore, we provide a final attitude rule which will give her a default response to all statements:

The last attitude rule:
she is only mildly annoyed.

Test me with "ask maggie about love / ask maggie about income / take off hat / ask maggie about income".

There are plenty of contexts where we might want named outcomes for clarity but not want to print the results literally afterward.

170



Example Ferragamo Again

WI

Using the same phrase to produce different results with different characters.

Here we use phrases that match individual items where possible, and the general kind otherwise:

"Ferragamo Again"

The Break Room is a room. Vanessa, Tina, and Lisa are women in the Break Room. Mark and Holman are men in the Break Room.

Understand the commands "ask" and "tell" and "answer" as something new.

Understand "talk about [any subject]" as talking about. Talking about is an action applying to one visible thing.

Understand "talk about [text]" as talking randomly about. Talking randomly about is an action applying to one topic. Carry out talking randomly about: say "Mostly you're interested in [the list of subjects]."

Carry out talking about something:
now the previous subject is the noun.

Report talking about something:
say "You chat for a while about [the noun]."

A subject is a kind of thing. Assyrian vowel sounds, designer handbags, and instant run-off voting are subjects. Understand "linguistics" and "mute" and "stop" as sounds. Understand "prada" and "tods" and "coach" and "carmen marc valvo" as designer handbags. Understand "reform" and "election" and "election fraud" and "two-party system" and "Diebold" as instant run-off voting.

To say (annoyed-person - a person) gestures in irritation:
say "[The annoyed-person] sighs heavily. [run paragraph on]"

To say (annoyed-person - Vanessa) gestures in irritation:
say "[The annoyed-person] takes off her glasses and polishes them on her sleeve. [run paragraph on]".

To say (annoyed-person - Holman) gestures in irritation:
say "Holman bobs his head. [run paragraph on]"

The previous subject is a subject that varies.

Instead of talking about something for more than one turn:

if the noun is the previous subject, say "[a random visible person who is not the player gestures in irritation]Maybe you should let this one go.[line break] [paragraph break]";

otherwise continue the action.

Test me with "talk about chocolate / talk about vowel sounds / g / talk about handbags / talk about prada / talk about tods".

241



Example **Murder on the Orient Express**

WI

A number of sleuths (the player among them) find themselves aboard the Orient Express, where a murder has taken place, and one of them is apparently the culprit. Naturally they do not agree on whom, but there is physical evidence which may change their minds...

The following example creates two new relations, and two new verbs, in order to set up a tangled web of intrigue.

"Murder on the Orient Express"

The Dining Car is a room. Lord Peter is a man in the Dining Car. Sherlock Holmes is a man in the Dining Car. Miss Marple is a woman in the Dining Car. Adam Dalgliesh is a man in the Dining Car.

Suspecting relates various people to one person.

The verb to suspect means the suspecting relation.

Dalgliesh suspects Holmes. Holmes suspects Lord Peter. Lord Peter suspects Holmes. Miss Marple suspects the player.

Exculpating relates one thing to various people.

The verb to exculpate means the exculpating relation.

The silver bullet exculpates the player. The pipe ash exculpates Holmes. The poison pen letter exculpates Lord Peter. The poison pen letter exculpates Miss Marple. [Poor Dalgliesh. I guess he did it.]

The pipe ash, the letter and the silver bullet are carried.

Given this, we can then set up elaborate rules:

Instead of showing something to a person who suspects the player:

say "'You would say that,' remarks [the second noun] darkly."

Instead of showing something which exculpates the player to someone:

say "'How striking!' says [the second noun]. 'Almost I begin to distrust myself.'"

Test me with "show the letter to miss marple / show the silver bullet to holmes".

And so on: "if Dalglish suspects someone who is exculpated by something carried by the player...", for instance, makes a fitting final example for this chapter. The description

someone who is exculpated by something carried by the player

expresses a complicated idea in very few words, and in such a way that a passer-by looking at the source text would immediately see what was meant.

The moral is that relations allow sophisticated patterns of behaviour to be created in a way that reads back naturally as English.

235



Example The Problem of Edith

WI

A conversation in which the main character tries to build logical connections between what the player is saying now and what went immediately before.

Suppose that we have a core set of issues we want to be able to bring up with all the characters, and we want characters to draw intelligent connections between different conversation topics. We will need some model of how things relate to one another, so:

"The Problem of Edith"

Suggestion relates things to each other. The verb to suggest means the suggestion relation.

A subject is a kind of thing. The current subject is a thing that varies. greeting is a subject.

Understand "ask [someone] about [any subject]" as asking it about the subject.
Understand "tell [someone] about [any subject]" as asking it about the subject.

Asking it about the subject is an action applying to one thing and one visible thing.

Carry out asking it about the subject:

say "'Hmm, [the second noun],' says [the noun]. ";
relate the current subject with the second noun;
now the current subject is the second noun.

And if we wanted to offer the player some hints about angles he could pursue:

Instead of thinking:

say "You contemplate [a list of things suggested by the current subject]."

For that matter, we could use the same system to have characters make sense of any physical evidence the character shows them:

Instead of showing something which suggests the current subject to someone:
say "[The second noun] nods impatiently."

Instead of showing something to someone:
let the next subject be the next step via the suggestion relation from the noun to the current subject;
if the next subject is a subject:
try asking the second noun about the subject the next subject;
otherwise:
say "[The second noun] shrugs."

When play begins:
now the left hand status line is "Discussing: [current subject]";
now the right hand status line is " ".

Broughton Hall is a room. Lady Uckfield is a woman in Broughton Hall. "Lady Uckfield sits at her desk, looking wholly composed."

The nasty letter is a thing carried by the player. The nasty letter suggests infidelity and penmanship. The ten-pound note is carried by the player. It suggests money.

Infidelity is a subject. Infidelity suggests marriage and divorce. Marriage suggests love. Marriage, love, and divorce are subjects.

Penmanship is a subject. Penmanship suggests education. Education is a subject. Class status and money are subjects. Class status suggests education. Money suggests class status and marriage.

The current subject is divorce.

Now we can define what gets said when the subject is changed, regardless of whether the segue was introduced in speech or by a shown object. Since rows are blanked after use, the speaker will never repeat herself; if we provide more than one line about the same pair of topics, the first one will be used, then the second, and so on, until the table runs out:

To relate (initial - a subject) with (next - a subject):
repeat through Table of Remarks:
if the initial is starting entry and the next is the final entry:
say "[comment entry][paragraph break]";
blank out the whole row;
rule succeeds;
say paragraph break.

Table of Remarks

starting	final	comment
divorce	love	"As it seems to me, all the love is on one side,' she says crisply. 'And that rarely works.'"
divorce	love	"Stop making that plea: it won't work."
divorce	infidelity	"Frankly, I rather think there would have been cause enough for divorce without the perversely plentiful evidence of unfaithfulness."

divorce money ""If you mean that the divorce will be expensive, I know it,' she says. 'But I can think of no happier investment.'"
 marriage money ""If you wish me to understand that it was a marriage for money, you could have spared your energy. That was patent from the outset."
 infidelity money ""I'm sorry, but I don't see how having married for money excuses a subsequent infidelity."

If we had more than one character in the scenario, we could provide multiple tables, but this will do to demonstrate the idea.

Of course, we can override specific instances, if we want the character always to say the same thing regardless of how we came to this point:

Instead of asking Lady Uckfield about the subject penmanship:
 now the current subject is penmanship;
 say "She sighs. 'So few people write really beautifully these days.'"

Test me with "think / ask lady about infidelity / show nasty letter to lady / show note to lady / think / ask lady about divorce / ask lady about love / ask lady about marriage / ask lady about divorce / ask lady about love / ask lady about penmanship".

We would have to be careful about this system, since we have applied a various-to-various relation to every single object in the game. In practice it would probably be wisest to restrict it a bit, with judicious definitions of kind and so on.



Example Questionable Revolutions

An expansion on the previous idea, only this time we store information and let characters answer depending on their expertise in a given area.

"Questionable Revolutions"

Interrogative is a kind of value. The interrogatives are who, what, when, where, how, and why.

Current question is an interrogative that varies.

After asking someone about something: respond to the question. After answering someone that something: respond to the question.

After telling someone about something: say "You're here to ask questions."

Country is a kind of value. The countries are Czechoslovakia, Georgia, Sweden, Italy, Spain.

Table of Information

topic	date	place	definition
"velvet revolution"	1989	Czechoslovakia	"A bloodless revolution in Czechoslovakia, in which popular protests led to the resignation of the communist president Gustav Husak, and the election of Vaclav Havel in his place."
"rose revolution"	2003	Georgia	"A revolution in which President Eduard Shevardnadze was interrupted by protesters in the middle of his speech, and forced to flee."
"spanish"	1936	Spain	"An anarchist and socialist movement during the Spanish civil war."

revolution"

After reading a command:

if the player's command includes "[interrogative]", now the current question is the interrogative understood.

To respond to the question:

repeat through the Table of Information:
if the topic understood includes topic entry:
if the current question is what or the current question is who, say definition entry appropriately;
if the current question is when, say date entry appropriately;
if the current question is where, say place entry appropriately;
rule succeeds;
say "[The noun] shrugs."

Understand "ask [someone] [text]" as asking it about.

Comprehension is a kind of value. The comprehensions are vague, erroneous, and correct.

Table of Understanding

character	years	geography	general comprehension
Dr Tweedy	correct	correct	correct
Ms Finch	erroneous	erroneous	correct
Ms Clarion	vague	vague	erroneous

When play begins:

say "Here you are in the first class cabin, but no matter how fancy the seats are, you can still get bored circling over Zurich for three hours on end. To kill time, you and the other passengers are playing a trivia game, and the final topic is your specialty: revolutions."

First Class Cabin is a room. Dr Tweedy is a man in First Class. Ms Finch and Ms Clarion are women in First Class.

To say (year - a number) appropriately:

choose row with character of the noun in the Table of Understanding;
if years entry is correct:
say "'[year],' replies [the noun] promptly.";
increment the quiz score of the noun;
if years entry is erroneous:
let guess be a random number between 1900 and 2005;
say "'[guess]?' guesses [the noun], with an air of diffidence[if guess is the year]. Which is right, as it happens[end if].";
if guess is the year, increment the quiz score of the noun;
if years entry is vague:
let offset be a random number between -5 and 5;
let year be year + offset;
say "'I think [year]. About then. Close, anyway,' replies [the noun][if the offset is 0], getting it right[end if].";
if offset is 0, increment the quiz score of the noun.

To say (spot - a country) appropriately:

choose row with character of the noun in the Table of Understanding;

```

if geography entry is correct:
    say "[spot], ' replies [the noun] promptly.";
    increment the quiz score of the noun;
if geography entry is erroneous:
    let guess be a random country;
    say "'Er... [guess]?' says [the noun][if guess is the spot]. Which is of course
correct[end if].";
    if guess is the spot, increment the quiz score of the noun;
if geography entry is vague, say "'Europe,' replies [the noun] with confidence."

```

```

To say (explanation - some text) appropriately:
    choose row with character of the noun in the Table of Understanding;
if general comprehension entry is correct:
    say "[explanation]"[paragraph break];
    increment the quiz score of the noun;
otherwise:
    choose a random row in the Table of Information;
    say "[definition entry]"[paragraph break];
    if the definition entry is explanation:
        say "[A random other person who is not the noun] looks surprised that
this came out right.";
        increment the quiz score of the noun.

```

A person has a number called quiz score. The quiz score of Tweedy is 48. The quiz score of Finch is 2. The quiz score of Clarion is 4.

Definition: a person is other if it is not the player.

```

When play begins:
    now left hand status line is "T: [quiz score of Tweedy] F: [quiz score of Finch]
C: [quiz score of Clarion]";
    now right hand status line is "[time of day]".

```

Test me with "dr tweedy, where was the velvet revolution located / ms finch, when was the rose revolution / ms finch, what was the rose revolution / ms clarion, when was the spanish revolution / g / g / ms finch, when was the spanish revolution".

We have so far seen several ways to write conversational characters in Inform, and we will see more before the end of the manual. This naturally raises the question, which should we use? To which the answer is: it depends on the sort of game we're writing, and what we want our characters to do. The more rich and complex the system, the more likely that it will require a lot of content; if we add question types as well as keywords, for instance, we instantly multiply the number of responses we have to write by five or six. It is not worth doing this unless there is some corresponding advantage within the game.



Suppose we want the player to ask questions of slightly more complexity - we might want to build in a system that understood "who", "what", "where", and "when", for instance. We could use a topic table for this, too:

"The Queen of Sheba"

Interrogative is a kind of value. The interrogatives are who, what, when, where, how, and why.

Current question is an interrogative that varies.

After asking someone about something: respond to the question. After answering someone that something: respond to the question.

After telling someone about something: say "You're here to ask questions and test Solomon's wisdom, not to give him a sample of your own."

Table of Wise Answers

topic	question type	reply
"rain/weather/clouds/cloud/rains"	what	"Clouds are a disturbance made by the paths of birds,' Solomon replies. 'The air beaten by their wings becomes agitated, as when a river is stirred and the mud churns up.'"
"rain/weather/clouds/cloud/rains"	where	"Weather is contained in a great silk bag which holds in the heavens,' replies Solomon."
"hunger/food/eating"	when	"Sorry, are you getting hungry?' he says, and rings a bell to summon servants."
"hunger/food/eating"	why	"Men were made to need food in order that they must farm and cook and dine together,' Solomon replies. 'Otherwise, they might live apart, each sufficient in himself. But no man can feed himself alone all through his life.'"
"Solomon/he/himself"	who	"As you see,' he says, holding out his arms to each side."
"Solomon/he/himself"	what	"I am an ordinary man,' he answers."

One of the nice things about this system is that it only resets the "current question" when we get a new question word. For instance, this test will produce different replies to the question about Solomon himself, because the second time he is still in the mode of answering "what" questions:

Test me with "ask solomon about himself / ask solomon what rain is / ask solomon about himself".

If Solomon is to live up to his reputation at all, his wisdom table will have to be quite a bit longer - though one also would want to be careful, because forcing the game to cycle through a really immense table could be quite time-consuming. In fact, for the sake of this example, let's reward the player for managing to stay within the (narrow) range of Solomon's knowledge:

The Hall of Almug Tree Pillars is a room. "The pillars of the room are made of almug tree, the ceiling made of silk and the floor of glass." Solomon is a man in the Hall of Almug Tree Pillars. Solomon has a number called wisdom. The wisdom of Solomon is 0.

Every turn:
if the wisdom of Solomon is 3:

```
say "Truly, Solomon has answered all your questions, and his wisdom is
even as great as you had heard!";
end the story saying "Your heart beats strangely fast".
```

When play begins, say "'Oh, you've arrived,' says Solomon."

In a real game we'd need to be a great deal subtler. All the same, if we have a character of quite limited resources to present to the player, it's a good idea to give the player some incentive to stay on topic, ask questions the character can answer, and generally interact within the parameters we're prepared for.

Now, this last bit requires some trickery from later chapters, particularly those on Understanding and Activities, to pull the question words out of the player's command:

```
After reading a command:
  if the player's command includes "[interrogative]", now the current question is
  the interrogative understood.
```

```
To respond to the question:
  repeat through the Table of Wise Answers:
    if the topic understood includes topic entry:
      if the current question is the question type entry:
        say "[reply entry][paragraph break]";
        increment the wisdom of Solomon;
        rule succeeds;
      say "Solomon looks blank, appalled by a question for which he was not
      prepared.";
    end the story saying "You have befuddled Solomon!"
```

Understand "ask [someone] [text]" as asking it about.

And now we have a game that will accept (though not always respond very sensibly to) questions of almost any form we might put to another character: ASK SOLOMON WHAT RAIN IS will be answered, but then again, it won't be distinguished from, say, ASK SOLOMON WHETHER THIS PERSISTENT RAIN IS A DIVINE PUNISHMENT OR WHAT.

All the same, a system that allowed the player a bit more specification of questions than simple keyword-use might be useful in a mystery game, for instance, where we might want to let our detective conduct inquiries into specific details. An alternative approach to the rather free one above would be to force the player to use only questions of the form WHAT IS RAIN? or WHO ARE YOU?: this would cut down on false-positive matches. But we might still choose to store the responses in a table of this type.



Suppose we have a conversation system in which it is important to keep track of which subjects the player has heard mentioned. If we're careful to mark subjects in brackets, we can use the "printing the name of" activity to record which things have been mentioned so far:

"Chronic Hinting Syndrome"

A subject is a kind of thing.

Knowledge relates various people to various subjects. The verb to know means the knowledge relation.

