



EAMON ADVENTURER'S LOG

NATIONAL EAMON USER'S CLUB

Volume 1 Number 4 - October 1984

Our staff

John Nelson - President and Senior Staff Writer
Bob Davis - Reviewer Extrordinaire and
generally silly person
Jeff Harris - Anybody know where he is?
Dan Cross - Semi-local testor and kibitzer
Thomas Zuchowski - Bug reporter, 1st class
Gary "Norbert" Flanagan - Art director -
Omaha Office

The Dover Boys - True Barbarians

A word from the Editor ...

Some unwary adventurers have been burned. We distribute adventure disks that are as bug free as possible, although we admit problems do occur sometimes. Some people go for the cheaper sources of adventures and end up getting what they pay for (and less.) We recently heard of several cases where people had ordered from these cheap sources (to remain nameless) with the result being about half of the disks they received were totally blank. The warranty? NONE! You pays yer money and you takes yer chances. But not us! We try to compete with these sources, but some of them buy their diskettes by the boxcar full and we can't match that volume.

We tried to just fix bugs as they occurred within the adventures, but people sent in bugs from adventures that they received from other sources. We ended up chasing our tail stamping out bugs that we'd stamped out three years ago.

As a result, we have made a policy decision and would like to explain our actions. We print bug fixes to the versions of the adventures that we distribute. There is no way we can fix a bug to a version that we don't have. We have enough problems doing the work we have now. So if you buy your diskettes from other sources and they don't work, we can't help if it is not the same version we have. We would be glad to look into it and fix it, but there's no way we can afford to take the time to fix the same bug eighty times!

To solve these problems, we have taken two steps:

1. We built a data base to keep track of what adventures we are distributing and to whom.
2. We are making up and maintaining a list of revised dates for all programs and adventures. We will print this list in futures issues so you can compare your versions to the ones we have.

If you write to us about a bug, tell us where you got the adventure. If you got it from us, we will check into it right away. If not, we are still willing to help you; but if the copy is old, the solution might be to order a new copy. (There may have been dozens of changes since you got your copy!)

P.S.

A small delay may be noticed when ordering or corresponding just before newsletter time. This is unavoidable and your patience during our peak times is appreciated.

All National Eamon User's Club offices in the continental United States will be closed from Friday, December 21, 1984, through Wednesday, January 2, 1985, for the Christmas and New Year Holidays. Offices in Iran, Bahrain and Moscow will remain open. (Drive-up windows will be open half-days.)

CLUB NEWS

Tournament Plans have been indefinitely postponed. The time needed to set one up is not currently available and low interest in it does not justify setting a higher priority on it.

The notebook idea has struck a lot of members fancy and we would like to proceed with it. We need a definite commitment to go ahead with the making of the notebooks. The cost of the notebooks to members will be \$10. each plus postage and handling. They are 100% Vinyl with padded covers and backs, silkscreened with the Eamon Dragon and lettered in Old English script along the spine. They will have pockets on the inside for placing notes and will be a standard size (8 1/2 x 11) 1" thick. If we can sell enough of them, we may be able to get them cheaper, but we need a commitment for 100 notebooks (minimum order from factory). The documentation will be distributed separately. Some members have expressed an interest in buying the notebooks and using vinyl diskette pages for holding their Eamon diskettes. For this reason (and because the documentation is not ready yet) we decided to sell the notebooks and documentation separately. Because the manufacture of the notebooks is expensive, we will need a firm commitment and pre-payment by November 30th. If not enough orders are received by that date, we will refund your money.

BUG BYTES

DDD 6.0 - DUNGEON EDIT program - The Generate Dead Bodies routine doesn't work correctly. The variables are not initialized and the artifact counter is not set up by the number of bodies created. To correct this, make the following changes in the DUNGEON EDIT program.

Line 34510 after INPUT A\$ insert
FOR X2 = 1 TO 6:
INPUT MDZ(X2): NEXT

Line 34540 add
: NZ(2) = NZ(2) + 1
to the end of the line

insert a new line:

```
34535 FOR X2 = 1 TO AF + 4: ADZ(X2) = 0:  
NEXT: ADZ(2) = 1: ADZ(3) = MDZ(6)
```

Another problem in this program is if you accidentally hit <RETURN> instead of a valid value while editing a monster, the program beeps and you have to start over with that monster. To solve this, split line 14050 into two lines so that there are two lines as follows :

create a new line: 14040 FOR M = 1 TO MF

remove the above code from line 14050

A third bug in this program is a minor one, but should be fixed. When you are in LOCK mode, the change room function goes to the add room routine after the first one.

new line : 1150 GOTO 800

In the MAIN PGM program, if you do a LOOK in the room containing an exit, then try to use that exit at any time after that, the program bombs on OUT OF DATA.

