The Dungeon Builder

GRAPHIC ADVENTURE SYSTEM

Program and Manual

by

RICHARD PARRATT

Dream Software —
P.O. Box 64, Basingstoke RG21 2LB

Printed by GL Print, Aldermaston, Berkshire (07356) 5215

We will, of course, be pleased to market programs on your behalf.
We offer:

- Attractive payments
- Superb packaging and marketing
- National and international distribution

Send your program to:-
The Software Manager
Dream Software Ltd
P.O. Box 64
Basingstoke
Hants
RG21 2LB

The Dungeon Builder

GRAPHIC ADVENTURE SYSTEM

Program and Manual

by

RICHARD PARRATT

CONTENTS

This program may be used to write programs for commercial use. We do not ask a fee for this — we simply ask that you credit this program in your documentation and that you leave the title page as we have programmed it.

ILLUSTRATIONS

EXAMPLES

1) Starting off.

To start using the system type:

LOAD "" ENTER

with the tape at the beginning of side 1. The program will take about 10 minutes to load. When initially loaded the demonstration adventure is in the computer. To play type P

The program understands commands such as:

keep	save your current game position to tape
restore	load your previous game back
invent	give list of everything you are carrying
score	give your score
look	describe the room you're in.
quit	finish the game
n,s,e,w,up,d	direction commands
take "object"	pick up an object.

Commands may be chained together with "and"

i.e. south and take gold and north and up

The game inputs text in a similar way to Sinclair BASIC. The main difference is that the cursor is a flashing square over the character in front of the one just typed. Pressing → (CAPS SHIFT & 8) and ← (CAPS SHIFT & 5) moves the cursor backwards and forwards. Pressing DELETE (CAPS SHIFT & 0) removes the character immediately before the cursor. (Often the one just typed) Press ENTER to enter your command into the game.

The game understands more words than this but to reveal them would make life too easy.

It's a good idea to play the game for a bit to get an idea of what you are trying to achieve in the design of a game.

Typing "quit" will give the message *'Another game?'* - press Y to play again and N to return to the **outer menu** screen.

Load, Save, Verify, Modify, Create, Play game	*Outer Menu*
Open, Block, Cond, conseQuential, Modify, cReate object, Exit	*Edge Editing Menu*
Edit Score, Description, Picture Modify, cReate object Exit	*Centre Editing Menu*
Edit Name, Description, Score Commands, Position, Weight, Exit	*Object Editing Menu*
Edit Name, Condition, Actions Exit	*Verb Editing Menu*
Position? Any, Carried, Here Select.	*Position Menu*
Say, Die, Move, sTatus, Jump	*Command Menu*

MAIN MENU'S

Fig. 0

2) Simple maps and placing your player.

Having played the demo game, you will have found action takes place in a series of interchained rooms These are marked out on a grid.

To create a game you have to set up a number of rooms and provide a description for them. You also have to tell the computer which room to start off in. The example below sets up a very simple two room adventure.

If you are still in the demo game type *quit* and reply N to the 'Another game?' prompt. This will take you back to the **outer menu** where you started.

The Dungeon Builder Menus

The main menus are shown in figure 0. Each one is pictured in full the first two times they occur in the examples. Thereafter they are referred to by name only.

Full details on using menus are given later.

Example 1:

(Note: All examples start from the **Outer menu.** You should have the "The Dungeon Builder" banner displayed. If you can see a grid of cells as shown in fig 1, pressing E should take you to this stage.)

Load, **S**ave, **V**erify, **M**odify,

Create, **P**lay game

1) Press C — To create a game and destroy a previous one

Response: — Are you sure? Y/N

2) Press Y — Computer wants confirmation

Response: — The map screen (Fig1) is displayed together with the menu.

Open, **B**lock, **C**ond, conse**Q**uential,

Modify, c**R**eate object, **E**xit

Edge Editing Menu

(Please refer to the fold out diagram of fig 1 at the end of the book)

The flashing square in the top left hand corner is called the cursor and marks the current cell. Each octagonal cell represents a room with 8 walls.