Awareness relates various people to various subjects. The verb to be aware of means the awareness relation.

Definition: a subject is pending if the player is aware of it and it is not known by the player.

Instead of thinking:

if the number of pending subjects is 0, say "You have no fresh leads at the moment.";

otherwise say "You recall that thus far you have not followed up with questions about [the list of pending subjects]."

After printing the name of a subject (called idea):
now the player is aware of the idea.

Now suppose that as an added convenience for the player, we let him turn on a mode in which useful conversation topics are always automatically highlighted in the text, so he doesn't waste his time trying to follow up dead leads:

Setting is a kind of value. The settings are bright and dull. Understand "on" as bright. Understand "off" as dull.

Highlighting is a setting that varies. Highlighting is dull.

Understand "highlighting [setting]" as setting highlighting. Setting highlighting is an action out of world, applying to one setting.

Carry out setting highlighting:

now highlighting is the setting understood.

Report setting highlighting:

say "Highlighting is now [if highlighting is dull]off[otherwise]on[end if]."

Before printing the name of a subject (called idea) when highlighting is bright:
unless the player knows the idea:

say "[bold type]".

After printing the name of a subject when highlighting is bright:

say "[roman type]".

...And the rest is peripheral.

The Sickbay is a room. "A place arranged for Nathan's comfort, since his sickness has been prolonged and because he becomes so irritating when not comfortable." The Hallway is outside from the Sickbay.

A supporter can be untried or rejected. A supporter is usually untried.

The Sickbay contains a wobbly pedestal, a table, and a sickbed. Understand "bed" as the sickbed. The pedestal, the table, and the sickbed are supporters. Nathan is a man on the sickbed. The sickbed is scenery. The initial appearance of the wobbly pedestal is "A wobbly pedestal near the door has sometimes been known to support vases of flowers, but is currently bare." The initial appearance of the table is "There is also a table of a more ordinary sort."

Nathan can be active or passive.

After printing the name of Nathan: now Nathan is passive.

Instead of putting the sculpture on the table:
now the table is rejected;
say "[Not there], [Nathan] snaps. 'The table is way too far from the sickbed.'"

Instead of putting the sculpture on the sickbed:
now the sickbed is rejected;
say "[Not there], [Nathan] rebukes you. 'You don't want me knocking it over if I roll around. In pain.'"

Instead of putting the sculpture on the pedestal:
now the pedestal is rejected;
say "The pedestal starts to wobble so ominously that you don't dare let go.

'[Not there], says [Nathan]. 'That thing is falling apart.'"

Before putting something on the down: try dropping the noun instead.

To say not there:
if all the supporters are rejected:
say "Look, the floor would be fine";
otherwise if the number of rejected supporters is 1:
say "Yeah, anywhere but there, I'm afraid";
otherwise:
say "Come on, use your head -- I can't be watching you all the time, I'm sick".

Instead of going outside when the player is carrying the sculpture:
say "You've still got this sculpture to get rid of."

Instead of going outside when the breakage is pending:
say "You can't very well smash in front of [Nathan] his prize sculpture and then just scamper off without saying something. Appealing though the thought is at the moment."

Instead of going outside when a subject which is not the breakage is pending:
say "'Yeah, go ahead,' says [Nathan], with a martyr-like air. 'It's probably best that you don't hear about [the random pending subject]. It's not something I'd go into normally.'"

The breakage is a subject. The description is "It's not a big deal. I'll just buy a new [mental wave generator].' A slight awkward pause. 'I mean, this one was a [gift], but don't worry about it". Understand "accident" or "smashing" or "breaking" or "shard" or "mishap" or "shards" or "mistake" as the breakage. Understand "sculpture" as the breakage when the player is not carrying the sculpture.

Instead of saying sorry in the presence of Nathan when the player is aware of the breakage:

try asking Nathan about the matter of breakage.

Instead of asking Nathan to try saying sorry when the player is aware of the breakage:

try asking Nathan about the matter of breakage.

The mental wave generator is a subject. The description is "They're kind of expensive but I can save up. I really need one, though, because of my [dreams]".

The dreams is a subject. The description is "They're not the kind of dream you want to have.' He closes his eyes and contemplates these undesirable dreams. 'Have you ever woken up convinced you were dead? No, probably not. Well... At any rate, I need the [generator]. Oh, don't worry, they're expensive but not so expensive that I won't be able to save up for another, in a few months".

The gift is a subject. The description is "[The mental wave generator] was a present from a girl named [Shari]. Actually I'm not sure she'd take to being called a girl".

Shari is a subject. The description is "Look, let's just not go into it, okay? I don't really want to relive all that right now. I still have a six-inch [scar] shaped like a banana in the middle of my back".

The scar is a subject.

Instead of asking Nathan about the matter of the scar:

say "Nathan clears his throat, lowers his voice, and begins to tell you the story...";

end the story saying "End of Demo -- Register to Continue!!"

Understand "ask [someone] about [any subject]" as asking it about the matter of.

Asking it about the matter of is an action applying to one thing and one visible thing.

Check asking it about the matter of:

if the player is not aware of the second noun, say "What [second noun]?" instead;

if the noun does not know the second noun, say "'I've no idea,' replies [the noun]." instead;

if the player knows the second noun, say "You've already covered that. The response was [description of the second noun]." instead.

Carry out asking it about the matter of:
now the player knows the second noun.

Report asking it about the matter of:
say "[description of the second noun],' says [the noun]."

Instead of telling Nathan about something:
say "He pinches the bridge of his nose. 'I can't really follow this right now,' he says. 'I'm sorry.'"

Instead of asking Nathan about something:
say "He shrugs weakly."

When play begins:
say "'Just put that down anywhere,' says [Nathan], as you come into the room. He's sitting in the sickbed with his legs straight out in front of him. 'I don't care where.' His voice is weak, but it sharpens up for the last remark: 'And don't make a lot of noise about it.'"

Considering that he woke you from a sound slumber to beg you to bring this thing over, his attitude is a bit much. You stare dubiously at the awkward crystal sculpture in your hands."
now Nathan knows every subject.

Instead of asking Nathan about something while the player carries the sculpture, say "[Nathan] moans dramatically and refuses to be drawn into conversation."

The player is carrying an awkward crystal sculpture. Understand "objet" or "objet de hideous" as the sculpture. The description of the sculpture is "It might possibly be natural, or it might be man-made. It might appeal to someone, but it is certainly not to your tastes."

Instead of showing the sculpture to Nathan:
say "'Please put it anywhere,' he says."

Instead of giving the sculpture to Nathan:
say "'No, it doesn't work if I touch it. That's why I couldn't-- well, just put it down.'"

After dropping the sculpture:
now the player is aware of the breakage;
now the sculpture is nowhere;
say "You are incredibly careful, but somehow the sculpture slips -- you might almost say slithers -- from your fingers and crashes into a thousand shards at the feet of [Nathan]."

There is a tense silence."

Before reading a command: now Nathan is active.

Every turn while not asking:
 if Nathan is passive, rule succeeds;
 if the player is carrying the sculpture:
 if showing or giving, rule succeeds;
 say "[Nathan] opens one eye and stares at you meaningfully. He is waiting for you to deposit his objet de hideous somewhere.";
 rule succeeds;
 if the breakage is pending:
 if dropping, rule succeeds;
 say "You're not quite sure where to begin, but you can't very well leave without making at least some remark on the smashing of the sculpture.";
 rule succeeds;
 if a subject is pending:
 choose a random row in the Table of Offhand Reminiscences;
 say "[line entry][paragraph break]".

Table of Offhand Reminiscences

line

"It actually is kind of a funny story about [the random pending subject],' [Nathan] remarks casually."

"[Nathan] chuckles under his breath. 'Man, I hadn't thought about [the random pending subject] in ages.'"

"He glances sideways at you. 'It's nothing personal, you know, but I don't feel comfortable discussing [the random pending subject] with just anyone.'"

"I don't know why I brought up [the random pending subject] just now,' [Nathan] comments. 'Don't mention it to anyone, if you don't mind.'"

"Okay, see, the thing about [the random pending subject] is...' [paragraph break]'Yes?' you ask, on cue.[paragraph break]...never mind."

"[Nathan] makes an explosive exasperated sound. 'Don't you want to ask me about [the random pending subject]?' he demands."

Test me with "highlighting bright / put sculpture on pedestal / put it on table / put it on sickbed / drop it / think / ask nathan about breakage / think / ask him about generator / ask him about dreams / ask him about gift / ask him about shari / ask him about scar".

426

🔦 Example Robo 1

W1

A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.

"Robo"

The Experimentation Chamber is a room. Robo is a man in the Experimentation Chamber. "Robo, your prototype tin companion, stands awkwardly beside you. In the middle of his chest is a red enamel button[if the red button is switched on], currently depressed[otherwise], currently un-depressed[end if]."

The red button is a device. It is part of Robo. Instead of pushing the red button: if the red button is switched off, try switching on the red button; otherwise try switching off the red button.

After switching on the red button:
say "CLICK! Robo is now in play-back mode."

After switching off the red button:
say "CLACK! Robo is now in observation mode."

Definition: Robo is watching if the red button is switched off.

The current instruction set is a list of stored actions that varies.

After doing something when Robo is watching and Robo can see the player:
now the actor is Robo;
add the current action to the current instruction set;
now the actor is the player;
say "Robo watches you [the current action][one of], his yellow eyes lamp-like and observant[or]. In his metal head, gears whirr[or], his brushed-copper handlebar moustaches twitching[or] impassively[at random].";
continue the action.

Every turn when Robo is not watching:
if the number of entries in the current instruction set is 0:
say "Robo has run out of behavior and grinds to an (expectant) halt.";
now the red button is switched off;
otherwise:
let the next task be entry 1 of the current instruction set;
try the next task;
remove entry 1 from the current instruction set.

The red block and the blue cylinder are things in the Experimentation Chamber. The counter is a supporter in the Experimentation Chamber. The counter is scenery.

Report Robo examining Robo:
say "Robo examines each of his hands in turn, then each of his legs (bending over mostly double in the middle to do this)." instead.

Report Robo examining the player:
say "Robo stares at you, unblinkingly, for several seconds together[if a random chance of 1 in 7 succeeds]. His left moustache-bar twitches infinitesimally upward[end if]." instead.

Report Robo taking the cylinder:
say "[one of][Robo] needs several attempts to get his metal fingers around [the cylinder] -- they are not designed for grasping small objects elegantly. But at last he succeeds[or]Once again, Robo struggles a bit before picking up [the cylinder][stopping]." instead.

Test me with "z / take cylinder / take block / put cylinder on counter / put block on counter / x robo / x me / get block / drop block / press red button / z / z / z / z / z / z / z / z / z / z / z".

Your hard-working mother uses a list as a stack: urgent tasks are added to the end of the list, interrupting longer-term plans.

Suppose the player's mother is supposed to be cleaning the living room, but she can be interrupted by the need to pick up things the player has dropped. New tasks are added to the end of her "current plan" list; every turn, she attempts to do whatever is the last entry on that list.

"Your Mother Doesn't Work Here"

A person has a list of stored actions called the current plan.

Every turn:

- repeat with culprit running through people who are not the player:
 - if the number of entries in current plan of the culprit is greater than 0:
 - let N be the number of entries in the current plan of the culprit;
 - try entry N of the current plan of the culprit;
 - remove entry N from the current plan of the culprit.

The Living Room is a room. It contains a somewhat muddy Persian rug. Your mother is a woman in the Living Room.

West of the Living Room is the Kitchen.

Instead of your mother rubbing the rug:

- say "Your mother scrubs the stained areas of the rug, muttering to herself."

Instead of taking something:

- say "Nah, Mom'll get that."

Report your mother taking something:

- say "Your mother picks up [the noun][one of], sighing deeply[or], jaw tight[or], with assorted comments on your manners[or]; to judge from her comments, she is also indulging in a pleasant fantasy about Swiss boarding schools[stopping]." instead.

When play begins:

- add mother going west to the current plan of mother;
- add mother rubbing the rug to the current plan of mother.

Every turn:

- if mother is not in the Living Room, end the story finally.

Carry out dropping something:

- add mother taking the noun to the current plan of mother.

The player carries some dirty socks, some dirty shoes, a dirty hat, a pair of dirty trousers, and a backpack.

Test me with "drop socks / z / drop shoes / drop hat / drop all / z / z".

As goal-seeking goes, this is fairly rudimentary; "Boston Cream" provides an alternative (and slightly more sophisticated approach), but for really complex goal-seeking characters, it is probably best to turn to the character extensions designed for Inform.

161



Example Pine 2

WI

Pine: Adding a conversation with the princess, in which a basic set of facts must be covered before the scene is allowed to end.

"Pine"

A person can be asleep or awake. A person can be active or passive.

The Spinning Tower is a room. "A remote corner of the old castle, reserved for spinning and weaving tasks."

Sleeping Beauty is an asleep woman in the Spinning Tower. "[if asleep]Sleeping Beauty lies here, oblivious to your presence[otherwise]Sleeping Beauty stands beside you, looking a little confused[end if]." The description is "She is even more magnificent than the rumors suggested." Understand "woman" or "girl" or "princess" or "lady" as Sleeping Beauty.

Discovery is a scene. Discovery begins when play begins. Discovery ends when Sleeping Beauty is awake. Marriage Proposal is a scene. Marriage Proposal begins when Discovery ends.

When Discovery ends: say "Throughout the palace you can hear the other sounds of stirring and movement as the spell of centuries is broken."

Instead of waking an awake person: say "Redundant."

Instead of waking an asleep person: say "Yes, but how?"

Instead of attacking an asleep person:

now the noun is awake;
say "[The noun] sits bolt upright. 'Hey! Ow!' So much for that true love's kiss nonsense."

Instead of kissing an asleep person:

now the noun is awake;
say "[The noun] slowly stirs to wakefulness!"

Instead of throwing water at an asleep person:

now the second noun is awake;
now the noun is nowhere;
say "You pour out [the noun] on [the second noun]."

[The second noun] wakes, shuddering. 'Agh! I had a terrible dream about drowning and then-- Hey!'"

The player carries a jug of water. Understand "pour [something] on [something]" or "splash [something] at/on [something]" as throwing it at.

So much, we had before. Now, suppose we want a conversation style which allows the player to move conversation forward by asking appropriate questions, but which will keep moving forward even if he doesn't. To this end, we provide a table -- a borrowing from a later chapter. In the table, we record two ways of performing each conversation bit, one which reflects the player's participation, and one in which the character moves things onward:

Table of Conversation

topic	reply	quip
"dream/dreams/nightmare/nightmares/sleep"	"Sleep well?"	you ask solicitously.
'Not really,' she replies, edging away from you. So much for that angle."	"Ghastly nightmares,' she remarks. You nod politely."	
"marriage/love/wedding/boyfriend/beau/lover"	"So,' you say. 'This is a little weird since we just met, but, um. Would you like to get married?'"	
She looks at you nervously. 'Do I have to?'"	"I, er,' she says. 'I hope I'm not supposed to marry you or something.'"	
"marriage/love/wedding/boyfriend/beau/lover"	"I was told I was going to marry you and inherit the kingdom,' you say, apologetically. 'Would that be very bad?'"	

'Oh, it's not you -- I'm seeing someone,' she says, smiling quickly.

You try to think how to point out that it's been a hundred years since she last saw her boyfriend."	"Do you think I could go look for someone? I'm seeing him, you see, and I think I've been... sick... for a while, so he might be worried.'
---	--

You try to think how to point out that it's been a hundred years since she last saw her boyfriend."

"marriage/love/wedding/boyfriend/beau/lover"	"You've been up here for a hundred years,' you say. An unpleasant thought occurs to you. 'Was your young man in the castle somewhere?'"
She shakes her head mutely."	"She goes to the window and looks out at the now-fading thicket of briar. 'That took a while to grow,' she observes. 'I've been up here longer than I thought.'"

You shrug, uncomfortable."

Instead of asking an awake beauty about a topic listed in the Table of Conversation:

now Beauty is passive;
say "[reply entry][paragraph break]";
blank out the whole row.

The "now Beauty is passive" line prevents her from making any conversation of her own on a turn when we've spoken to her. This keeps the conversation from progressing too quickly.

Instead of telling an awake beauty about something: try asking the noun about it.

Instead of asking an asleep person about something:
say "[The noun] snores."

Marriage Proposal ends when the number of filled rows in the Table of Conversation is 0.

Every turn during Marriage Proposal:
if Beauty is active:
repeat through Table of Conversation:
say "[quip entry][paragraph break]";
blank out the whole row;
make no decision.

After we've generated any spontaneous conversation, we return her to her regular active state.