To fix this, change the following line in MAIN PGM.

```
3540 PRINT DK$;"READ EAMON.ROOMS,R";ROOM  
NX = 0: FOR X = 1 TO ND: INPUT RDZ(X):  
IF RDZ(X) > -99 AND VZ(ROOM) > 1 THEN  
RDZ(X) = ABS(RDZ(X))
```

The drink routine will bomb if you drink a healing potion when you are not injured. You can correct this by changing line 22130 as follows:

```
22130 add IF MDZ(0,13) < 0 THEN MDZ(0,13) = 0  
to the end of the line
```

DUNGEON LIST

The line count wasn't correct when printing rooms. Add LP = LP + 3 to line 2100 and LP = LP + 2 in line 2120.

On DDD 6.0 there is a text file that will convert the base program from a six direction dungeon to a ten direction dungeon. This file is called CONVERT MAIN > MAIN 10. Some of the earlier versions (prior to July 20, 1984) neglected to change line 290 to a ten direction version. This can be corrected by following these steps:

1. Type NEW to remove any old programs from memory
2. Insert DDD 6.0 and type
EXEC CONVERT MAIN > MAIN 10
3. Type 10 D\$ = CHR\$(4):
?D\$;"OPEN CONVERT MAIN > MAIN 10":
?D\$;"WRITE CONVERT MAIN > MAIN 10":
POKE 33,33: LIST 50,:
?D\$;"CLOSE":END
4. Type 290 ON C GOTO 3000,3000,3000,3000,3000,
3000,3000,3000,3000,3000,4000,
4000,5000,6000,6000,7000,8000,
9000,10000,11000,12000,13000,
14000,15000,16000,23000,17000,
18000,19000,20000,21000,22000,
24000,25000,26000
5. Type UNLOCK CONVERT MAIN > MAIN 10
6. Type DELETE CONVERT MAIN > MAIN 10
7. Type RUN
8. Type TEXT

Adventure #15 - Oops! We made a mistake last issue. line 1920 should be as below:

```
1920 DATA NORTH,SOUTH,EAST,WEST
      ,UP,DOWN,GET,TAKE,DROP,PUT,L
      OOK,EXAMINE,ATTACK,FLEE,GIVE
      ,INVENTORY,BLAST,HEAL,POWER,
      SPEED,SMILE,WAVE,SAY,READ,USE
      ,READY,DRINK
```

Also line 19000 should have GOTO 300 added to the end of the line.

Adventure #71 - Operation Crab Key

```
line 1140 change DIM VZ(96)
           to DIM VZ(NR)
```

EAMON NEWS

On a serious note, we have some bad news about the man we call the Grandfather of Eamon.

Donald Brown, the father of Eamon, was inspired to write Eamon at the urging of a good friend: Bill Fesselmeier who was a great fantasy role-playing fan and Baron of the local SCA (Society for Creative Anacronism) in Des Moines. Bill was killed September 14th in an automobile accident along with his wife and two of their friends. A fifth person was in critical condition at a local hospital here.

The Baron was buried in full battle armour: his friends from the SCA in full dress armour at the funeral.

I think it would be appropriate for all Eamon fans to doff their helmets, and dedicate a moment of silence for the person responsible for their Wonderful World of Eamon.

Then, let's all go out there and slay a Dragon for the Baron!

I separated the above sad news from the below happy news for obvious reasons.

New Eamon adventures:

The following are new Eamon adventures since our last issue:

- 77. Temple of the Trolls by John Nelson
- 78. The Prince's Tavern by Bob Davis

We also received three other adventures and are testing them. They will be ready before the next issue is out.

TERMINOLOGICAL INEXACTITUDES

Things appearing in this column must not be confused with actual facts.