3) Press → (CAPS SHIFT & 8) four times
Press ↓ (CAPS SHIFT & 6) four times
To move to the centre of cell (2,2)

Response: The cursor moves to the centre of cell (2,2) and the centre editing menu is displayed

Edit **S**core, **D**escription, **P**icture

Modify, c**R**eate object **E**xit

Centre Editing Menu

Edit **S**core, **D**escription, **P**icture

Modify, c**R**eate object **E**xit

Centre Editing Menu

4) Press D To describe a room.
Response: Enter description:

5) Type in the text: *You are in an entrance hall, a passage leads south* ENTER
—This will be printed on entry to this room.
Response: Cursor in middle of cell (2,2)

Edit **S**core, **D**escription, **P**icture

Modify, c**R**eate object **E**xit

Centre Editing Menu

6) Press ↓ (CAPS SHIFT & 6) once. — To move to the edge of cell (2,2).

Response: Cursor on south edge of cell (2,2).

Open, **B**lock, **C**ond, conse**Q**uential,

Modify, c**R**eate object, **E**xit

Edge Editing Menu

7) Press letter O - To open up a gap
Response: The cursor turns bright green.

8) Press ↓ (CAPS SHIFT & 6) once - To move to the top of cell (2,3).
Response: Cursor on north edge of cell (2,3), Edge Editing Menu.

9) Press letter O - to open up a gap for travel back.
Response: The cursor turns bright green.

10) Press ↓ (CAPS SHIFT & 6) once - to move to the centre of the cell (2,3).
Response: Cursor in centre of cell(2,3)
Centre editing menu

11) Press D - to describe the room
Response: Enter description:

12) Type in the description: *You are in a small room with a passage leaving to the north.* ENTER
Response: Cursor in centre of cell (2,3)
Centre editing menu

13) Press ↑(CAPS SHIFT & 7) three times - to move back to the centre of call (2,2).
Response: Cursor in centre of cell(2,2)
Centre editing menu

14) Press M - To modify an object.
Response: 1 .player

15) Press 1 - to select the object '.player' to be modified.
Response:

Edit Name, Description, Score
Commands, Position, Weight, Exit

Object Editing Menu

16) Press P - To modify the position of '.player'. (The starting position of the game.)
Response:

Position? Any, Carried, Here
Select.

Position Menu

17) Press H - To set the position of '.player' to Here. (The room with the cursor in)
Response:

Edit Name, Description, Score
Commands, Position, Weight, Exit

Object Editing Menu

18) Press E - To exit.
Response: Centre editing menu

19) Press E - To exit
Response:

Load, Save, Verify, Modify,
Create, Play game

Outer Menu

20) Now press P - to play the game. Test the directions you can move in. Compare these with the displayed map. Type Quit and reply N to get back to the **outer menu.**

This example should have give you a feel for the system. The following section describes in detail how to generate maps. It's a good idea to save this example on tape (See chapter 3) before continuing.

On a green background at the bottom of the screen is the following message:

> **L**oad, **S**ave, **V**erify, **M**odify,
> **C**reate, **P**lay game

Outer Menu

This is called a menu. It allows the selection of a number of options. Options are selected by pressing the highlighted letter. For example pressing P starts play of the game. Menus like this control most of the system. This particular menu is the **outer menu**. Press C to start creating a new game.
The program will display:

Are you sure? (Y/N)

This is done in a number of places to make sure you do not destroy a game inadvertently. If you have

typed in example 1 and not saved it yet, reply N and do so now. Reply Y to continue.

The main editing screen will now appear. At the top a status line gives the amount of memory free for the text and pictures of your game. Initially this is about 10,000 bytes. (Normally a byte stores one character but TDB employs a compression system giving around 1.5 characters per byte)
There is also an 'invisible' additional memory space of about 10,000 bytes that is used for storing your object descriptions and room locations.