Every turn: now Beauty is active.

When Marriage Proposal ends: end the story saying "This is going to take some explaining."

Test me with "x beauty / wake beauty / pour water on beauty / ask beauty about sleep / z / ask beauty about marriage".

Now we have a scenario in which the player can ask her some questions out of order if he really wants to, but the scene will not end until the basic conversation topics have been exhausted. If we wanted to add some other chit-chat, not as part of the main conversation strand, but by way of optional enrichment, we might make a second conversation table and record alternative outcomes in it.

429



Example Robo 2

WI

A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.

We have seen how we can make a robot that watches the player, then plays back the same actions again. A slightly more adventurous implementation would be to let the player create a whole series of named scripts which the robot will run on command.

To do this, we'll need each program to have a command that sets it off (stored as text, since this is the best way to capture and preserve arbitrary text entered by the player) and then the script of actions that must result:

"Robo 2"

Use dynamic memory allocation of at least 16384.

Chapter 1 - Programming

Section 1 - The Programs Themselves

The hard drive is a container. A program is a kind of thing. 15 programs are in hard drive. A program has some text called the starter command. A program has a list of stored actions called the script. Understand the starter command property as describing a program.

Rule for printing the name of a program (called the target) which is not blank:
say "[starter command of the target in upper case]".

Definition: a program is blank if the number of entries in its script is 0.

The current instruction name is some text that varies. The current instruction set is a list of stored actions that varies.

Now, we want to let Robo learn new programs; for this purpose, we'll emulate the code from our previous implementation, so that Robo watches what the player does and stores those actions in his script:

Section 2 - Learning New Programs

Understand "learn [text]" as learning. Learning is an action applying to one topic.

Check learning:

say "You have already learned all you need to know. Robo, however, remains to be trained." instead.

Check Robo learning:

if Robo is watching, say "Robo is already recording '[the current instruction name]'." instead.

Carry out Robo learning:

truncate the current instruction set to 0 entries;
now the current instruction name is the topic understood;
now Robo is watching.

Report Robo learning:

say "'Learning [the current instruction name in upper case],' Robo replies."

After doing something when Robo is watching and Robo can see the player:

now the actor is Robo;
add the current action to the current instruction set;
now the actor is the player;
say "Robo watches you [the current action][one of], his yellow eyes lamp-like and observant[or]. In his metal head, gears whirr[or], his brushed-copper handlebar moustaches twitching[or] impassively[at random].";
continue the action.

Of course, we also need to be able to switch learning mode off, and store any script learned this way. We'll also use the same STOP command to make Robo terminate a program he's in the middle of running.

Section 3 - Returning to Standby Mode

Understand "stop" as stopping. Stopping is an action applying to nothing.

Check stopping:
say "The command is useful only for Robo." instead.

Check Robo stopping:
if Robo is standing by, stop the action.

Carry out Robo stopping when Robo is watching:
let N be a random blank program;
if N is a program:
now the starter command of N is the current instruction name;
now the script of N is the current instruction set;
say "'Stored [current instruction name in upper case].";
otherwise:
say "FAILURE: no program slots remaining."

Carry out Robo stopping:
now Robo is standing by.

Report Robo stopping:
say "Robo is now standing by."

Next, we need to be able to play these programs back again. We'll give Robo a "current program" to store which program he's currently working on, and a number called "stage" which will record where he is in the script. Our previous implementation simply had Robo erase entries from his script list as he performed them, but this time we would like Robo to be able to remember and rerun the same scripts over and over, so we need something a little more subtle.

Section 4 - Running Learned Programs

Understand "run [any program]" as running. Running is an action applying to one visible thing.

Check running:
say "Only Robo can perform Robo's programs." instead.

Check Robo running:
if Robo is not standing by, stop the action.

Unsuccessful attempt by Robo running:
say "'ERROR: Robo can launch new programs only when on standby.'"

Carry out Robo running:
now the current program of Robo is the noun;
now the stage of Robo is 1;
now Robo is replaying.

Report Robo running:
say "'Running [the starter command in upper case], ' Robo confirms.'"

Every turn when Robo is replaying:
let the chosen script be the script of the current program of Robo;
let maximum be the number of entries in the chosen script;
let N be the stage of Robo;

let the next step be entry N of the chosen script;
try the next step;
increment the stage of Robo;
if the stage of Robo is greater than the maximum:
 say "Robo's program ends, and he reverts to stand-by mode.";
 now Robo is standing by;
 now the stage of Robo is 1.

For the player's sanity, we should also provide a way to find out which programs Robo has stored in memory and what they do, so we design two listing commands:

Section 5 - Listing Learned Programs

Understand "list programs" as requesting program list. Requesting program list is an action applying to nothing.

Check requesting program list:
 say "You will have to ask Robo to list programs." instead.

Carry out Robo requesting program list:
 say ""The available program[if more than one program is not blank]s[end if] [is-are list of programs which are not blank]."".

Understand "describe [any program]" or "list [any program]" as requesting script of. Requesting script of is an action applying to one visible thing.

Check requesting script of:
 say "You will have to ask Robo to give you the script." instead.

Carry out Robo requesting script of:
 say "The script of [noun] is: [script of the noun]."

And to complete the suite, in case the player runs into Robo's fifteen-program limit:

Section 6 - Deleting Learned Programs

Understand "delete [any program]" as deleting. Deleting is an action applying to one visible thing. Understand the command "erase" as "delete".

Check deleting:
 say "You will have to instruct Robo to delete [the noun]." instead.

Check Robo deleting (this is the can't delete except in standby rule):
 if Robo is not standing by, stop the action.

Unsuccessful attempt by Robo deleting:
 say ""ERROR: programs may only be deleted while Robo is in stand-by mode."" instead.

Carry out Robo deleting:
 truncate the script of the noun to 0 entries;
 now the starter command of the noun is "".

Report Robo deleting:
say ""Program deleted.""

Now we use pretty much the same set-up as before to test Robo's abilities:

Chapter 2 - The Scenario

The Experimentation Chamber is a room.

Robo is a man in the Experimentation Chamber. "Robo, your prototype tin companion, stands awkwardly beside you[if watching], watching[end if]." Robo can be watching, replaying, or standing by. Robo is standing by. Robo has a program called the current program. Robo has a number called the stage.

Persuasion rule: persuasion succeeds.

The red block and the blue cylinder are things in the Experimentation Chamber. The counter is a supporter in the Experimentation Chamber. The counter is scenery.

Report Robo examining Robo:
say "Robo examines each of his hands in turn, then each of his legs (bending over mostly double in the middle to do this)." instead.

Report Robo examining the player:
say "Robo stares at you, unblinkingly, for several seconds together[if a random chance of 1 in 7 succeeds]. His left moustache-bar twitches infinitesimally upward[end if]." instead.

Report Robo taking the cylinder:
say "[one of][Robo] needs several attempts to get his metal fingers around [the cylinder] -- they are not designed for grasping small objects elegantly. But at last he succeeds[or]Once again, Robo struggles a bit before picking up [the cylinder][stopping]." instead.

Test me with "test chocolate / test vanilla".

Test chocolate with "learn chocolate / stop / list programs / Robo, learn chocolate / take red / put all on counter / Robo, stop / Robo, list programs / Robo, run chocolate / z / Robo, run chocolate / Robo, stop / z".

Test vanilla with "Robo, learn vanilla / take all / i / drop all / x robo / x me / Robo, stop / Robo, list programs / Robo, list vanilla / Robo, run vanilla / z / z / robo, delete vanilla / robo, stop / robo, list vanilla / robo, delete vanilla / robo, list programs".

We could also have written this so that Robo learns new scripts by accepting the player's instructions, so that

```
>ROBO, LEARN LIBRARY THEFT
>ROBO, TAKE BOOK
>ROBO, EAST
>ROBO, STOP
```

...would store the script 'library theft' with the actions taking the book and going east. The implementation there would have been mostly identical, except that instead of an "after doing something..." rule, we would have captured commands as we asked Robo to perform them, and added those to the command list in progress. The alternative code might look something like this:

Before Robo doing something other than stopping when Robo is watching:
add the current action to the current instruction set;
say "'CHECK: [current action] added to script,' says Robo." instead.

Unsuccessful attempt by Robo doing something when Robo is watching:
say "He does not actually perform the action."

77

★ Example Mistress of Animals

WI

A person who moves randomly between rooms of the map.

Suppose we want a restless sort of character always pacing from room to room. It is quite easy to use adjacency to achieve this effect:

"Mistress of Animals"

Corinth is a room. Athens is east of Corinth. Epidaurus is southeast of Corinth and east of Mycenae. Mycenae is south of Corinth. Olympia is west of Mycenae. Argos is south of Mycenae. Thebes is northwest of Athens. Pylos is south of Olympia. Sparta is east of Pylos and south of Argos. Delphi is northwest of Thebes.

Artemis is a woman in Thebes.

Every turn:

if Artemis is in a room (called the current space):
let next space be a random room which is adjacent to the current space;
if Artemis is visible, say "Artemis heads to [the next space].";
move Artemis to next space;
if Artemis is visible, say "Artemis arrives from [the current space]."

Test me with "z / z / z / z / z / z".

Of course, it helps that Artemis is the sort to like open spaces. The implementation would become more complicated if there were doors which might block transit between these locations.

39

★ Example Van Helsing

WI

A character who approaches the player, then follows him from room to room.

Suppose we want to write a character who tries to be in the same room as the player. We will do this by testing every turn whether the character's location and the player's location are the same; if the answer is no, the character will look for a path to the player's location, then try to move along that path. (We will learn more about finding paths and giving characters instructions later.)

The result will be that if the player ever moves to another location, the character will automatically pursue him.

"Van Helsing"

The Drawbridge is a room. North of the Drawbridge is the Immensely Enormous Entry Hall. West of the Entry Hall is the Vast Dining Area. North of the Vast Dining Area is the Colossal Kitchen. The Spooky Guano-filled Attic is above the Entry Hall.

Count Dracula is a man in the Attic.

In the following condition, we could also have written "if the location of Count Dracula is not the location", because "location" by itself is always understood to be the player's location. But it seemed better for clarity to write it this way:

Every turn:

if the location of Count Dracula is not the location of the player:
let the way be the best route from the location of Count Dracula to the location of the player, using doors;
try Count Dracula going the way;
otherwise:
say "'Muhahaha,' says Count Dracula."

Test me with "z / z / n / w / e / u / z / d".

274



Example Odyssey

WI

A person who follows a path predetermined and stored in a table, and who can be delayed if the player tries to interact with her.

"Odyssey"

Corinth is a room. Athens is east of Corinth. Epidaurus is southeast of Corinth and east of Mycenae. Mycenae is south of Corinth. Olympia is west of Mycenae. Argos is south of Mycenae. Thebes is northwest of Athens. Pylos is south of Olympia. Sparta is east of Pylos and south of Argos. Delphi is northwest of Thebes.

Athena is a woman in Athens.

Athena will proceed, unless delayed, through a list of locations stored in a simple table. Rather than using Inform's route-finding abilities ("the best route from..."), we

simply move Athena from one location to the next, not even using the going action: she moves in mysterious ways, as befits a goddess.

Table of Athena's Movement

destination

Thebes

Delphi

Thebes

Athens

Corinth

Mycenae

Every turn when Athena is active:

repeat through the Table of Athena's Movement:

let last space be the location of Athena;

if Athena can be seen by the player, say "Athena heads to [the destination entry].";

move Athena to destination entry;

if Athena can be seen by the player, say "Athena arrives from [the last space].";

blank out the whole row;

break.

By blanking out the table line by line, we make sure that we never lose our place in the path.

Since we want the player to be able to talk to Athena, we need a way to stall her in her path, as well.

Athena can be active or passive. Athena is active.

Before doing something to Athena:

now Athena is passive;

say "Athena waits around patiently, though you can tell she would like to leave..."

Instead of telling Athena about something:

say "She watches you patiently as though to say that she already knows."

Instead of asking Athena about something:

say "Her response is inscrutably ancient and Greek. Afterwards you remember only the flash of bright eyes."

Finally, we do need to wake Athena up again if she has become passive. The following rule will occur after the movement rule just because of code ordering, though we could make matters more explicit if we needed to:

Every turn when Athena is passive:

now Athena is active.

Test me with "east / northwest / wait / examine athena / wait".



Example Actaeon

A FOLLOW command allowing the player to pursue a person who has just left the room.

Suppose we want the player to be able to go after characters who are moving around the map. The trick, of course, is that once characters are gone they are no longer visible to "follow [person]", so we need "follow [any person]" to find them.

"Actaeon"

A person has a room called last location.

Understand "follow [any person]" as following. Understand the commands "chase" and "pursue" as "follow".

Following is an action applying to one visible thing.

Check following:

if the noun is the player, say "Wherever you go, there you are." instead;
 if the noun is visible, say "[The noun] is right here." instead;
 if the last location of the noun is not the location, say "It's not clear where [the noun] has gone." instead.

Here again the best route comes in handy:

Carry out following:

let the destination be the location of the noun;
 if the destination is not a room, say "[The noun] isn't anywhere you can follow." instead;
 let aim be the best route from the location to the destination;
 say "(heading [aim])[line break]";
 try going aim.

Corinth is a room. Athens is east of Corinth. Epidaurus is southeast of Corinth and east of Mycenae. Mycenae is south of Corinth. Olympia is west of Mycenae. Argos is south of Mycenae. Thebes is northwest of Athens. Pylos is south of Olympia. Sparta is east of Pylos and south of Argos. Delphi is northwest of Thebes.

Artemis is a woman in Corinth.

We do also have to make sure that whenever we move a person from room to room, we record where they were moved from; otherwise, our clever restrictions about whom the player can pursue will not work properly.

To move (pawn - a person) tidily to (target - a room):

now the last location of the pawn is the holder of the pawn;
 move the pawn to the target.

Every turn:

let current location be the location of Artemis;
 let next location be a random room which is adjacent to the current location;

if Artemis is visible, say "Artemis heads to [the next location].";
move Artemis tidily to next location;
if Artemis is visible, say "Artemis arrives from [the current location]."

Test me with "wait / follow artemis / follow artemis / follow artemis".

185



Example **Latris Theon**

WI

A person who can accept instructions to go to new destinations and move towards them according to the most reasonable path.

To begin with, we create an action for going to a named place. All that this action will do is to change that person's hoped-for destination: the actual moving around comes later.

"Latris Theon"

A person has a room called destination.

Understand "go to [any room]" as going vaguely.

Going vaguely is an action applying to one visible thing.

Carry out someone going vaguely:
now the destination of the person asked is the noun.

Report someone going vaguely:
say "[The person asked] looks amused, but accepts the commission to go to [the noun]."

It stands to reason the player plays Zeus or at the very least Apollo, but let's not let this go to the player's head. Note that the following rule applies to the player, but not to anyone else, so HERMES, GO TO ATHENS will work but GO TO ATHENS will not.

Carry out going vaguely:
say "You're too thoroughly lost."

And finally we recreate Greece and one of its heroes.

Corinth is a room. Athens is east of Corinth. Epidaurus is southeast of Corinth and east of Mycenae. Mycenae is south of Corinth. Olympia is west of Mycenae. Argos is south of Mycenae. Thebes is northwest of Athens. Pylos is south of Olympia. Sparta is east of Pylos and south of Argos. Delphi is northwest of Thebes.

Hermes is a man in Corinth. The destination of Hermes is Corinth. [So he is initially happy where he is.] Persuasion rule for asking Hermes to try going vaguely: persuasion succeeds. [But he is open to suggestions.]

Every turn when the destination of Hermes is not the location of Hermes:
let the right direction be the best route from the location of Hermes to the destination of Hermes;
try Hermes going the right direction.

Test me with "hermes, go to athens / e".

It simplifies matters that our map of Greece makes it possible to reach any location from any other location, by some sequence of movements: if there were an isolated location -- say, Crete -- with no map connection to the mainland, then we would have to worry about the "right direction" not being a direction at all. The following version of Hermes' trekking rule is protected against the possibility:

Every turn when the destination of Hermes is not the location of Hermes:
let the right direction be the best route from the location of Hermes to the destination of Hermes;
if the right direction is a direction, try Hermes going the right direction.

402



Example Patient Zero

WI

People who wander around the map performing various errands, and in the process spread a disease which only the player can eradicate.

"Patient Zero"

Use the serial comma and no scoring.

Understand "about" as asking for information. Asking for information is an action out of world.

Carry out asking for information: say "An implementation of the following creative brief:

People wander around some small map, on errands. One, sad to tell, has Gelato's Syndrome, a tragic condition turning one's skin the colour of a random flavour of ice cream (raspberry ripple, neapolitan, etc.). When two people are in the same room, there's a 1/3 chance that an infected person will infect a non-infected one. The player can cure any single person: victory condition - to stamp out the disease."