After tallying the results of the Questionnaires we sent out, we have taken a hard look at the "What things would you like to see improved in Eamon?" question and have decided to rewrite Eamon to have all of the improvements! All we have to do is put Hi-Res 256 color graphics in it with animation routines, 12 octave sound, allow room on the disk for 44000 rooms, (twelve levels deep) 64000 artifacts, 94000 intelligent monsters (capable of intellectual pursuit) 4 million new built in features that will run on a 48K machine! -- NO PROBLEM!!! Near as we can tell, it would only take a 400 megabyte machine with disk access speed of a trillionth of a nano-second and a capacity of 400 billion K! Anybody out there know some REALLY GOOD programmers? They'd have to be willing to work 24 hours a day for the next 46 years and of course it is all volunteer, since Eamon is non-proprietary.

Did you know most of you are named David?

ADVENTURE TIPS

Watch out for this one! In Quest for Trezore if you manage to get Zobar's power sceptre away from him, and then manage to get into the Chamber of Trezore and do a power spell, your diskette will melt in the drive! This is particularly hard on the read head.

Using covert tactics seldom seen in the free world, I was able to get ahold of Bob's first adventure - The Prince's Tavern. It's the story of a young prince who has to work his way through law school while supporting an elderly grandmother.

QUESTIONS & ANSWERS

In this column we answer commonly asked or interesting questions we receive.

Q. I bought about a dozen adventures from another distributor about three weeks ago and about half of them are blank. I tried to get this distributor to replace them, but I was told there was no warranty and that the disks are sold "as is". Can you tell me how much it would cost to replace them for me?

A. The club will re-copy an Eamon diskette for our members at the cost of \$3.00 plus .50 for postage, if you send the diskette to us. If you previously purchased the diskette from us and are getting an upgraded copy, we charge only \$2.00 plus .50 postage. If the copy you got from us is bad (a bad COPY - don't confuse this with a bug - bugs are not covered), we will re-copy it free, providing you send us the diskette packaged to prevent it from damage in transit. If your copy is not actually a bad copy, but contains bugs, please report the bugs to us and we will clean them up and provide you with a fix.

Q. I recently obtained a copy of the Death Star and it is riddled with bugs. [Bug descriptions omitted..] I would appreciate it if you could fix these for me or send me a new copy.

A. The bugs you described do not exist on the club copies of the diskette. I checked the records and have found no bugs such as those you report. I tested the adventure and they do not occur on our copy. That must mean one of two things. Either the club copy was fixed years ago (before we kept records on bugs) or they never existed on our copy. It is possible someone was playing around with the adventure and cobbled it up. I would suggest you order a new copy.

Last issue we reported that the most you would have to spend to bribe a monster was 100 gold pieces. This is not exactly accurate. This is the most you would have to spend to bribe a monster with a friendliness of 50% or more. When we noticed this error, we immediately calculated how much you would need for other friendliness ratings. The maximum amount you would ever have to spend is 9900 gold pieces (for a monster with a friendliness of 1) For a monster with a friendliness of 2 it drops to 4800.

DDD 6.0 presents some interesting problems for the player. Because this new version allows such sneaky-isms as artifacts in the room that are not listed as artifacts, the player must be extremely watchful and clever. This helps make the adventures harder and the player must be able to figure out more complex problems. For example in the Picnic in Paradise adventure, one of the rooms is described as having a desk in the room. The desk is not actually listed as a separate artifact unless you examine it. Also the desk, since it really is in the room could be a necessary artifact. For example, you might be able to open it. Other things might be more common in a version 6.0 dungeon. Things such as secret passages, locked doors or gates, monsters bound in the room, healing potions, and things inside other things. This is because these are the standard available features of a version 6.0 dungeon and they are very simple for the designer to install.

To combat some of this, the adventurer will have to be extremely cautious. The designer also must be much more careful. For example, a room could be described as having a large statue in the center of the room and pillars on each side. In version 5.0 and before, the player would not have to consider these items when playing the adventure, except that it would make the description of the room slightly more interesting. In version 6.0, it would be possible to have the statue actually be an artifact in the room, without the program listing the statue as a separate artifact. Even the pillars could be artifacts. The problem this presents for the adventurer is that he has to search very diligently and watch the descriptions very carefully, but if he gets into the habit of examining every item in every description, it'll take him years to get through a very good descriptive adventure. Actually, this also presents a problem for the designer as well. He should be careful enough with his descriptions so that he won't mislead the player.

The big problem then is how to determine if an adventure is 6.0, 5.0 or earlier. Right now the only version 6.0 dungeons in production are #48 - Picnic in Paradise and #77 - Temple of the Trolls.

DESIGNER'S DEN

To make a gate block a passage, you can do it one of several ways. Version 6.0 of the Dungeon Designer Diskette supplies the designer with a method that requires no programming, so we will not discuss this version here. If you have version 6.0, simply use the method provided.