Below this is a grid of octagonal cells. This is the map. Each cell represents a room. Since the cells are octagonal, each room has 8 walls, corresponding to movement directions N, NE, E, SE, S, SW, W, NW. A black line represents an impenetrable wall. At the moment all the walls are like this.

In the top left hand corner is a flashing square. This is called the cursor and indicates where you are currently working. The cursor is moved about with the cursor control keys. These are activated as in BASIC by pressing the CAPS SHIFT key and one of the characters 5 to 8.

The cells are numbered on a coordinate system, x coordinates are along the top of the screen and y coordinates along the left hand side, rather like a road map or A-Z street plan. The map is 40 x 40 cells but only part is visible at any one time. To see the rest move the cursor off the right hand side or bottom of the screen and the screen will scroll. If you try to move off the edge of the grid, i.e. above row 1, the computer will bleep.

At the bottom of the screen is a menu. You will notice that this changes as you move the cursor about. Position the cursor in the centre of any cell. The following menu will display.

Edit Score, Description, Picture
Modify, cReate object Exit

Centre Editing Menu

This is called the **centre editing menu.** We want to describe the room so press D. The program will then prompt:

enter description

Type in your description, in the same way that you gave a command to the game. Remember that you can use the keys←and→and DELETE to correct any mistakes. When you are satisfied press ENTER This is the description that is printed when you enter a room. You have now defined your first room. It is now time to set up a route to another room. This is done by making a hole in the wall to move through to another room. In order to do this first move to a wall of the room. A new menu will display, reading:

Open, Block, Cond, conseQuential,
Modify, cReate object, Exit

Edge Editing Menu

This is called the **edge editing menu.** To open up a gap press O for open. Move the cursor through the gap to the next room. Now press O on the other side of the wall to open up a two way thoroughfare. It is important to remove both parts of the wall, we will see later the effect of not doing this. Describe the next room as you did before. Using this method you can set up a labyrinth of rooms. Remember to open both walls of each room. (Note that you can block a gap up by pressing B)

The program needs to be told where to start. This is done by altering the special object ".player".

Move the cursor to the position you wish to start at and press M to Modify an object. A menu of objects will display with just one member, ".player". Press the highlighted number by it (normally 1) to select it.

The menu

Edit Name, Description, Score
Commands, Position, Weight, Exit

Object Editing Menu

will display. Press P to alter the position. Then press H to set the players position to "here", that is the room with the cursor in. The game will always start in the room where the object '.player' is. Press E to exit from this menu.

This returns you to an editing menu. Press E to get back to the outer menu and then P to play your created game.

If you have set the ".player" position incorrectly you will get the description:

You are in a completely featureless room.

and/or you will not be able to move anywhere. If this happens quit out and set the position of ".player" correctly, in a room with a description and passages out of it.

Summary:

1) Your map is set out on a scrolling grid.

2) A menu is a list of options.

3) Gaps in a wall may be made with the Open command.

4) A room is described with the Description command.

5) The player is an object put in place by the Modify object command.

3) Using tapes.

It is posssible to save your game on tape for later alteration or use. To do this press S from the outer menu. The program will ask for a filename. Type one in. The program then asks you to start the tape and hit a key. Press play and record on your recorder with a blank tape in place.

To check that your program has been saved correctly use the V option. This will ask for a filename and verify that the game in that file is the same as that in memory.

To reload your program use the L command from the outer menu. This first asks for confirmation, since loading a new game destroys anything already in store. It then asks for a filename and loads it.

If a tape loading error occurs the program will give an error message. Pressing [ENTER] will return you to the **outer menu.** Pressing [CAPS SHIFT & BREAK] during a tape operation will also cause a message to display and [ENTER] should be pressed.

4) Adding treasures.

Without anything in the Rooms the game is rather boring. In the demonstration game there were a number of treasures, for example the gold.

The example below creates a jewel in room 2,3. It is picked up with the command "TAKE JEWEL" and dropped with the command "DROP JEWEL".

It assumes that you have already typed in example 1. If you have example 1 saved on tape, [LOAD] it as described previously.