When play begins:

say "Gelato's Syndrome. It's struck, and it's struck hard. In these sticky summer months, there's no telling who will contract the disease next.";
now the command prompt is "[if the destination of the player is not blank] (heading to [destination of the player]) [end if]>".

Section 1 - Errands

The current actor is a person which varies. The current owner is a person which varies.

Every turn:
if player is active, follow the character movement rules.

Every turn:
now the last person named is the player;
now the last thing named is the player;
now every person is active.

A person can be active or passive. The player is passive.

The character movement rules are a rulebook.

The first character movement rule:
now group size is 1;
now the last person named is the player;
now the last thing named is the player;
now the player is passive.

A character movement rule:
repeat with mover running through innocent people:
now the current actor is the mover;
follow the shopper rules;
now the current actor is passive;
follow the movement reporting rule.

A character movement rule:
repeat with next mover running through mercantile people:
now the current owner is the next mover;
follow the shopowner rules;
now the current owner is passive;
follow the infection rule.

To decide whether movement has not yet occurred:
if the player is passive, no;
yes.

Definition: a person is mercantile if it owns a room. Definition: a person is innocent if it is not mercantile and it is not the player.

The shopowner rules is a rulebook.

A shopowner rule:
let the shop be a random room owned by the current owner;
if the shop is air-conditioned and an open door (called the escape) protects the shop, try the current owner closing the escape instead.

Report someone closing a door when the person asked owns the location:
say "[The person asked], muttering darkly about air-conditioning and electricity, closes [the noun]." instead.

Report Vanessa closing the metal door when the metal door is visible:
if Vanessa is visible, say "Vanessa watches serenely as the metal door slides automatically back in place, sealing Cold Comfort." instead;
otherwise say "The metal door slides heavily back into place." instead.

A shopowner rule:

if the location of the current owner encloses a submitted artwork (called the target):

try the current owner filing the target.

Filing is an action applying to one thing.

Before someone filing something which is not carried by the person asked:

try the person asked taking the noun instead.

Carry out someone filing:

if the person asked does not carry the noun and the person asked is visible, say "[The person asked] tries unsuccessfully to get [the noun]." instead; now the noun is nowhere.

Report someone filing:

say "[The person asked] registers [the noun] and files it away."

The shopper rules is a rulebook.

A shopper rule:

if the current actor carries something (called the problem), try the current actor resolving the problem instead.

A shopper rule:

if the current actor is not in the pool hall and the air conditioner is switched on:

try the current actor approaching the pool hall;

otherwise:

let way be a random direction;

try the current actor going the way.

Definition: a room is air-conditioned:

if it is outdoors, no;

if it is the Pool Hall and the air conditioner is switched off, no;

if it is protected by a door, yes;

no.

Protection relates a door (called X) to a room (called Y) when the front side of X is Y or the back side of X is Y. The verb to protect means the protection relation.

Ownership relates one person to various rooms. The verb to own means the ownership relation.

Resolving is an action applying to one thing.

An artwork is a kind of thing. Before printing the name of an artwork, say italic type. After printing the name of an artwork, say roman type. An artwork can be submitted or reserved.

A book is a kind of artwork.

Before someone resolving a book when the person asked is not in the Public Library:

try the person asked approaching the Public Library instead.

Carry out someone resolving a book:
move the noun to the Public Library;
now the noun is submitted.

Report someone resolving a book:
say "[The person asked] turns in [the noun]."

Before listing contents: group books together.

Before grouping together books: say "books entitled ".

A stamped envelope is a kind of thing.

Before someone resolving a stamped envelope when the person asked is not in the Post Office:
try the person asked approaching the Post Office instead.

Carry out someone resolving a stamped envelope:
now the noun is nowhere.

Report someone resolving a stamped envelope:
say "[The person asked] slips [a noun] into the outgoing mail slot."

Instead of someone resolving a stamped envelope when the person asked carries at least two stamped envelopes:
if the person asked is visible, say "[The person asked] shoves into the mail slot [a list of stamped envelopes carried by the person asked].";
repeat with item running through stamped envelopes carried by the person asked:
now the item is nowhere.

A DVD is a kind of artwork.

Before someone resolving a DVD when the person asked is not in the Rental Store:
try the person asked approaching the Rental Store instead.

Carry out someone resolving a DVD:
now the noun is submitted;
move the noun to the Movie Rental Store.

Report someone resolving a DVD:
say "[The person asked] returns [the noun]."

Instead of someone resolving a DVD when the person asked carries at least two DVDs:
if the person asked is visible, say "[The person asked] turns in [a list of DVDs carried by the person asked].";
now every DVD carried by the person asked is submitted;
now every DVD carried by the person asked is in the location of the person asked.

Before listing contents: group DVDs together.

Before grouping together DVDs: say "DVDs of " .

Approaching is an action applying to one thing.

Carry out someone approaching:

- let the way be the best route from the location of the person asked to the noun, using doors;
- if the way is a direction, try the person asked going the way;
- otherwise stop the action.

A coupon is a kind of thing.

Carry out someone resolving a coupon:

- try the person asked giving the noun to Vanessa.

The block giving rule is not listed in any rulebook.

Check giving something to someone (this is the block player giving rule):
abide by the block giving rule.

Before someone resolving a coupon when the person asked is not in Cold Comfort:

- try the person asked approaching Cold Comfort instead.

After someone giving a coupon to Vanessa:

- let the reward be a random ice cream cone;
- let the new flavor be a random infection color;
- now the infection color of the reward is the new flavor;
- move the reward to the person asked;
- now the noun is nowhere;
- if Vanessa is visible, say "[The person asked] trades in [the noun] and receives [a reward] from Vanessa."

Infection color is a kind of value. The infection colors are french vanilla, whole-bean vanilla, mint, chocolate, dark chocolate, chocolate chip, chocolate fudge, mint chocolate chip, chocolate chocolate chip, triple chocolate, white chocolate, white chocolate chip, aztec cocoa-chili, raspberry ripple, neapolitan, rum raisin, dulce de leche, strawberry chunk, rocky road, blackberry sorbet, lemon sherbet, lime ice, caramel swirl, mango, saffron silk, and cookie dough cream.

To say list of flavors:

- let current color be french vanilla;
- while current color is not cookie dough cream:
 - say "[current color], ";
 - now current color is the infection color after the current color;
 - say "and [current color]".

Understand "ask vanessa for [flavored ice cream]" as buying the flavor.
Understand "buy [flavored ice cream]" as buying the flavor.

Buying the flavor is an action applying to one infection color.

Check buying the flavor:

- unless the player can see Vanessa:

say "It would help if you were in the presence of an ice cream salesperson."
instead.

Carry out buying the flavor: say "'Do you have a coupon?' Vanessa demands.
You admit you do not. 'No [infection color understood] for you!'"

Understand "ice cream" or "cream" or "ice" or "sherbet" or "sorbet" as "[ice
cream]".

Understand "[infection color]" or "[infection color] [ice cream]" as "[flavored ice
cream]".

An ice cream cone is a kind of thing. An ice cream cone is always edible. An ice
cream cone has an infection color. An ice cream cone can be half-eaten or fresh.
Understand the infection color property as referring to an ice cream cone.

Carry out someone resolving an ice cream cone:
try the person asked eating the noun instead.

Instead of someone eating a fresh ice cream cone:
now the noun is half-eaten;
if the person asked is visible, say "[The person asked] licks [the noun]."

Report someone eating an ice cream cone:
say "[The person asked] pops the end of [the noun] into [if the person asked is
female]her[otherwise]his[end if] mouth and swallows." instead.

Before printing the name of an ice cream cone:
say "[if half-eaten]half-eaten [end if][infection color]".

Section 2 - Infection Rules

This is the infection rule:
if an infected person (called typhoid mary) can see a clean person (called
random bystander) and a random chance of 1 in 3 succeeds:
try typhoid mary sneezing on the random bystander.

A person can be infected or clean. A person has an infection color.

Every turn:
if the player is infected, say "You feel itchy."

Definition: a person is other if it is not the player. Definition: a person is another if
it is other.

When play begins: now right hand status line is "Sick: [number of infected
people]/[number of people]".

Every turn:
if every person is infected, end the story saying "Everyone succumbs";
if every person is clean, end the story finally saying "The Syndrome is
eradicated".

Understand "sneeze on [something]" as sneezing on. Sneezing on is an action applying to one thing.

Check sneezing on:

- if the player is clean, say "You're not sickly." instead;
- if the noun is the player, say "Ew." instead;
- if the noun is not a person, say "[The noun] cannot be infected." instead.

Carry out sneezing on:

- now the noun is infected;
- now the infection color of the noun is a random infection color.

Carry out someone sneezing on:

- now the noun is infected;
- now the infection color of the noun is a random infection color.

Report sneezing on:

- say "Unable to control yourself, you sneeze on [noun].".

Report someone sneezing on:

say "[The person asked] sneezes on [if the noun is the player]you[otherwise] [noun][end if]!".

Understand "inject [someone] with [something]" as injecting it with. Understand "inject [someone] with [syringe]" as injecting it with. Understand "use [syringe] on [someone]" as injecting it with. Understand the commands "innoculate" and "vaccinate" as "inject".

Injecting it with is an action applying to two things.

Check injecting it with:

- if the second noun is not the syringe, say "[The second noun] cannot inject anything." instead;
- if the noun is clean:
 - if the noun is the player, say "You're not infected yet." instead;
 - say "[The noun] is not infected, and the syringe contains a cure, not a vaccine." instead.

Carry out injecting it with: now the noun is clean.

After injecting the player with something: say "You inject yourself, wincing at the sting. But the itching fades almost at once."

Report injecting it with: say "You inject [the noun], who is now cured (but could easily be reinfected)."

Section 3 - Geography

Include Locksmith by Emily Short.

Understand "go to/toward/into [any room]" as going toward. Understand "enter [any room]" as going toward.

A person has a room called the destination.

Going toward is an action applying to one thing.

Check going toward:

if the noun is the location, say "You're already in [the location]." instead.

Carry out going toward:

now the destination of the player is the noun;

let heading be the best route from the location to the noun, using even locked doors;

if heading is not a direction, say "You can't think how to get there from here."

instead;

try going heading;

if the location is the destination of the player, now the destination of the player is blank.

Instead of waiting when the destination of the player is not blank:

if the destination of the player is the location:

now the destination of the player is blank;

otherwise:

try going toward destination of the player;

if the location is the destination of the player, now the destination of the player is blank.

Understand "stop" or "cease" as stopping. Stopping is an action applying to nothing. Carry out stopping: now the destination of the player is blank. Report stopping: say "You stop in your tracks."

After going to an air-conditioned room:

say "You step into the mercifully air-conditioned surroundings of...";

continue the action.

After going from an air-conditioned room:

say "You emerge from the air-conditioning into heat like a wall...";

continue the action.

Instead of listening to an air-conditioned room:

say "The air-conditioning hums softly."

The Alfred Cralle Pool Hall is a room. "The town's most popular gathering-place, the pool hall is decorated in honor of the inventor of the ice cream scoop." The air conditioner is a device in the Pool Hall. "[if switched off]An air conditioner sits in the corner, unhappily inert[otherwise]The air conditioner hums briskly[end if]."

The felt door is west of the Pool Hall. The felt door is a door. The felt door is open. The felt door is lockable and unlocked. The key to the city unlocks the felt door. The description of the felt door is "It has a prominent lock, designed for an old-fashioned key."

After locking a door with something in the presence of an other person (called audience):

say "[The audience] looks a little non-plussed when you lock [the noun], but shrugs."

Nancy Johnson Memorial Square is west of the felt door. The description of Nancy Johnson Memorial Square is "Waves of August heat rise from the

pavement: more than once you've had the fancy that your shoes are simply going to stick. At the center of the square, rubbed to a brownish polish by many adoring hands, is the statue of Mrs. Nancy Johnson of New Jersey."

The statue is scenery in Memorial Square. Understand "nancy" or "johnson" or "mrs" as the statue. The description of the statue is "Mrs. Johnson is pictured with a hand-cranked ice cream freezer tucked under one arm. Her other hand grips an ice cream scoop, ready to serve frozen dessert to the huddled masses." A hand-cranked ice cream freezer is part of the statue. The description is "The hand-cranked ice cream freezer was Mrs. Johnson's invention in 1846, though it was William Young who had the sense to patent the thing in 1848." The scoop is part of the statue. The description of the scoop is "An anachronism: Alfred Cralle would not invent the tool until 1897."

The Post Office is northwest of Nancy Johnson Memorial Square. "Service at the post office is on the slow side since everything went automated." The slot is scenery in the post office. The slot is a container. Carry out inserting something into the slot: now the noun is nowhere. Report inserting something into the slot: say "[The noun] falls out of sight, and you know you will never see it again."

Hamwi Street is northeast of an iron gate. "A U-shaped street running from Main Street around to the Memorial Square, Hamwi Street was recently added by ambitious city planners. The small and straggly line of trees has yet to grow enough to provide perceptible shade, so the street is even hotter and more unforgiving than the other parts of town."

The iron gate is northeast of Nancy Johnson Memorial Square. The iron gate is a door. It is lockable and unlocked.

Before printing the name of the iron gate while not opening or closing or locking or unlocking:

```
if the person asked is the player:
    if the gate is open, say "open ";
otherwise if the gate is locked:
    say "locked ";
otherwise if the gate is closed:
    say "closed ".
```

Cold Comfort Ice Cream is north of a metal door. The metal door is north of Hamwi Street. A poster is fixed in place in Cold Comfort. "A poster fills one wall with the blazing promise of treats to come." The description of the poster is "Coming soon! Thai ice creams! Durian, jackfruit, taro, and coconut flavors!"

The metal door is a door. "A frosty metallic door separates [the location] from [the other side of the metal door]." The metal door is lockable and unlocked. The key to the city unlocks the metal door.

Marciony Street is southeast of Nancy Johnson Memorial Square. "A semi-circular terrace, named somewhat fancifully after one claimant to the invention of the ice cream cone -- though Hamwi Street competes for the same honor. There are wedges of cool shadow here and there thanks to the buildings, but for the most part the southern exposure keeps Marciony unpleasantly hot."

The Movie Rental Store is west of a glass door. The glass door is a door. It is west of Marciony Street. The glass door is lockable and unlocked. The key to the

city unlocks the glass door.

Main Street is southeast of Hamwi Street. Main Street is northeast of some bronze gates.

The emergency box is in Main Street. The emergency box is fixed in place. "A fire-red box with a glass front faces the sidewalk, with 'In case of emergency, BREAK GLASS' lettered on it." The emergency box is closed and transparent. Understand "glass" as the box. Instead of attacking the closed emergency box: say "You hit the emergency box, which shatters open."; now the emergency box is open. Instead of attacking the open emergency box: say "The glass has already been thoroughly broken."

The syringe is in the emergency box. The description of the syringe is "It contains the cure for Gelato's Syndrome. You can inject anyone you like with it."

The bronze gates are northeast of Marciony Street. The bronze gates are a door. The bronze gates are lockable and unlocked. The description of the bronze gates is "Erected during the milk-taint revolution of 1937, they were designed to keep Main Street safe from the depredations of dairy-starved rioters."

The Public Library is east of Main Street. "Built in the 1920s during the height of the dairy boom, the public library has lush pink velvet seats, marble walls the color of fresh cream, and a motif of cherries carved around every doorframe. An incongruous sign hangs from the ceiling." The incongruous sign is scenery in the Public Library. The description of the incongruous sign is "Eating and drinking in the library is STRICTLY PROHIBITED."

Town Hall is southeast of Main Street. "Town Hall was built during the slow days of the ice-cream bust, and therefore it is as joyless and utilitarian as the Public Library is ridiculous. Unwilling to be reminded of their pain, the inhabitants steered clear of any decoration that might remotely be construed to resemble a scoop of anything: so there are no curves, only disciplined right angles." The key to the city is in Town Hall. It unlocks the iron gate. It unlocks the bronze gates. The description of the key to the city is "A skeleton key."

A room can be indoors or outdoors. The Post Office, the Alfred Cralle Pool Hall, the Store, Cold Comfort, Town Hall, and the Library are indoors.

Use full-length room descriptions.