For version 5.0 and prior the following may help make the design easier for the author.

Many different ways can be found by looking at how others have done it in various adventures. If you would like to look at how others have done it, the following adventures are among those that contain gates where you have to have the key or some kind of a passcard to get through. The list is by no means complete, but it will demonstrate a wide variety of methods.

Lair of the Minotaur - Iron Gate
Devil's Tomb - Steel Door
Abductor's Quarters - brick wall
Quest for Trezore - Brass Door
Hogarth Castle - cell door
The Black Death - Iron Door, safe,
desk, file cabinet
Nuclear Nightmare - gate, bubble, vault
Assault on the Mole Man - gates
Revenge of the Mole Man - gates

Here we will go through one of the easier methods and I think one of the better ways to handle locked doors.

The method you use will depend on how you want to activate the opening of the gate. Some adventures will have a special UNLOCK command, some will simply check to see if you are carrying the key and if so, let you pass. Others will have you use an OPEN command and if you are carrying the key, let you go and if not, print a message that the gate or door is locked.

The easiest way is to simply check to see if the person is carrying the key, and having two artifacts for the gate or door. One artifact would be the gate or door when it is open and the other would be the gate or door when it is closed. To operate the door the program would have a fixed number for the door that it would check for.

For example, if the key were artifact #6, and the closed/locked door were artifact #7 and the open/unlocked door were artifact #8, the program could work like this:

1. In the move routine:

```
3035 IF ADZ(7,4) = ROOM AND D = 3 THEN  
PRINT "THE DOOR IS LOCKED. YOU'LL  
NEED A KEY":GOTO 100
```

The variable ADZ(7,4) is the room number of the locked door. The D = 3 check means the door will block passage only to the east. (East is the 3rd direction in N S E W U D in the command table and D is always the direction of movement). It is compared to the Room number to see if the door is locked. (If it is in the room, the door is automatically locked, because the OPEN routine will do this:)

2. An OPEN command will be needed to get the locked door (artifact #7) out of the room and the open door (artifact #8) into the room:

```
20000 REM /// OPEN COMMAND
```

First, check to see if they typed OPEN DOOR or OPEN (name of door) instead of OPEN LUNCH BOX or some such silliness:

```
2010 IF (S$ = AN$(7) OR S$ = "DOOR")  
THEN GOTO 20100
```

If they did say something crazy, print an error message:

```
20020 PRINT "I DON'T UNDERSTAND.":GOTO 100
```

Now, for the routine that will verify everything is o.k. First, we have to make sure the door is really in the room, and if not, print an error message:

```
20100 IF ADZ(7,4) <> ROOM THEN  
PRINT "THERE'S NO DOOR HERE!":  
GOTO 100
```

Now check to see if they have the key. If the room number of the key (artifact #6) is = -1, then the person does have the key. If this is the case, you open the door by removing the locked one (changing it's room number to 0) and bringing the open one (artifact #8) into the room by setting it's room number equal to ROOM: (cont'd)...


```

20110 IF ADX(6,4) = -1 THEN
  PRINT "YOU'VE OPENED THE DOOR!"
  :ADX(7,4) = 0: ADX(8,4) = ROOM:
  GOTO 300

```

If the person doesn't have the key, we have to print a message:

```

20120 PRINT "IT'S LOCKED. YOU DON'T HAVE A KEY
THAT FITS.":GOTO 100

```

The door or gate has now been installed and should work with no problems. The adventurer will come to the door or gate, try to proceed in that direction and get the message YOU CAN'T GO THAT WAY! The description of the gate or door should explain that the east exit (or whichever exit it is) is sealed off by the gate or door. The adventurer may then try the open command. If he has the key, the gate will open.

An alternative would be to have a USE command for using the key. This would mean the adventurer would have to know which key to use or would have to experiment. This could be done to make it more challenging for the player.

The USE command may look something like this:

```

22000 REM /// USE COMMAND

```

First we have to be sure the player provided a subject to be used. (i.e. KEY or whatever the name of the item he is trying to use.)

```

22010 GOSUB 4900 :IF S$ = "KEY" OR
  S$ = AN$(n) THEN 22100

```

If there is anything else in the adventure that can be used, the coding to check for this would go between 22010 and 22099.