Example 2:

(Must be entered after example 1)

21) Press M - To modify the stored game.
    Response: Screen with cursor in NW of cell(1,1), Edge editing menu
22) Use cursor keys to move to centre of cell (2,3)
    Response: Screen with cursor in centre of cell (2,3) Centre editing menu
23) Press R - To c R eate an object.
    Response: Enter object name:
24) Type in *jewel* [ENTER] - the name of the object.
    Response: Object editing menu
25) Press S - To enter a score.
    Response: Enter score:
26) Type in 25 [ENTER] - The score which will be given for carrying the object.
    Response: Object editing menu

27) Press C - To edit the verb list
Response: Add verb? Y/N

28) Press Y - You want to add a new verb.
Response: Enter command verb:

29) Type in *take* [ENTER] -command is to be "TAKE JEWEL"

Edit Name, Condition, Actions
Exit

Verb Editing Menu

30) Press A - To change the actions which are executed by the "TAKE JEWEL" command.
Response: Add command? Y/N

31) Press Y - Add an action.

Say, Die, Move, sTatus, Jump

Command Menu

32) Press S - SAY command.
Response: Enter text to SAY:

33) Type in: *You pick up the jewel.* [ENTER] - This text will be printed when the "TAKE JEWEL" command is used.
Response: Add command? Y/N

34) Press Y - another action.
Response:

Say, Die, Move, sTatus, Jump

Command Menu

35) Press M - MOVE command.
Response: New position? Y/N

36) Press Y - To give a position to move the jewel to.
Response:

Position? Any, Carried, Here
Select.

Position Menu

37) Press C - To select Carried.
Response: New object? Y/N

38) Press Y - To select an object to move
Response:

1	.player
2	jewel

39) Press 2 - To select the jewel as the object to be moved.
Response: Add command? Y/N

40) Press N - You've entered all the actions for the TAKE verb
Response:

Edit NName, CCondition, AActions
EExit

Verb Editing Menu

41) Press E - To Exit, you've created a verb "TAKE".
Response: Add verb Y/N

42) Press Y - To add another verb
Response: Enter command verb:

43) Type in *drop* ENTER - to create command "DROP JEWEL"
Response: Verb editing menu.

44) Press A - To enter actions.
Response: Add command? Y/N

45) Press Y
Response: Command menu

46) Press M - Move command.
Response: New position? Y/N

47) Press Y
Response:

Position? AAny, CCarried, HHere
SSelect.

Position Menu

48) Press S - select position by coordinates.
Response: Enter Y position:

50) Type in 0 ENTER - Room 3,0 = The floor.
Response: Enter X position:

50) Type in: 3 ENTER
Response: New object? Y/N

51) Press Y - To select the object to be dropped.
Response:

1	.player
2	jewel

52) Press 2 - to select the jewel.
Response: Add command? Y/N

53) Press N - No more actions.
Response: Verb editing menu

54) Press E - To exit.
Response: Add verb? Y/N

55) Press N - No more verbs needed.
Response: Object editing menu.

56) Press E - To exit.
Response: Centre editing menu

57) Press E - to exit
Response: Outer menu

58) Press P to play the new game, locate the jewel and use the commands "TAKE JEWEL" and "DROP JEWEL" on it.