After looking in an outdoors room:

```
let started printing be false;
now every proximate door is not mentioned;
if an indoors room is adjacent:
  let started printing be true;
  say "From here you can head into [the list of adjacent indoors rooms][if a
proximate door is not mentioned], or go through [the list of proximate doors
which are not mentioned][end if]. [run paragraph on]";
if an outdoors room is adjacent:
  say "You could[if started printing is true] also[end if] go ";
  let count be the number of adjacent outdoors rooms;
  let index be count;
  repeat with next room running through adjacent outdoors rooms:
    let way be the best route from the location to the next room;
```

```
    say "[way] to [the next room]";
    decrement index;
    make delimiter index of count, continuing;
    if a proximate door is not mentioned:
        let started printing be true;
        say "[if started printing is true]Also available[otherwise]Your available
exits[end if] [is-are the list of proximate doors which are not mentioned].";
    otherwise:
        if started printing is true, say paragraph break.
```

Definition: a door is proximate:
if the front side of it is the location, yes;
if the back side of it is the location, yes;
no.

Before exiting when the player is in an indoors room:
if the player can see a door (called nearest exit), try entering the nearest exit
instead;
repeat with way running through directions:
let next room be the room way from the location;
if the next room is a room, try going way instead.

Blank is a room. The destination of the player is Blank. Blank contains 15 ice
cream cones.

Section 4 - Other Players

Vanessa is a woman in Cold Comfort. Vanessa owns Cold Comfort.

Francine is a woman in the Public Library. Francine carries a book called Phlox
for Phyllis. Francine carries a stamped envelope called a pink stamped
envelope.

Lewis is a man in the Alfred Cralle Pool Hall. Lewis carries 3 stamped
envelopes. Lewis carries a book called Idiot's Guide to Dating. Lewis carries a
book called How to Meet Women. Lewis carries a book called Seduction in
Three Easy Steps. Lewis carries a DVD called Sleepless in Seattle.

Gene is a man in Nancy Johnson Memorial Square. Gene carries a stamped
envelope. Gene carries a DVD called Casablanca. Gene carries a coupon.

Rhoda is a woman in Marciony Street. Rhoda carries a book called The
Marciony Street Murders. Rhoda carries a DVD called Unsolved Serial Killings
XVIII. Rhoda carries a stamped envelope called a squashy package.

Martin is a man in Main Street. Martin carries a DVD called The Lifecycle of the
South Sea Tortoise. Martin carries a coupon.

Antony is a man in Movie Rental. Antony carries a coupon. Antony carries a
stamped envelope called a postcard.

Shelby is a man in the Town Hall. Shelby carries a DVD called Conducting An
Orderly Meeting. Shelby carries 5 stamped envelopes. Shelby carries an ice
cream cone. Shelby carries a coupon.

Christopher is a man in the Library. Christopher owns the Library.

Linnea is a woman in the Alfred Cralle Pool Hall. Linnea owns the Alfred Cralle Pool Hall.

Ned is a man in the Movie Rental Store. Ned owns the Movie Rental.

After printing the name of someone (called target) while listing contents: if the target owns the location of the target, say " (the owner)".

The description of a person is usually "[The noun] [if the noun is clean]looks healthy[otherwise]is the color of [infection color of the noun][end if]."

After examining another person who is carrying something: say "[if the noun is female]She[otherwise]He[end if] is carrying [a list of things carried by the noun]."

When play begins: let Patient Zero be a random other person; now patient zero is infected.

This is a light variation of a previous example, but we use it here because it is convenient:

Section 5 - Conversation

A person has a table name called conversation.

Instead of asking someone about something:
let the source be the conversation of the noun;
if topic understood is a topic listed in source:
if there is a turn stamp entry:
say "You have already heard that [summary entry].";
otherwise:
now turn stamp entry is the turn count;
now the character entry is the noun;
say "[reply entry][paragraph break]";
otherwise:
say "[The noun] stares at you blankly."

Instead of telling someone about something:
try asking the noun about it.

Understand "recap" or "recall" or "review" as recalling conversations.

Recalling conversations is an action applying to nothing.

Carry out recalling conversations:
repeat with speaker running through other people:
let source be the conversation of the speaker;
sort source in turn stamp order;
say "[The speaker] has so far told you: [line break]";
let index be 0;
repeat through source:
if there is a turn stamp entry and the speaker is character entry:
let index be 1;

say " [summary entry][line break]";
 if index is 0, say " absolutely nothing[line break]";
 say line break.

The conversation of a person is usually Table of General ChitChat.

Table of General ChitChat

topic	reply	summary	turn stamp	character
"weather/heat/warmth"	"It's appalling, isn't it? You'd think we didn't pay our taxes."	"that the weather is appalling"	a number	a person
"sun/sunlight"	"Good thing the town mostly switched to solar power, har, har."	"that the town is mostly relying on solar power"		
"rain"	"Nope, there isn't going to be rain for 132 days," replies [the noun]."	"that rain is not expected for another 132 days"		
"snow/hail/ice"	"This hilarious sally is greeted with hoots of laughter only."	"that the concept of snow is downright laughable"		
"disease/sickness/illness/syndrome"	"You get a cold, fixed stare in response. 'That's not funny,' [the noun] replies finally."	"that discussing the disease is more or less taboo"		
"cold comfort"	"'If you haven't tried it, you should,' says [the noun]. 'Best ice cream in town, and that's saying something, you bet.'"	"that Cold Comfort has the best ice cream in town"		
"town/city/village"	"'Yeah, it's a mite odd,' allows [the noun]. 'Not to everyone's taste, like...' [the noun as pronoun] considers for a moment. 'Like ginger ice cream. Big pieces of crystallized ginger... not everyone likes that.'"	"that the town is a mite odd"		
"forecast/weatherman" or "weather forecast/man"	"'Oh, the weather man's gotten a lot more reliable since the gummint started making it for us,' says [the noun]. 'Now he just reads off the schedule on the air every morning. Pretty much takes the fun right out of the news, if you ask me.'"	"that the weather is all generated by schedule"		
"taxes/tax" or "weather tax"	"A snort. 'You'd think for the rates we pay we'd get something a little pleasanter, don't you?'"	"that the weather tax really ought to be paying for something nicer than what you get"		
"job/employment/work"	"'[if the noun owns a room (called the shop)]I own [the shop],' replies [the noun][otherwise]Work at the creamery, like most folk around here,' answers [the noun]."	"this and that about employment in town"		
"book/books/reading"	"'The Public Library has a good selection, excepting only the cookbook section,' says [the noun]. 'That got censored way back when-- well, way back.'"	"that the Public Library has a good collection, except for the cookbook section"		

The conversation of Vanessa is the Table of Vanessa Chatter.

Table of Vanessa Chatter

topic	reply	summary	turn stamp	character
"ice cream" or "sorbet/sherbet/behavior/flavors/flavour/flavours/ice/ices"	"The flavors are [list of flavors],' she responds promptly, without needing to draw breath."	"that the flavors are [list of flavors]"	a number	a person

```

After reading a command:
  while player's command includes "the":
    cut the matched text.

```

This strips 'the' out of the command, so that ASK PERSON ABOUT THE RAIN will be understood as well as ASK PERSON ABOUT RAIN.

Now we try something a bit unusual. Inform on its own will report each action on its own line, so that each character who walks into or out of a room will be described in a separate paragraph. This is usually fine, but in a game with a lot of characters moving around simultaneously, it can become a bit overwhelming. Instead, we may want to condense these reports into a single line, such as "Ben and Jerry enter from the south". The following accomplishes that goal by replacing some of the reporting rules, storing the information in a table, and then reading the table back later, once all the character movement has been resolved and the reports can usefully be collated:

Section 6 - Movement Description

A person has some text called walk style. The walk style of a man is usually "stride". The walk style of a woman is usually "strut". The walk style of Gene is "[one of]wander[or]stroll[purely at random]". The walk style of Francine is "waddle". The walk style of Antony is "scamper". The walk style of Rhoda is "sashay".

Table of Visible Exits

```

character      second  third  heading chosen total
a person      a person a person a direction    a number
with 10 blank rows.

```

Table of Visible Entrances

```

character      second  third  heading chosen total
a person      a person a person a direction    a number
with 10 blank rows.

```

```

To clear (current table - a table name):
  repeat through current table:
    blank out the whole row.

```

```

To tidy departures of (current table - a table name):
  let next direction be up;
  repeat through current table:
    if heading chosen entry is next direction:
      let accomplice be character entry;
      choose row with heading chosen of next direction in the current table;
      if total entry is 1:
        now second entry is accomplice;
        now total entry is 2;
      if total entry is 2:
        unless the second entry is accomplice:
          now third entry is accomplice;
          now total entry is 3;
      choose row with character of accomplice in the current table;
      blank out the whole row;

```

otherwise:
let next direction be heading chosen entry.

A door has a person called last opener.

Report someone opening a door:
now group size is 1;
now the last opener of the noun is the person asked;
if the person asked is visible, say "[The person asked] opens [the noun]. [run paragraph on]" instead;
otherwise say "[The noun] opens from the other side. [run paragraph on]" instead.

Report someone going through a door (called route):
if the person asked is not the last opener of the route, continue the action;
if the person asked is the last person named, say "[The person asked as pronoun]";
otherwise say "[The person asked]";
say " [if the person asked is in the location]comes[otherwise]goes[end if] through[if the last thing named is not the route] [the route][end if]." instead.

The last thing named is a thing that varies. Before printing the name of something (called target) which is not a person: now the last thing named is the target.

Report someone going a direction:
if the person asked is in the location,
choose a blank row in the table of visible entrances;
otherwise choose a blank row in the table of visible exits;
now character entry is the person asked;
now total entry is 1;
if the person asked is in the location,
now heading chosen entry is the opposite of the noun;
otherwise now heading chosen entry is the noun;
stop the action.

This is the movement reporting rule:
sort the Table of Visible Entrances in heading chosen order;
tidy departures of the table of visible entrances;
sort the Table of Visible Exits in heading chosen order;
tidy departures of the table of visible exits;
let total row count be the number of filled rows in the Table of Visible Entrances plus the number of filled rows in the Table of Visible Exits;
if total row count is 0, rule succeeds;
generate descriptions from the Table of Visible Entrances;
generate descriptions from the Table of Visible Exits;
clear the Table of Visible Entrances; clear the Table of Visible Exits.

To generate descriptions from (current table - a table name):
let count be the number of filled rows in the current table;
if count is 0, rule succeeds;
let index be count;
repeat through the current table:
let accomplice be character entry;
if character entry is a person, now character entry is marked for listing;
if there is a second entry and second entry is a person, now second entry is

marked for listing;
 if there is a third entry and third entry is a person, now third entry is marked for listing;
 let target be the room the heading chosen entry from the location;
 if total entry is 3, say "[The character entry], [the second entry][optional comma] and [the third entry] ";
 if total entry is 2, say "[The character entry] and [the second entry] ";
 if total entry is 1:
 if the character entry is the last person named, say "[The character entry as pronoun] ";
 otherwise say "[The character entry] ";
 if total entry is 1, say "[walk style of the character entry]s ";
 otherwise say "walk[if total entry is 1]s[end if] ";
 if the character entry is in the location:
 if location is indoors and target is indoors, say "over from ";
 if location is outdoors and target is indoors, say "out of ";
 if location is indoors and target is outdoors, say "in from ";
 if location is outdoors and target is outdoors, say "over from ";
 otherwise:
 if location is indoors and target is indoors, say "over to ";
 if location is outdoors and target is indoors, say "into ";
 if location is indoors and target is outdoors, say "out [if a door is visible] [the random visible door][end if] to ";
 if location is outdoors and target is outdoors, say "over to ";
 if target is outdoors, say "[the heading chosen entry]";
 otherwise say "[the target]";
 if the total entry is 1 and count is 1 and accomplice carries something, say ", carrying [a list of things carried by the accomplice]";
 decrement index;
 make delimiter index of count, continuing;
 now group size is total entry;
 if a marked for listing person is infected:
 [eliminate the case in which we have already seen this description because we just typed LOOK and the patient was in the room at the time]
 if looking and a marked for listing person is not in the location:
 clear marked people;
 say paragraph break;
 otherwise:
 describe patients;
 otherwise:
 clear marked people;
 say paragraph break.

The last person named is a person that varies. Before printing the name of a person (called target): now the last person named is the target. Group size is a number that varies. Group size is 1.

To clear marked people:
 repeat with named party running through people:
 now the named party is not marked for listing.

Before listing nondescript items:
 if the number of people who are marked for listing is 0, make no decision;
 say "You can see [a list of people who are marked for listing] here. ";
 now group size is the number of people who are marked for listing;
 describe patients;
 now every marked for listing person is not marked for listing.

To describe patients:

if every marked for listing person is infected and at least three people are marked for listing:

say "They are all sick as dogs, every one.";

clear marked people;

rule succeeds;

otherwise:

if the number of people who are marked for listing is greater than two and the number of infected people who are marked for listing is greater than the number of clean people who are marked for listing:

say "Only [the list of clean people who are marked for listing] currently remain[if the number of clean people who are marked for listing is 1]s[if] untainted.";

clear marked people;

rule succeeds;

let count be the number of marked for listing other people who are infected;

if count is 0:

say paragraph break;

make no decision;

let index be count;

repeat with patient running through marked for listing other people who are infected:

if index is count:

if count is 1 and the patient is the last person named:

say "[The patient as pronoun]";

otherwise:

say "[The patient]";

otherwise:

say "[the patient]";

say " [looks as though dipped in for index] [infection color of the patient]";

decrement index;

make delimiter index of count;

clear marked people.

To say (named character - a man) as pronoun:

if group size is 1, say "He"; if group size is 2, say "The latter"; if group size is greater than 2, say "The last".

To say (named character - a woman) as pronoun: if group size is 1, say "She"; if group size is 2, say "The latter"; if group size is greater than 2, say "The last".

To say looks as though dipped in for (index - a number):

let divider be the number of filled rows in the Table of Dipping Phrases;

if index is greater than 4, let index be the remainder after dividing index by divider;

choose row index in the Table of Dipping Phrases;

say dipping entry.

Table of Dipping Phrases

dipping

"looks as though dipped in"

"could have been rolling in"

"has a bad case of"

"suffers from"

"contracted a virulent"

A door is usually scenery.

The next part could be simpler, but for rigor we will write it in such a way that it will work whether or not the serial comma is in use. This requires some extra work.

```
To make delimiter (index - a number) of (count - a number), continuing or halting:
  if index is 0:
    if continuing, say ". [run paragraph on]";
    otherwise say ".";
  otherwise if index is 1:
    if count is 2, say " and ";
    otherwise say "[optional comma] and ";
  otherwise:
    say ", ".
```

```
To say optional comma:
  if the serial comma option is active:
    say ", ".
```

```
Test me with "go to cold comfort / z / z / z / z / ask vanessa for french vanilla /
ask vanessa for chocolate / ask vanessa about flavors / ask vanessa for
chocolate chocolate chip".
```

Because so much of this game is randomized, it will not be possible to provide a test command that systematically solves it. A good strategy is to go to Main Street, get the syringe; go to the Town Hall and get the key; then visit the shops, inject everyone, and lock them in when they've all been injected. Then go to the Pool Hall, turn on the air conditioner, and wait for the remaining parties to show up.

This is also something that could get fairly slow if we added many more rooms and characters to it. In that case, we might want to select fast route-finding so that character movement won't take as long. This will cost memory, possibly forcing the game into Glulx format if it isn't already, but significantly reduce the run-time for large maps with numerous people moving each turn:

Use fast route-finding.

184

★ Example Virtue

Defining certain kinds of behavior as inappropriate, so that other characters will refuse indignantly to do any such thing.

WI

"Virtue"

The Cloister is a room. Lady Teresa is a woman in the Cloister. Mother Margaret is a woman in the Cloister.

Attacking someone is misbehavior. Kissing someone is misbehavior.

Instead of asking someone to try misbehavior: say "[The person asked] stiffens, offended by your suggestion."

Test me with "kiss margaret / margaret, kiss me".

186

★ Example The Hypnotist of Blois

WI

A hypnotist who can make people obedient and then set them free again.

"The Hypnotist of Blois"

A person is either hypnotized or alert. A person is usually alert.

Persuasion rule for asking a hypnotized person to try doing something:
persuasion succeeds.

Understand "hypnotize [someone]" as hypnotizing.

Hypnotizing is an action applying to one thing.

Check hypnotizing:
if the noun is hypnotized, say "Already done." instead.

Carry out hypnotizing:
now the noun is hypnotized.

Report hypnotizing:
say "[The noun] slips under your control."

Instead of waking someone hypnotized:
now the noun is alert;
say "[The noun] returns abruptly to consciousness."

We will learn more about check rules for other characters shortly, but the following prevents the hypnosis patient from trying to hypnotize us in turn:

Check someone hypnotizing someone:
stop the action.