Now we need to print a message in case they try to use something the program doesn't recognize:

```

22099 PRINT "I DON'T KNOW HOW TO USE THAT."
  :GOTO 100

```

The actual use key portion of the routine:

```

22100 REM /// USE KEY

```

See if the adventurer is really carrying the key:

```

22110 IF ADX(6,4) <> -1 THEN
  PRINT "YOU DON'T HAVE ANY KEY!"
  : GOTO 100

```

Check to see if the locked door is in this room:

```

22120 IF ADX(7,4) = ROOM THEN
  PRINT "THE DOOR UNLOCKS AND
  SWINGS OPEN!":ADX(7,4) = 0:
  ADX(8,4) = ROOM: GOTO 300

```

```

22130 PRINT "THE KEY DOESN'T FIT
  ANYTHING IN THIS ROOM."
  :GOTO 100

```

An added complication could be inserted by making the adventurer use both a USE (or UNLOCK) command and an OPEN command. This would probably require the use of a switch. A variable (for example UL for UnLock) would be set to 1 when the door has been unlocked. Then, in the OPEN command, this switch would be checked and if it is not set, it would say the door is locked. If the switch were set the open artifact (#8 in the above example) would be put into the room by setting ADX(8,4) = ROOM and the closed/locked artifact (#7 in the above example) would be moved into the twilight zone (room 0).

Another alternative - one that does not require the use of two artifacts (one open and the other closed) for the door - would be to set switches to indicate when the door is locked or unlocked. You could set up this kind of door without any kind of artifact at all. If you did it this way, instead of changing the room numbers of the open and closed artifacts, you would set the switch. Zero would be locked, 1 would be unlocked. When the adventurer comes to the locked door, the direction of movement code would be something like -1 for the direction blocked by the door. The program would then have statements such as that below in the move routine:

```

3070 IF R2 = -1 AND UL THEN R2 = nn

```

The nn in the above statement would be the number of the room to move into.

There are probably millions of ways to handle locked doors and gates in Eamon, this has been just a sampling. It should help out if you get stuck or do not wish to re-invent the wheel.

.....

Testing Tips
by Bob Davis

"Okay, now I've got the key and I'll unlock the safe and ..." - BOOP! SYNTAX ERROR LINE 21140. "Crudy Bumpkins! Now I'll have to change the line and start the testing all over - go through 43 rooms, find all the artifacts needed, kill the troll ..."

How many hours are wasted testing and retesting, and how many bugs are not detected because of this tedious method will never be known, but we can guess - many. If only there were easier and faster ways to test newly written adventures ...

Well, here they are! The following routines can be temporarily added to MAIN PGM (versions 5.0 and 6.0) to facilitate testing. After testing is complete and the code is removed, one final test should be given.

FRESH MEAT - Remark the line that deletes FRESH MEAT. This will allow you to RUN MAIN PGM anytime without having to come from the Master Diskette (after you do it the first time.) Type in the following (or insert REM if you have a line editor.)

```
1055 REM ?D$"DELETE FRESH MEAT"
```

CAN'T KILL ME - When testing a monster's toughness and/or death traps, this little routine will revive you whenever you die, setting your damage to zero. A change is needed in the ATTACK routine so the program doesn't bring in the artifact that it thinks is the adventurer's dead body (lines 7640 and 7690).

```
2005 IF DIE THEN DIE=0?:
?YOU'RE NOT REALLY DEAD.":
MDZ(0,13)=0 : ?YOU'RE FEELING
MUCH BETTER NOW. ": GOTO 210
```

Change 7640 to be;

```
7640 MDZ(DF,13)=MDZ(DF,13)+D2 :
IF MDZ(DF,13) >= MDZ(DF,1) THEN 7690
```

from;

```
7640 MDZ(DF,13)=MDZ(DF,13)+D2 :
IF MDZ(DF,13) >= MDZ(DF,1) THEN 7700
```

```
7690 IF NOT DF THEN ? : ?YOU ARE DEAD!"
: GOTO 7740
```

AUTO KILL - This will kill any monster in the room that you deem unfit to live. To activate, SAY KILL. This will set an indicator (which can be shut off by typing SAY KILL again) and then simply ATTACK monster name. For example, if you knew a tough dragon was in the next room and you needed to kill him to get past and test the next part of the dungeon you would type SAY KILL, move into the next room and type ATTACK DRAGON.

```
7295 IF TEST THEN OF = 0 : DF = M : D = 90 :
S = 5 : GOSUB 7635 : GOTO 300
```

```
16800 IF LEFT$(S$,4) = "KILL" THEN TEST =
NOT TEST : GOTO 100
```