THIS DIAGRAM SHOWS THE INFORMATION STORED FOR THE OBJECT' JEWEL' CREATED IN EXAMPLE 2

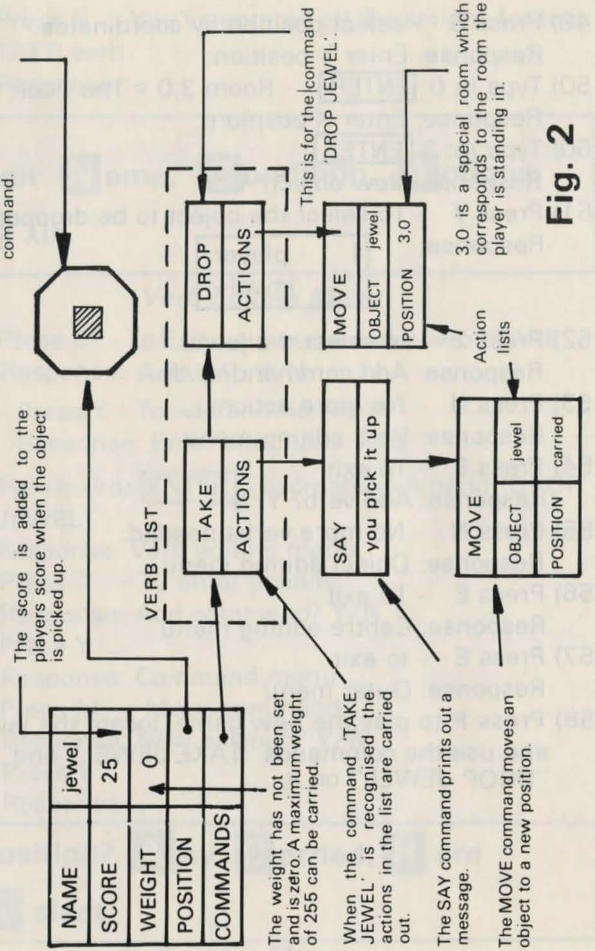

NAME
jewel
SCORE
25
WEIGHT
0
POSITION
COMMANDS
The score is added to the players score when the object is picked up.
This is the starting position and is changed by a MOVE command.
VERB LIST
TAKE
ACTIONS
DROP
ACTIONS
SAY
you pick it up
MOVE
OBJECT
jewel
POSITION
carried
MOVE
OBJECT
jewel
POSITION
3,0
This is for the command 'DROP JEWEL'.
3,0 is a special room which corresponds to the room the player is standing in.
Action lists
The weight has not been set and is zero. A maximum weight of 255 can be carried.
When the command 'TAKE JEWEL' is recognised the actions in the list are carried out.
The SAY command prints out a message.
The MOVE command moves an object to a new position.

Fig. 2

As in chapter 2 there now follows a general description of the use of objects. If you are going to work through it, SAVE your examples first.

A treasure is an object and can be created using the "cReate object" command. Up to 256 objects can be created. This command is activated by pressing R from an editing menu. Move to the room where you want the object to be placed at the start of the game and do this. (Step 22,23)

The program will reply:

Enter name:

Type in the name you wish to give the object. Only the first 8 letters of the word are stored and used for matching as most words differ in their first 8 letters. The editing keys can be used as for a room description.

Press ENTER when you are finished. (Step 24)

This takes you to the **object editing menu**

The object has automatically been placed in the room pointed to by the cursor. An object can be given a score which is added to the players score when the object is picked up. To do this type S for score. (Step 25) The program will reply:

Enter score:

Type in the score, which should be less than 255 and may be edited with the←,→and DELETE keys. Press ENTER when finished (Step 26)

Each object has a list of command verbs (action words) which the computer understands. This is called the verb list. (See fig 2) We now have to define

a command to pick up the object. Press C to edit the commands.
The prompt:

Add verb? (Y/N)

will appear. Reply Y to this. The program will then say:

Enter verb name? (Y/N)

Enter the verb name i.e. *"take"* in the same way as you entered an object name. (Step 29) The system will reply:

Edit N ame, C ondition, A ctions E xit

Verb Editing Menu

This is the **verb editing menu.** As shown in the diagram each verb has attached to it a list of actions which are executed when the command (i.e. "TAKE JEWEL") is recognised. This is called the Action list.
Press A to edit it (Step 30)
The message

Add command (Y/N)?

will appear. Press Y and then the **command menu**:

S ay, D ie, M ove, s T atus, J ump

appears.

To pick up the object we want to move it to the position "Carried" so we use the Move command.
Press M. (Step 35) The message

New position? (Y/N)

will appear. Press Y as you need to set the position to move the object to.
This causes the **Position menu**

Position? A ny, C arried, H ere S elect.