Maison de la Magie is a room. "In a darkened room, a few hundreds of paces from the chateau of Blois, you give to tourists three shows a day: displaying to them power they do not comprehend and spectacles they do not deserve."

A volunteer is a woman in the Maison. "A volunteer from the audience stands facing you, [if alert]skeptically awaiting hypnosis[otherwise]her face worshipful and obedient[end if]." The printed name of the volunteer is "volunteer from the audience". The description is "A distracted, susceptible woman." The volunteer wears a t-shirt and a baseball hat.

The player wears a top hat and a cape.

Test me with "volunteer, remove hat / hypnotize volunteer / look / volunteer, remove hat / wake volunteer / volunteer, wear hat".

371

🌟 Example Latin Lessons

WI

Supplying missing nouns and second nouns for other characters besides the player.

If we're defining actions for other characters to follow, we may want to include them in our "rule for supplying a missing noun". We can do this if we write our "while..." clause to apply to any actor, as follows:

"Latin Lessons"

The Latin Studio is a room. Rick is a man in the Studio.

A dance-name is a kind of thing. Argentine tango, samba, merengue, cha-cha, street salsa are dance-names.

Dancing is an action applying to one visible thing. Understand "dance [any dance-name]" as dancing. Understand "dance" as dancing.

Rule for supplying a missing noun while an actor dancing:
now the noun is street salsa.

Report someone dancing:
say "[The actor] dances a few steps of [the noun] for you."

Report dancing:
say "You dance a few steps of [the noun]."

Persuasion rule for asking someone to try dancing: persuasion succeeds.

Test me with "dance / dance samba / rick, dance / rick, dance merengue".

188

★★ Example Generation X

WI

A person who goes along with the player's instructions, but reluctantly, and will get annoyed after too many repetitions of the same kind of unsuccessful command.

"Generation X"

The Volunteer Center is a room. "A fairly spartan office, though there are a few attractive posters from advertising campaigns of the past, and an ominous map charting the deforestation of Brazil."

The desk is scenery in the Volunteer Center. "Your standard metal desk, with a drawer for postal supplies and the like." On the desk is a pile of leaflets and a pile of business cards. The power stapler is a device on the desk. A drawer is part of the desk. It is openable and closed. In the drawer are a sponge and a roll of bulk-mail stamps.

Instead of doing something:

- if examining, continue the action;
- if searching, continue the action;
- if looking, continue the action;
- if asking Jenna to try doing something, continue the action;
- say "The whole point of this exercise is to train Jenna. Once you have her on the envelope-stuffing, you can go make some calls."

Instead of answering Jenna that something:

- say "You're going to have to break the instructions down to the simplest ones possible, given that she's in the mood to misunderstand on purpose."

A cardboard box is in the Center. In the cardboard box is a pile of empty envelopes. The box is openable and closed. The description is "A recycling symbol is prominent on the side, which makes you feel a little bit better about using a mailing campaign for this cause."

Jenna is a woman in the Volunteer Center. "Your daughter Jenna is here, barely visible through the cloud of resentment." The description is "Jenna [if jenna carries something]has [the list of things carried by Jenna][otherwise]returns your stare, with added hostility[end if]."

A persuasion rule for asking Jenna to try doing something: persuasion succeeds.

Unsuccessful attempt by Jenna doing something:

- repeat through Table of Retorts:
- if the reason the action failed is the cause entry:
 - say "[response entry][paragraph break]";
 - rule succeeds;
- say "'I don't see how I'm supposed to do that, Mom,' Jenna says."

Table of Retorts

cause	response
can't take yourself rule	"Is that like 'get a grip on yourself' or something?' Jenna asks, momentarily diverted."
can't take what's fixed in place rule	"[physical impossibility]"
can't take scenery rule	"[physical impossibility]"
can't take what's already taken rule	"[already done]"
can't drop what's already dropped rule	"[already done]"
can't wear what's already worn rule	"[already done]"
can't take off what's not worn rule	"[already done]"
can't close what's already closed rule	"[already done]"
can't open what's already open rule	"[already done]"
can't switch off what's already off rule	"[already done]"
can't switch on what's already on rule	"[already done]"
can't unlock what's already unlocked rule	"[already done]"
can't lock what's already locked rule	"[already done]"

To say physical impossibility:

say "'Maybe you should've brought someone a little stronger,' Jenna says. 'Like the Incredible Hulk.' "

To say already done:

repeat through Table of Bored Remarks:

say "[response entry]";

blank out the whole row;

rule succeeds;

say "'Okay, I'm going to be, like, in the car,' says Jenna. 'Outside.' ";
end the story saying "Jenna has gotten fed up".

Table of Bored Remarks

response

"'Did that,' says Jenna."

"'Check,' says Jenna."

"'Yeah, Mom, I already did that, okay?'"

"'Look, if I have to be here doing dumb stuff, could you at least tell me to do stuff that isn't already done?' Jenna asks wearily."

"Jenna gives a great upward sigh, riffling her bangs. 'Once again... there is totally no point.'"

Test me with "e / x jenna / jenna, get stapler / jenna, get stapler / x jenna / jenna, drop stapler / jenna, drop stapler / jenna, open box / jenna, open box / jenna, switch stapler on / jenna, switch stapler on / jenna, take desk / jenna, open box / jenna, open box".

423



Example Northstar

WI

Making Inform understand ASK JOSH TO TAKE INVENTORY as JOSH, TAKE INVENTORY. This requires us to use a regular expression on the player's command, replacing some of the content.

Most of the time, Inform understands commands to other characters when they take the form "JOSH, TAKE INVENTORY" or "JOAN, WEAR THE ARMOR". But novice players might also try commands of the form ASK JOSH TO TAKE INVENTORY or ORDER JOAN TO WEAR THE ARMOR.

The easiest way to make Inform understand such commands is to meddle directly with the player's command, changing it into the format that the game will understand, as here:

"Northstar"

The Northstar Cafe is a room. "The Northstar is crammed with its usual brunch crowd, and you were lucky to get a table at all. You are now awaiting the arrival of your ricotta pancakes."

Josh is a man in The Northstar Cafe. "Josh is on his way past your table." The description of Josh is "He is a waiter here, but you also know him socially, so he tends to be more chatty than the other waiters." A persuasion rule: persuasion succeeds.

After reading a command:

```
let N be "[the player's command]";  
replace the regular expression "\b(ask|tell|order) (.+?) to (.+)" in N with "\2, \3";  
change the text of the player's command to N.
```

Test me with "ask Josh to take inventory / tell Josh to take inventory / order Josh to take inventory".

Note that we have to copy N back explicitly to replace the player's command.

239



Example For Demonstration Purposes

WI

A character who learns new actions by watching the player performing them.

Suppose we want to have a character who can dynamically learn new actions by observing the player performing them. We could do this by adding the actions to a list of things the character can do, but using a relation to express the same idea allows for tidier, easier-to-read code.

Thanks to Jesse McGrew for the initial design of this example.

"For Demonstration Purposes"

Section 1 - Procedure

Capability relates various people to various stored actions. The verb to be capable of means the capability relation.

Persuasion rule:

```
let CA be the current action with no specific actor;  
if the person asked is capable of CA:  
    persuasion succeeds;  
otherwise:  
    say "[The person asked] look[s] confused. Maybe a demonstration would  
help.";  
    persuasion fails.
```

The action requester is an object that varies. The action requester variable translates into I6 as "act_requester".

To decide which action is the current action with no specific actor:

```
let old actor be the person asked;  
let old requester be the action requester;  
now the person asked is the player;  
now the action requester is nothing;  
let CA be the current action;  
now the person asked is the old actor;  
now the action requester is the old requester;  
decide on CA.
```

The learning by observation rule is listed after the report stage rule in the specific action-processing rules.

Definition: a person is other if he is not the player.

This is the learning by observation rule:

repeat with the viewer running through other people who can see the player:
if the player is the actor and viewer is not capable of the current action:
say "[The viewer] watches your behavior with interest. Seems like [they] [are] learning.";
now the viewer is capable of the current action.

Section 2 - Scenario

The Daily Planet is a room. Clark is here. He is a man.

When play begins:

now Clark is capable of taking inventory.

Test me with "Clark, inventory / Clark, x me / x me / Clark, x me".

205



Example Under Contract

WI

Creating a person who accepts most instructions and reacts correctly when a request leads implicitly to inappropriate behavior.

"Under Contract"

The Sound Stage is a room. The description is "A somewhat creaky and unconvincing reproduction of the docks of San Francisco."

Clark Gable is a man in the Sound Stage. "Clark is here, carrying [the list of things carried by Clark][if Clark wears anything] and wearing [the list of things worn by Clark][end if]." Clark carries a swagger stick. Clark carries an open openable player's holdall called a briefcase. The briefcase contains a signed contract. Clark wears a fedora and a pair of pants. A handle is part of the briefcase.

The player carries an open openable player's holdall called a frilly bag. The carrying capacity of the player is 2. The player wears a sparkly scarf and a slinky dress. A strap is part of the bag. The tent is a portable enterable container.

Instead of taking off the slinky dress: say "Rowr! Not that kind of movie, babe."

Instead of taking off the pants: say "That's not your job, whatever the society pages might suggest."

Unsuccessful attempt by Clark doing something:

repeat through table of Clark Retorts:
if the reason the action failed is the cause entry:
say "[response entry][paragraph break]";

rule succeeds;
say "'I don't think that's in the script,' says Clark dubiously."

Table of Clark Retorts

cause	response
can't take yourself rule	"'I'm always self-possessed,' Clark remarks. You've heard that line before, but it sounds so much more convincing from him."
can't take other people rule	"'I don't think it would be appreciated if I tried to do that to [the noun],' he rumbles."
can't take component parts rule	"'I don't want to rip [the noun] out,' Clark remarks."
can't take people's possessions rule	"'I don't cotton to acting like a thief,' Clark replies. 'It ain't proper.'"
can't take what you're inside rule	"'Do you see where I am, babe?' Clark demands."
can't take what's already taken rule	"[already done]"
can't take scenery rule	"'I'm not the stunt man, darling,' he says with a wry twinkle."
can't take what's fixed in place rule	"'I'm not the stunt man, darling,' he says with a wry twinkle."
can't exceed carrying capacity rule	"Clark grins. 'I've only got so many hands, darling,' he says."
can't insert into closed containers rule	"[physical impossibility]"
can't go that way rule	"[physical impossibility]"
can't go through closed doors rule	"[physical impossibility]"
can't enter closed containers rule	"[physical impossibility]"
can't exit closed containers rule	"[physical impossibility]"
can't drop yourself rule	"'We're inseparable, me and me,' Clark replies, with a smile."
can't drop what's already dropped rule	"[already done]"
can't drop what's not held rule	"'Not under my control, [the noun],' replies Clark."
can't drop clothes being worn rule	"[salacious retort]"
can't put something on itself rule	"'I lack the dexterity,' says Clark. Oh, he's so modest."
can't put onto what's not a supporter rule	"'[[The second noun] won't support a thing,' says Clark reprovingly."
can't put clothes being worn rule	"[salacious retort]"
can't insert clothes being worn rule	"[salacious retort]"
can't give worn items rule	"[salacious retort]"
can't wear what's not clothing rule	"'Costuming just gets stranger every year,' says Clark. 'In short: no.'"
can't wear what's already worn rule	"[already done]"
can't eat unless edible rule	"'What're you trying to do, poison me?'"
can't eat clothing without removing it first rule	"[salacious retort]"
can't take off what's not worn rule	"[already done]"
can't close what's already closed rule	"[already done]"
can't open what's already open rule	"[already done]"
can't switch off what's already off rule	"[already done]"
can't switch on what's already on rule	"[already done]"
can't unlock what's already unlocked rule	"[already done]"
can't lock what's already locked rule	"[already done]"

To say already done:

repeat through Table of Clark's Bored Remarks:

say "[response entry]";

blank out the whole row;

rule succeeds;
say "'Already done.' "

Table of Clark's Bored Remarks

response

"I have anticipated your every demand."

"'We've been through this part of the script already,' Clark remarks, with just a hint of creeping tedium."

"'Right right, already taken care of...'"

"'I'm a believer in rehearsal,' Clark says, sounding more bored than ever, 'but we really have got all this done already.'"

To say salacious retort:

repeat through Table of Clark's Flirtatious Remarks:

say "[response entry]";

blank out the whole row;

rule succeeds;

say "Clark, driven past the point of endurance, seizes you in his arms and kisses you.";

end the story finally.

Table of Clark's Flirtatious Remarks

response

"Clark allows his mustache to quirk at the suggestion."

"Clark wiggles his eyebrows at you."

"'That kind of thing isn't in my contract, sweetie,' says Clark. 'If you're interested you'll have to ask off-camera.'"

"'I'd feel so... bare,' Clark says, with a sidelong look at you."

"'You first,' says Clark."

"Clark's stare is intense. You may be on the verge of breaking him."

To say physical impossibility:

repeat through Table of Clark's Frustrated Denials:

say "[response entry]";

blank out the whole row;

rule succeeds;

say "Clark makes a helpless gesture about his ability to move himself, or parts of himself, through solid objects. "

Table of Clark's Frustrated Denials

response

"Clark frowns. 'I can't reach through things -- unless you were planning a special effect?'"

"'Look,' says Clark. 'If you want a stage magician, hire one.'"

"Clark just groans."

"Clark says drily, 'Is this in the script, or are we doing improvisation now?'"

The block giving rule is not listed in the check giving it to rules. The block showing rule is not listed in the check showing it to rules. The report smelling rule is not listed in the report smelling rules. The report listening rule is not listed in the report listening to rules.

Carry out listening to something:

do nothing.

Report listening to something: say "Your attention bears no interesting result."

Carry out smelling something:
do nothing.

Report smelling something: say "Your attention bears no interesting result."

Report someone listening to something: say "[The person asked] concentrates, listening."

Report someone smelling something: say "[The person asked] sniffs at [the noun]."

Instead of asking someone for something: try asking the noun to try giving the second noun to the player.

Carry out showing something to someone: say "You reveal [the noun] to [the second noun]."

Carry out Clark showing something to someone:
if the second noun is the player:
say "Clark shows you [the noun]. [run paragraph on]";
try examining the noun;
otherwise:
say "Clark reveals [the noun] to [the second noun]."

Instead of asking someone to try saying yes: try saying yes. Instead of asking someone to try saying no: try saying no. Instead of asking someone to try saying sorry, try saying sorry.

And now we make Clark a very amenable type:

A persuasion rule for asking people to try doing something: persuasion succeeds.

There are quite a few actions for which no automatic other-character behavior is provided at all, because they always end in failure when the player tries them. Currently the response is not very interesting. Let's spice it up a bit:

Burning something is useless action. Waking up is useless action. Thinking is useless action. Cutting is useless action. Jumping is useless action. Tying something to something is useless action. Drinking something is useless action. Swinging is useless action. Rubbing is useless action. Setting something to something is useless action. Waving hands is useless action. Buying is useless action. Climbing is useless action. Sleeping is useless action. Kissing is useless action. Throwing something at something is useless action. Attacking is useless action. Asking something about something is useless action. Telling something about something is useless action. Answering something that something is useless action. Waking something is useless action.

A persuasion rule for asking people to try useless action:
say "Clark Gable frowns. 'I don't mean to be difficult, but I can't see any point in that.'";
persuasion fails.

A persuasion rule for asking Clark to try taking off the pants:

say "He gives you a look of mocking exasperation. Apparently you'll have to take a (slightly) more subtle approach.";
persuasion fails.

Check someone giving something to someone (this is the can't give worn items rule):

if the person asked wears the noun, stop the action.

And because we do not want Clark automatically taking off the pants as the result of a drop action:

Instead of Clark taking off the pants: do nothing.

We know that this can only come about as a default action, because we have arranged matters so that he cannot be persuaded directly.

Test me with "give dress to clark / clark, take off pants / clark, eat pants / clark, give me the pants / clark, drop pants / clark, put the pants in the briefcase / again / g / g / g"

189

★ Example IQ Test

WI

Introducing Ogg, a person who will unlock and open a container when the player tells him to get something inside.

We have already seen how Before... rules can generate implicit actions for the player, so that the player will, for instance, open doors before trying to walk through them. The same can be done for characters other than the player; so for instance:

"IQ Test"

The Donut Shop is a room. "Vibrantly decorated in donut colors: pink, brown, and cream."

Ogg is a man in the Donut Shop. "Ogg is slumped in the corner[if Ogg carries something] with [a list of things carried by Ogg][end if]. He wears a nametag which says 'HELLO MY NAME IS OG.'" Understand "og" as Ogg. Ogg wears a nametag. The description of the nametag is "Sadly misspelled."