TELEPORT - This routine will teleport you, your friends and possibly your enemies to the room of your choice. All you need to do is type SAY ROOM room number.

```
16820 IF LEFT$(S$,4) = "ROOM" THEN R2 =
VAL (MID$(S$,5)) : GOTO 3500
```

GIMME IT - This routine will give you any artifact by the artifact number without adding its weight to your carried weight. Just SAY GIMME artifact number.

```
16840 IF LEFT$(S$,5) = "GIMME" THEN
ADZ (VAL (MID$(S$,6)),4) = - 1
: GOTO 100
```

To remove these changes when testing is complete, delete the following line numbers:

```
2005, 7295, 7690, 16800, 16820, 16840
```

and lines 1055 and 7640 should be changed to:

```
1055 ?D$"DELETE FRESH MEAT"
```

```
7640 MDZ(DF,13)=MDZ(DF,13)+D2 : IF
MDZ(DF,13) >= MDZ(DF,1)
THEN 7700
```

With these test routines, testing should be faster, less tedious and (hopefully) more accurate.

SPOTLIGHT ON

#76 - Search for Yourself (8:6) - by Don Doumakes

Extra commands: SWITCH, SAVE

Playing time: 4 - 6 hours (if you don't know what you're doing - 4 to 6 weeks.)

I believe the author does an excellent job of describing this adventure; so as not to do him or his adventure any injustice, here's Don...

You were drinking too much. You know that now. If you hadn't been drinking too much, you wouldn't have made that foolish bet. Now it's too late.

The fellow looked so young, so inexperienced. He looked drunk, too. So when he bet you 500 gold pieces that he had a magic spell you couldn't get yourself out of, you naturally assumed he was bluffing.

He wasn't bluffing, and he wasn't drunk. You have awakened in an underground labyrinth. You feel tired and drawn. It's more than a hangover, though you certainly have one of those. No, it's as if a part of yourself were missing.

A part of yourself IS missing!

You have been split into two selves, and what is worse, your other self is somewhere else in this maze. You cannot survive outside this magical maze unless you merge back with your other self.

The barroom prankster was not completely merciless, however. You find that by exerting a little concentrated thought, you can transport your mind to your other self. (Type 'SWITCH' whenever you want to do this.) Unfortunately, if one self gets wounded, the other is injured simultaneously.

Remember, you must not try to leave the dungeon unless you have merged with your other self.

Bob again. Two big pluses this adventure has is 1) a very original idea and 2) the most complex maze in Eamon. It is very survivable (if you're careful), the room descriptions are generic (helping to make the maze confusing), the monsters and artifacts are interesting and the theme is refreshing, warranting special consideration.

(Ratings are given on a scale 1 to 10 with 10 highest. Format is R:D where R = rating for setting, description and plot; D = rating for difficulty, problem solving and survivability.) Playing times are based on actual tests.

#73 - The Deep Canyon (6:4) - by Kenn Blincoe

Extra commands: DIG, SET, WAIT, SAVE

Playing time: 1 - 2 hours

While walking along a plateau, you meet a man who professes to help you find your way and then promptly pushes you off a cliff into a deep canyon. Luckily, a magic spell saves you from being crushed on the hard, rocky ground below. After determining your physical state, you find yourself in a hot, dry environment with no idea how to get out of the deep canyon.

A basic escape adventure, the Deep Canyon is excellent for beginning and/or young adventurers with a small puzzle to solve before being able to leave. The descriptions are good, with some interesting effects and the designer did keep with the theme throughout the entire scenario.

I found the adventure to be easily survived using test character ALDO CELLA - hardness 15, chain armour and 55% sword ability - so go ahead and put any character through this adventure with little fear of losing him/her.

I think one hint is deserved here; the secret passages cannot be detected by the LOOK command, so the adventurer must stumble upon them. But after all, what's a secret passage if it isn't secret?

#71 - Operation Crab Key (0:3) - by Joe Vercellone

Extra Commands: None

Playing time: Seems like forever.

"You are in a charming double bedroom in modern Miami style, with dark green walls and a polished oak floor." Room after room, they're all the same. Another description the author painstakingly mullied over to set the proper mood, "Too normal to bother describing." The writer, obviously short on ideas after the first room or two, persevered to create 100 rooms of boredom (quantity vs. quality). One actual warning sums up this 'adventure': "Don't waste your time, just leave".