The Donut Shop contains a transparent closed openable locked lockable container called a case. The case contains some cake donuts. The donuts are edible.

The matching key of the case is a silver key. The silver key is carried by Ogg.

A persuasion rule for asking someone to try doing something:
persuasion succeeds.

Before someone opening a locked thing (called the sealed chest):
if the person asked is carrying the matching key of the sealed chest, try the

person asked unlocking the sealed chest with the matching key of the sealed chest;

if the sealed chest is locked, stop the action.

Before someone taking something which is in a closed container (called the shut chest):

try the person asked opening the shut chest;

if the shut chest is closed, stop the action.

The block giving rule is not listed in the check giving it to rules.

Test me with "open case / get donuts / og, get donuts / og, give me the donuts / eat donuts".

(A more detailed set of before rules for people using doors and locks is provided in Emily Short's Locksmith extension.)

203



Example The Man of Steel

WI

An escaping action which means "go to any room you can reach from here", and is only useful to non-player characters.

It will sometimes be handy to write actions that are only available to the non-player characters and not to the player. To do this, we just define an action which has no "understand": the player will never be able to specify this command himself, but other characters can try it.

This is particularly useful for creating abstract or general actions for when we want a character to eat something, wear something, or go somewhere, but aren't too particular as to what the object is; as here, where we just want Clark to move away from the kryptonite, regardless of direction:

"The Man of Steel"

Escaping is an action applying to nothing.

Carry out someone escaping:

let space be the holder of the person asked;

let place be a random room which is adjacent to the space;

let way be the best route from the space to the place;

try the person asked going way.

Every turn:

if Clark Kent can see kryptonite, try Clark Kent escaping.

The Daily Planet Newsroom is a room.

Perry White's Office is west of the Newsroom. In Perry White's Office are a desk and a poster of Elvis. On the desk is a lead-lined box. The box is openable. In the box is the green kryptonite crystal.

The Supply Closet is east of the Newsroom. The Elevator Bank is north of the Newsroom.

Clark Kent is a man in the Newsroom. "Clark [if Clark can see the kryptonite]looks a bit ill[otherwise]is here, frowning as he revises his latest article[end if]."

Test me with "west / get box / east / close box / east / west / north / south / west".

208



Example The Man of Steel Excuses Himself

WI

Elaborating the report rules to be more interesting than "Clark goes west."

Report rules can be a good point at which to add local color: while Inform provides default descriptions of character behavior, these are rather generic and can stand to be customized.

For instance, if we wanted to liven up our previous Clark Kent example:

Report Clark Kent going a direction (called the way):

say "[one of]With a particularly weak excuse[or]Muttering[at random] about [random excuse subject], Clark heads [way]." instead.

To say random excuse subject:

choose a random row in the Table of Lame Excuses; say "[reply entry]".

Table of Lame Excuses

reply

"needing a paper-clip"

"wanting an English-Tuvalu dictionary"

"walking a neighbor's dog"

"hearing air-raid sirens"

"having drunk too much coffee"

"thinking he smells smoke"

"wondering where Lois got to"

"needing to speak to Jimmy"

"noticing the Good Year blimp"

It's good to be careful, as the library report rules have been designed and tested to describe every contingency (going through doors, going in vehicles, etc.): so when replacing a report rule, we should try to consider all the possible variations of the action that we might want to describe.

However, in this case, our scenario is so simple that there are no doors, vehicles, or pushable objects, so we're safe in giving Clark a very simple reporting scheme.

190



Example Boston Cream

WI

A fuller implementation of Ogg, giving him a motivation of his own

and allowing him to react to the situation created by the player.

"Boston Cream"

Use scoring.

The Donut Shop is a room. "Vibrantly decorated in donut colors: pink, brown, and cream."

Ogg is a man in the Donut Shop. "Ogg is slumped in the corner[if Ogg carries something] with [a list of things carried by Ogg][end if]. He wears a nametag which says 'HELLO MY NAME IS OG.'" Understand "og" as Ogg. Ogg wears a nametag. The description of the nametag is "Very neatly written."

The Donut Shop contains a transparent closed openable locked lockable enterable container called a case. The case is fixed in place. The case contains some cake donuts, some jelly donuts, and some apple fritters. "The [if unopenable]damaged[otherwise]gleaming[end if] donut case [if something is in the case]contains [a list of things in the case][otherwise]has been stripped of its contents[end if]." The cake donuts, the jelly donuts, and the apple fritters are edible.

The matching key of the case is a silver key. The silver key is in a mesh basket. The mesh basket is closed, transparent, and openable. It is in the Donut Shop.

Before someone taking something which is carried by the player:

- if the person asked cannot touch the player:
 - say "Ogg looks with a fixed frown at [the noun].";
 - continue the action;
- say "[The person asked] comes up and shakes your arm until you drop [the noun].";
- say line break;
- silently try dropping the noun;
- stop the action.

Before someone unlocking a locked thing with something which is not carried by the person asked:

- try the person asked taking the second noun;
- stop the action.

Before someone opening a locked thing (called the sealed chest):

- if the person asked can see the matching key of the sealed chest:
 - if the matching key of the sealed chest is enclosed by the sealed chest:
 - say "Seeing the paradoxical location of [the matching key of the sealed chest], [the person asked] gives a howl of rage.";
 - increment the score;
 - end the story finally saying "You have thwarted Ogg";
 - otherwise:
 - try the person asked unlocking the sealed chest with the matching key of the sealed chest;
 - stop the action.

Before someone taking something which is in a closed container (called the shut chest):

try the person asked opening the shut chest;
stop the action.

Ogg has a number called hunger. The hunger of Ogg is 0.

Every turn:

increment the hunger of Ogg;
if the hunger of Ogg is 2 and Ogg is visible, say "Ogg's stomach growls.";
if the hunger of Ogg is 3 and Ogg is visible:
 if Ogg can see an edible thing (called the target):
 say "Ogg eyes [the target] with obvious intent.";
 otherwise:
 say "Ogg glances at you in a disturbingly shifty way.";
if the hunger of Ogg is greater than 3:
 if Ogg carries an edible thing (called the target):
 try Ogg eating the target;
 otherwise:
 let new target be a random edible thing which can be seen by Ogg;
 if the new target is a thing:
 try Ogg taking the new target;
 otherwise:
 if Ogg can touch the player, end the story saying "Ogg is gnawing your
ankle";
 otherwise try Ogg taking the player.

The crumbs are a thing. "Crumbs of [the list of edible things which cannot be
seen by the player] lie scattered over the whole floor."

Instead of asking Ogg to try doing something when Ogg cannot touch the player:
 say "Ogg tilts his head and shrugs, unable to hear your instruction clearly."

Instead of asking Ogg to try eating something:
 say "It's not as though Ogg really needs any encouragement in that
department, is it?"

Definition: Ogg is hungry if the hunger of Ogg is greater than 2.

Persuasion rule for asking Ogg to try doing something: if Ogg is hungry,
persuasion fails; persuasion succeeds.

Persuasion rule for asking Ogg to try giving something edible to the player:
persuasion fails.

Unsuccessful attempt by Ogg doing something:
 if the reason the action failed is a failing listed in the Table of Ogg Retorts:
 say "[reply entry][paragraph break]";
 otherwise:
 say "Ogg looks adorably confused."

Table of Ogg Retorts

failing	reply
can't take people's possessions rule	"Ogg too polite."
can't take other people rule	"Ogg not that greedy."
can't take scenery rule	"[The noun] very very heavy."
can't take what's fixed in place rule	"[The noun] very heavy."

can't drop what's not held rule "'Hunh?'"

Carry out Ogg eating an edible thing:
move the crumbs to the holder of Ogg;
now the hunger of Ogg is 0.

Report Ogg unlocking something with something:
say "Ogg struggles a bit with [the second noun] in the lock of [the noun], but does eventually succeed. 'Hunh!' says Ogg.";
stop the action.

Carry out Ogg opening the case when the case has been open:
now the case is unopenable.

Instead of closing the unopenable case:
say "The glass panels are no longer properly seated in their tracks, and the case cannot be closed ever again."

Report Ogg opening the unopenable case:
say "Ogg forces [the case] so hard that it does break.";
stop the action.

Report Ogg opening the case:
say "Ogg yanks [the noun] open with such force that you fear for its structural integrity.";
stop the action.

Report Ogg taking something edible:
say "Ogg acquires [the noun] with a look of tender affection.";
stop the action.

Report Ogg eating something:
say "Ogg chows down on [the noun], scattering crumbs in all directions.";
stop the action.

Report Ogg eating something when the number of visible edible things is 1:
say "Ogg eats [the noun] in his trademark style. You can no longer bear to watch.";
stop the action.

After entering the case:
say "You climb inside the case, folding yourself up uncomfortably."

After locking the case with something when the player is in the case:
say "You turn the key firmly in the lock -- amazing it locks from within, but it does -- and settle yourself for a long wait, hoping this thing is not air-tight."

The maximum score is 1.

Test me with "open mesh / get key / unlock case / open case / enter case / close case / lock case / wait / wait".

★ Example **Unthinkable Alliances**

People are to be grouped into alliances. To kiss someone is to join his or her faction, which may make a grand alliance; to strike them is to give notice of quitting, and to become a lone wolf.

The following is best tested by experimentally kissing and/or attacking, and typing RELATIONS after every change to see the effect.

"Unthinkable Alliances"

Unthinkable Solutions is a room. Sophie, Daisy, Ryan and Owen are in Unthinkable Solutions.

Alliance relates people to each other in groups. The verb to help means the alliance relation.

Sophie helps Ryan. Daisy helps Ryan. Owen helps the player.

Instead of kissing someone (called the blessed one):
say "Smack!";
now the player helps the blessed one.

Instead of attacking someone (called the vilified one):
say "Smack!";
now the player does not help the vilified one.

Test me with "relations / kiss sophie / relations / hit ryan / relations".

★ Example **The Abolition of Love**

A thorough exploration of all the kinds of relations established so far, with the syntax to set and unset them.

Suppose we are modeling a complex society seething with interpersonal relations of every kind.

"The Abolition of Love"

Section 1 - Relation types

Loving relates one person to one person.

Noticing relates various people to one person.

Impressing relates one person to various people.

Fancying relates various people to various people.

Acquaintance relates people to each other.

Marriage relates one person to another.

Alliance relates people to each other in groups.

The Chapel is a room. Elizabeth, Wickham and Darcy are people in the Chapel. Mr Bennett and Mrs Bennett are people in the Chapel. Georgiana is a person in the Chapel.

The verb to love means the loving relation.

The verb to notice means the noticing relation.

The verb to impress means the impressing relation.

The verb to fancy means the fancying relation.

The verb to know means the acquaintance relation.

The verb to be married to means the marriage relation.

The verb to be related to means the alliance relation.

Elizabeth loves Darcy. Elizabeth fancies Darcy. Elizabeth notices Darcy. Elizabeth impresses Darcy.

Mr Bennett is related to Mrs Bennett and Elizabeth. Mr Bennett is married to Mrs Bennett.

Georgiana is related to Darcy.

Now we want ways to set and unset all of these relations. (In the interests of thoroughness, we may get a bit far-fetched here. It is not recommended in practice that we make the player guess the verb "traduce".)

Section 2 - Setting and Unsetting Love (1-1)

Understand "infatuate [someone] with [someone]" as infatuating it with. Infatuating it with is an action applying to two visible things.

Carry out infatuating it with:
now the noun loves the second noun.

Report infatuating it with:
say "Now [the noun] loves [a random person loved by the noun][if the second noun loves someone], while [the second noun] loves [a random person loved by the second noun][end if]."

Understand "embitter [someone] toward [someone]" as embittering it toward. Embittering it toward is an action applying to two visible things.

Carry out embittering it toward:
now the noun does not love the second noun.

Report embittering it toward:

say "[The noun] sees [the second noun] in a different light and no longer feels any affection."

Because love is a 1-1 relation, a person cannot love more than one other character at a time. Whenever we set a character to love a new person, that person ceases to love the character loved before. It is a fickle world.

One to various relations are a bit more open: we can say someone impresses multiple other characters, and our additions to the list do not override the initial ones.

Section 3 - Setting and Unsetting Impressed (1-V)

Understand "commend [someone] to [someone]" as commending it to. Commending it to is an action applying to two visible things.

Carry out commending it to:

now the noun impresses the second noun.

Report commending it to:

say "[The second noun] takes a very decided interest in [the noun]."

Understand "traduce [someone] to [someone]" as traducing it to. Traducing it to is an action applying to two visible things.

Carry out traducing it to:

now the noun does not impress the second noun.

Report traducing it to:

say "[The second noun], hearing your story, decides not to be at all impressed with [the noun]."

And because this is a one-to-various relation, we can also make statements which set multiple relations at once, so:

Understand "celebrate [someone]" as celebrating. Celebrating is an action applying to one visible thing.

Carry out celebrating:

now the noun impresses every person.

Report celebrating:

say "[The list of people who are impressed by the noun] take a very decided interest in [the noun]."

Understand "slander [someone]" as slandering to. Slandering to is an action applying to one visible thing.

Carry out slandering to:

now every person is not impressed by the noun.

Report slandering to:

say "Now [the noun] impresses [the list of people who are impressed by the

noun]."

Note that the above unsetting is not equivalent to "now the noun does not impress every person" -- which would be ambiguous in spoken English, as well. Various-to-one relations are similar:

Section 4 - Setting and Unsetting Noticing (V-1)

Understand "draw the attention of [someone] to [someone]" as drawing the attention of it to. Drawing the attention of it to is an action applying to two visible things.

Carry out drawing the attention of it to:
now the noun notices the second noun.

Report drawing the attention of it to:
say "[The noun] glances thoughtfully in the direction of [the second noun]."

Understand "distract [someone] from [someone]" as distracting it from. Distracting it from is an action applying to two visible things.

Carry out distracting it from:
now the noun does not notice the second noun.

Report distracting it from:
say "You distract [the noun] from [the second noun]."

Understand "draw attention to [someone]" as drawing attention to. Drawing attention to is an action applying to one visible thing.

Carry out drawing attention to:
now every person notices the noun.

Report drawing attention to:
say "You quickly cause everyone to attend to [the noun]."

Understand "outshine [someone]" as outshining. Outshining is an action applying to one visible thing.

Carry out outshining:
now every person does not notice the noun.

Report outshining:
say "You quickly distract everyone from [the noun]."

Section 5 - Setting and Unsetting Fancying (V-V)

Understand "flatter [someone]" as flattering. Flattering is an action applying to one thing.

Carry out flattering:
now every person fancies the noun.

Report flattering:

say "You draw down universal admiration for [the noun] by casting him or her in a flattering light."

Understand "unflatter [someone]" as unflattering. [Okay, okay, but it's four am.]
Unflattering is an action applying to one thing.

Carry out unflattering:

now every person does not fancy the noun.

Report unflattering:

say "[The noun] gives everyone a universal disgust."

Understand "admire [someone]" as admiring. Admiring is an action applying to one thing.

Carry out admiring: now the player fancies the noun.

Report admiring: say "You find you rather fancy [the noun]."

Understand "loathe [someone]" as loathing. Loathing is an action applying to one thing.

Carry out loathing: now the player does not fancy the noun.

Report loathing: say "You take [the noun] in disgust."

Understand "cause chaos" as causing chaos. Causing chaos is an action applying to nothing.

Carry out causing chaos:

now every person fancies every person.

Report causing chaos: say "Now everyone fancies everyone else, which is quite an inconvenient state of affairs."

Understand "relieve chaos" as relieving chaos. Relieving chaos is an action applying to nothing.

Carry out relieving chaos:

now every person is fancied by no one.

Report relieving chaos: say "Now no one fancies anyone, which is safe but tedious."

Our options for setting and unsetting symmetrical relations are more limited again:

Section 6 - Setting and Unsetting Marriage (1-1 Symmetrical)

Understand "marry [someone] to [someone]" as uniting it in matrimony with.
Uniting it in matrimony with is an action applying to two visible things.

Carry out uniting it in matrimony with:
now the noun is married to the second noun.

Report uniting it in matrimony with:
say "You perform the marriage of [the noun] to [the second noun], joining them to the family of [a list of people related to the noun]."

Understand "divorce [someone] from [someone]" as divorcing it from. Divorcing it from is an action applying to two visible things.

Check divorcing it from:
if the noun is not married to the second noun, say "[The noun] is not married to [the second noun] anyway." instead.

Carry out divorcing it from:
now the noun is not married to the second noun.

Report divorcing it from:
say "[The noun] is now not married to [the second noun]."

When we unset the symmetrical relation on one side, it is automatically set or unset on the other. It is not necessary to say "the second noun is married to the noun" or "the second noun is not married to the noun", even though that is the case.

Section 7 - Setting and Unsetting Acquaintance (V-V Symmetrical)

Understand "introduce [someone] to [someone]" as introducing it to. Introducing it to is an action applying to two visible things.

Carry out introducing it to:
now the noun knows the second noun.

Report introducing it to:
say "You introduce [the noun] to [the second noun]. Now [the noun] is acquainted with [the list of people who are known by the noun], and [the second noun] is acquainted with [the list of people who are known by the second noun]."

Understand "announce [someone]" as announcing. Announcing is an action applying to one visible thing.

Carry out announcing:
now every person knows the noun.

Report announcing:
say "You announce [the noun] to the whole assembled company."

Understand "ostracise [someone]" as ostracising. Ostracising is an action applying to one visible thing.

Carry out ostracising:
now every person does not know the noun.

Report ostracising:

say "You cause everyone present to forget and pretend not to be acquainted at all with [the noun]."

And finally, setting groups:

Section 8 - Setting and Unsetting Familial Relations (Groups)

Understand "make [someone] adopt [someone]" as forcing it to adopt. Forcing it to adopt is an action applying to two visible things.

Carry out forcing it to adopt:

now the noun is related to the second noun.

Report forcing it to adopt:

say "Now [the second noun] is related to [the list of people related to the second noun]."

Understand "make [someone] disown [someone]" as forcing it to disown. Forcing it to disown is an action applying to two visible things.

Carry out forcing it to disown:

now the second noun is not related to the noun.

Report forcing it to disown:

say "Now [the second noun] is related to [the list of people who are related to the second noun], and [the noun] is related to [the list of people who are related to the noun]."

Notice that when we say "the second noun is not related", we remove that person from the group: they are now in a separate group of their own, while the rest of the group's members remain related to one another.

And finally, a long litany of test cases, complete with the relations lists:

Test acquaintance with "relations / introduce darcy to elizabeth / introduce darcy to wickham / announce mr bennett / relations / ostracise wickham / introduce georgiana to wickham / relations".

Test impression with "commend georgiana to elizabeth / relations / celebrate Mrs bennett / relations / traduce mrs bennett to darcy / relations / slander mrs bennett / relations".

Test notice with "draw the attention of darcy to elizabeth / relations / draw attention to mr bennett / relations / distract darcy from mr bennett / relations / outshine mr bennett / relations".

Test love with "embitter elizabeth toward darcy / relations / infatuate elizabeth with wickham / relations".

Test marriage with "marry elizabeth to darcy / relations / divorce elizabeth from darcy / relations".

Test alliance with "make mr bennett adopt georgiana / relations / make mrs bennett disown georgiana / relations".

Test fancying with "admire elizabeth / relations / loathe elizabeth / relations / flatter elizabeth / relations / unflatter elizabeth / relations / cause chaos / relations / relieve chaos / relations".

Test me with "test acquaintance / test impression / test notice / test love / test alliance / test fancying / test marriage".

436

Example **Lugubrious Pete's Delicatessen**

WI

In this evocation of supermarket deli counter life, a list is used as a queue to keep track of who is waiting to be served.

First, to set the scene:

"Lugubrious Pete's Delicatessen"

The Supermarket is west of the Delicatessen Counter. Lugubrious Pete is in the Delicatessen. "Lugubrious Pete, dolefully slicing meats and cheeses, serves at the counter." Alice, Beth, Gemma, Delia, and Eliza are women in the Supermarket.

The deli queue is a list of objects that varies.

Two processes compete here: one that fills the queue, the other which will empty it. The first process is the one which brings shoppers in to the counter, joining the back of the queue, which is where "add ... to ..." puts new entries by default:

```
Every turn when a woman is in the Supermarket and a random chance of 2 in 3 succeeds (this is the customer arriving rule):
  let the customer be a random woman in the Supermarket;
  now the customer is in the Delicatessen;
  if the player is in the Supermarket, say "[The customer] walks into the Delicatessen.";
  if the player is in the Delicatessen, say "[The customer] comes in from the Supermarket, and [if the number of entries in the deli queue is 0]can't believe her luck. The counter is free![otherwise]resignedly queues behind [the deli queue].";
  add the customer to the deli queue.
```

The competing process is the one which serves shoppers and thus gets rid of them again: unfortunately, it is slower. But Pete is fair if inefficient, and serves the customers in strict order of arrival. Each served customer is removed from the front of the list, and the others therefore all move up a place.

```
Every turn when the number of entries in the deli queue is not 0 and a random chance of 1 in 3 succeeds (this is the customer being served rule):
  let the customer be entry 1 of the deli queue;
  if the player is in the Delicatessen, say "Pete gives a droopy expression as he serves [customer], who nevertheless brightens and leaves.";
```

if the player is in the Supermarket, say "[customer] emerges cheerfully from the Delicatessen Counter, and goes about her regular shopping.";
now the customer is in the Supermarket;
remove entry 1 from the deli queue.

Instead of waiting in the Delicatessen when the number of entries in the deli queue is not 0, say "Time passes, for [deli queue] quite as much as for yourself."

Test me with "wait / wait / wait / east / wait / wait / wait / wait / wait".

That completes the example, but here is a variation to show that queues need not empty from the front. The Deli already looks a pretty sexist establishment, with the customers all being women, but it is about to get a whole lot worse:

Modesty is a kind of value. The modesties are positively prim, buttoned up, modest, flirty, revealing and downright immodest. Every woman has a modesty. Alice is positively prim. Beth is downright immodest. Gemma is modest. Delia is flirty. Eliza is revealing.

We could then rewrite the service rule like so:

Every turn when the number of entries in the deli queue is not 0 and a random chance of 1 in 3 succeeds (this is the customer being served rule):
let Pete's preference be the deli queue;
sort Pete's preference in reverse modesty order;
let the customer be entry 1 of Pete's preference;
let the first in line be entry 1 of the deli queue;
if the player is in the Delicatessen, say "[if the customer is the first in line]Pete gives a droopy expression as he serves [the customer], who nevertheless brightens and leaves.[otherwise]Outrageously, Pete scans the queue, notices [the customer] in her [modesty of the customer] clothes, and serves her next, while [the first in line] glares at him.";
if the player is in the Supermarket, say "[The customer] emerges cheerfully from the Delicatessen Counter, and goes about her regular shopping.";
now the customer is in the Supermarket;
remove the customer from the deli queue.

It is now heartbreakingly difficult for Alice to obtain her sliced chorizo sausage.

176



Example Strictly Ballroom

WI

People who select partners for dance lessons each turn.

Many simple repetitions can effectively be done with a "now..." instruction: it is quicker to say

now every person is angry

than

repeat with offended party running through people:
now the offended party is angry.

Repeat comes in handy when we have something a bit more complicated to do with each item:

"Strictly Ballroom"

A person can be alert or occupied. A person is usually alert.

When play begins:
now the player is occupied.

Dance is a kind of value. The dances are waltzes, polkas, cha-chas, charlestons, fox-trots, tangos, lambadas, and two-steps.

The current round is a dance that varies.

Manner is a kind of value. The manners are swiftly, primly, energetically, suavely, seductively, elegantly, and badly.

Every turn: now the current round is a random dance.

Every turn:
repeat with dancer running through people who are not the player:
if dancer is alert:
now dancer is occupied;
let partner be a random alert person who is not the dancer;
if partner is a person:
now partner is occupied;
say "[The dancer] [the current round][if a random chance of 1 in 5 succeeds] [a random manner][end if] with [partner]. ";
otherwise:
say "[paragraph break][The dancer] is forced to be a wallflower. Poor [dancer]. ";
say paragraph break.

Notice we did not say "repeat with dancer running through alert people who are not the player...". This is because Inform would draw up a list of alert people at the beginning of the repeat, and not take into account which people became occupied partway through the repetition. If we want to make sure that each person dances only with one other person, we have to continue checking alertness each time we run through the repetition.

After all the partners are assigned, we can set up for the next turn by making everyone alert again, and for this we do not need "repeat":

Every turn: now every person is alert; now the player is occupied.

Before doing something to someone: now the noun is occupied.

Before doing something when the second noun is a person: now the second noun is occupied.

Instead of doing something to someone: say "You successfully distract [the noun]."

The Pacific Ballroom is a room. "A rather utilitarian space at the moment, since this is a class and not a party." Timmy, Tommy, Joey, George, Mary, Martha, Yvette, McQueen, Linus, and Patricia are people in the Pacific Ballroom.

Test me with "z / ask linus about blanket / z / z".

353



Example Emma

WI

Social dynamics in which groups of people form and circulate during a party.

To start with, let's understand "room" to mean "a group of people talking". These groups can grow and shrink as people come and go, so we'll want to name and rename them; and we're also going to need some rules to motivate people moving around, and a description to narrate how they behave when we're with them.

"Emma"

by the banquet table is a room. at the corner is a room. next to the doorway is a room. by the window is a room.

Social clump is a kind of value. The social clumps are vacancy, lone person, couple, cluster, group.

A room has a social clump. Understand the social clump property as describing a room.

Before printing the name of a room:
say "a [social clump]".

After looking:
assign clumping;
say "Elsewhere in the room, you can see [the list of rooms which are not the location]."

Understand "go to [any room]" as joining. Joining is an action applying to one visible thing. Carry out joining: move player to the noun. Report joining: do nothing.

Understand "examine [any room]" as looking toward. Looking toward is an action applying to one visible thing. Carry out looking toward a room: say "In that direction you see [a list of other people in the noun]."

When play begins: assign clumping. Every turn: assign clumping.

To assign clumping:
repeat with space running through rooms:
now the social clump of the space is vacancy;

if the space contains exactly 1 person, now the social clump of the space is Lone person;
if the space contains exactly 2 people, now the social clump of the space is Couple;
if the space contains more than 2 people and the space contains fewer than 5 people, now the social clump of the space is cluster;
if the space contains more than 4 people, now the social clump of the space is group.

The room description heading rule is not listed in the carry out looking rules.

A person has a number called longevity. The longevity of a person is usually 0. A person can be active or passive.

Definition: a person is other if it is not the player.

Every turn:

repeat with mover running through other people:
now the mover is active;
increment the longevity of mover;
if longevity of mover is greater than 3 or the mover is bored:
assign value of spaces for the mover;
let destination be the nicest room;
if the destination is not the location of the mover:
if the player can see the mover, say "[The mover] makes excuses and drifts off to join [the destination].[paragraph break]";
move the mover to the destination;
now the mover is complacent;
now the longevity of the mover is 0;
if the player can see the mover, say "[The mover] wanders over.
[paragraph break]";
assign clumping;
now mover is passive.

A room has a number called attractiveness.

Definition: a room is nice if its attractiveness is 1 or more.

To assign value of spaces for (mover - a person):

repeat with space running through rooms:
now attractiveness of the space is 0;
repeat with figure running through people in the space:
if the mover is bored, decrease attractiveness of the space by 2;
if the mover likes the figure, increment attractiveness of the space;
if the mover dislikes the figure, decrement attractiveness of the space;
if the mover desires the figure, increase attractiveness of the space by 2.

Liking relates various people to various people. The verb to like means the liking relation.

Disliking relates various people to various people. The verb to dislike means the disliking relation.

Attraction relates various people to various people. The verb to desire means the attraction relation.

Mr Weston, Mr Woodhouse, Mr Elton, Mr Knightley, and Frank Churchill are men. Mrs Weston, Mrs Bates, Miss Bates, Harriet Smith, Emma Woodhouse, and Jane Fairfax are women.

Harriet Smith likes Mr Elton. Harriet Smith desires Mr Elton. Harriet Smith likes Emma Woodhouse.

Mr Elton desires Emma Woodhouse.

Emma Woodhouse likes Harriet Smith and Mr Knightley. Emma Woodhouse dislikes Jane Fairfax.

Mr Knightley likes Emma Woodhouse, Mr Weston, and Mrs Weston. Mr Knightley desires Emma Woodhouse.

Jane Fairfax desires Frank Churchill. Jane Fairfax likes Frank Churchill.

Frank Churchill desires Jane Fairfax and Emma Woodhouse. Frank Churchill likes Jane Fairfax.

Miss Bates likes Jane Fairfax, Emma Woodhouse, and Mrs Bates.

Mr Weston likes Frank Churchill, Emma, Knightley, and Mrs Weston.

Mrs Weston likes Frank Churchill, Emma, Knightley, and Mr Weston.

Mrs Bates likes Miss Bates.

A person can be complacent or bored.

When play begins:

repeat with character running through other people:

let space be a random room;

move character to space.

And now we use writing a paragraph about... to describe character behavior in groups, when we join them:

Rule for writing a paragraph about Frank Churchill:

if the location contains a woman (called flirt) who is desired by Frank:

say "[Frank Churchill] is talking with great animation and slightly more than becoming warmth to [the flirt][if an unmentioned other person is in the location], while [the list of unmentioned other people in the location] look on with varying degrees of amusement or irritation[end if].";

repeat with character running through people in the location:

if the character is not Churchill and the character is not the flirt, now the character is bored.

Rule for writing a paragraph about Mr Elton:

if the location contains an unmentioned woman (called flirt) who is desired by Elton:

say "[Mr Elton] hangs on the sleeve of [the flirt], offering an assortment of studied gallantries that make you wonder about his good sense.";

repeat with character running through people in the location:

if the character is not Elton and the character is not the flirt, now the character is bored.

Rule for writing a paragraph about Harriet Smith:

if the location contains Emma and Emma is unmentioned:
say "[Harriet] and [Emma] are conversing in low tones -- Harriet, apparently, being too shy to speak so that everyone can hear her."

Rule for writing a paragraph about Mr Knightley:

if the location contains an unmentioned man (called the listener) who is not Mr Knightley:

say "[Mr Knightley] is speaking with [the listener] about agricultural matters.";
now the listener is complacent.

Rule for writing a paragraph about Miss Bates:

say "[Miss Bates] is giggling about the weather[if an unmentioned other person is in the location]. This does not seem to compel the interest of [the list of unmentioned other people in the location][end if].";

repeat with character running through people in the location:
if the character is not Miss Bates and character is not Mrs Bates, now the character is bored.

Since this is just an example, we'll stop here, but there's no reason we couldn't write such paragraphs for everyone.

Test me with "z / z / z / look / x corner / x doorway / x window / x table / go to the table".

356



Example Happy Hour

WI

Listing visible characters as a group, then giving some followup details in the same paragraph about specific ones.

Often it is best to have an entire paragraph about the characters present in a room, but suppose we're narrating a large party with a lot of people moving around. In that case, it might be better to list everyone together, then add a few salient details by way of follow-up, like this:

"Happy Hour"

Before listing nondescript items:

say "You can see [a list of people who are marked for listing] here. ";
repeat with named party running through people:
now the named party is not marked for listing;
let count be the number of visible other people who are carrying something;
if count is 0:
say paragraph break;
continue the action;
let index be count;
repeat with holder running through visible other people who are carrying something:

```
if index is count, say "[The holder]";
otherwise say "[the holder]";
say " has [a list of things carried by the holder]";
decrement index;
make delimiter index of count.
```

The next part could be simpler, but for rigor we will write it in such a way that it will work whether or not the serial comma is in use. This requires some extra work.

To make delimiter (index - a number) of (count - a number), continuing or halting:

```
if index is 0:
  if continuing, say ". [run paragraph on]";
  otherwise say ".";
otherwise if index is 1:
  if count is 2, say " and ";
  otherwise say "[optional comma] and ";
otherwise:
  say ", ".
```

To say optional comma:

```
if the serial comma option is active:
  say ", ".
```

And now the scene:

The Banquet Hall is a room. "A large cheery banner over the door (which, incidentally, vanishes when you approach it) reads: HELLO NEW INDUCTEES! WELCOME TO THE AFTERLIFE!"

Fred, George, and Larry are men in the Banquet Hall. Fred carries a dry martini. Larry carries a cream puff. Matilda and Louise are women in the Banquet Hall.

Definition: a person is other if it is not the player.

Every turn:

```
let wanderer be a random other person;
let place be the holder of the wanderer;
let next place be a random room adjacent to the place;
let the way be the best route from the place to the next place;
try the wanderer going the way.
```

The Kitchen is west of the Banquet Hall. "Dominated by a pile of dirty plates which you imagine it will be someone's privilege to wash, later." Vanessa is a woman in the Kitchen. Vanessa carries a tray. On the tray is a salmon roll. The roll is edible.

Test me with "z / look / g / g / g ".