Position Menu

to appear.
Press C to select carried. (Step 37) The message

New object? Y/N

will be displayed. Press Y and a menu of objects will appear including the one you are defining. Press the number beside it to select it. (Step 39) The "Add command?" message will reappear. Reply Y, then press S to select the SAY command. (Step 32) When executed this command prints a message to the player. The program will prompt:

Enter text to SAY

Type in a message such as *Ok*, or *You pick it up* in the same way you typed in a description. (Step 33) Reply N to the "Add command?"message and exit from the verb editing menu. Reply N to "Add verb?", exit from the object editing menu and play the new game. You should find your created object and be able to pick it up with the command:

take "object" i.e. TAKE JEWEL

If not, check particularly the object position and the commands. You will not, however, be able to put it down as you have not defined a suitable command. To do this return to an editing menu and press M to modify an object. Select your object from the displayed menu as before by pressing the number next to it. The **Object editing menu** will redisplay. Press C to select Commands. The message

Edit command take Y/N?

will appear. Reply N. (Replying Y allows the "take" verb you just defined to be altered.) The "Add verb?" message will appear. Reply Y and define the verb "drop" in the same way as you defined "take" except for the position parameter of MOVE. (Steps 43-47) When the position menu appears press S for Select in place of C for Carried. (Step 48) This allows a room to be selected by it's coordinates. Room 3,0 is a special room which corresponds to the room containing the player. Thus, when you move an object to 3,0 it is "dropped" onto the floor of the room. The program will ask:

Enter Y position:

reply 0 to this and enter 3 as the x coordinate. Finish as before. (Steps 51-58) When you play the new game you should be able to "take" and "drop" objects.

If the object has a score assigned it will be added on

when you pick the object up and taken away again when you drop it.

An object can be given a weight in tenths of a kilogram. You can carry up to 50kg and an object may weigh up to 25.5kg. This can be used to restrict the amount of stuff the player can carry. The W command allows entry of a weight in the same manner as for a score.

If you want to create a book, i.e. of instructions you can create a "read" verb which uses a SAY command to print the text, instead of "OK" as in the "take" command. This is done in example 3 below which follows on from example 2.

Example 3

(This example starts from the outer menu)

59) Press M - To modify the stored game.
Response: Edge editing menu

60) Move to the centre of the cell (2,2)
Response:

Edit S core, D escription, P icture

M odify, c R eate object E xit

61) Press R - to cReate an object
Response: Enter object name

62) Type in *notice* ENTER
Response: Object editing menu

63) Press C - To enter a verb
Response: Add verb? Y/N

64) Press Y
Response: Enter command verb

65) Type in: *read* ENTER - "READ NOTICE"
Response: Verb editing menu

66) Press A - To change actions.
Response: Add command? Y/N

67) Press Y
Response: Command menu

68) Press S - SAY command.
Response: Enter text to SAY:

69) Type in: *Abandon hope all ye who enter here.*
ENTER - The message to be printed when the command "READ NOTICE" is recognised.
Response: Add command Y/N?

70) Press N
Response: Verb editing menu

71) Press E - to exit
Response: Add verb? Y/N

72) Press N
Response: Object editing menu

73) Press E - to exit
Response: Centre editing menu

74) Press E - to exit to outer menu.
Response: Outer menu

75) Press P to play the game; reading the notice should now be possible.

Summary:

1) Up to 256 objects can be created, with the cReate object command, to represent things such as treasures.

2) Each object can have a Score, Weight, Position and a number of verbs acting upon it.

3) A verb causes a list of actions to be executed on it's recognition.

4) The SAY action prints a message.

5) The MOVE action moves an object.

6) Room 3,0 is a special room corresponding to the position of the player.

5) Doors and cliffs

It is easy to set up a hill or cliff which can be climbed up but not down. This is done by opening the wall of the room from which movement is to be allowed but not the wall through which it is not. (fig 1) Suitable descriptions should also be entered in each room i.e. "To your south is a cliff which you can climb down but not back up".

This is demonstrated below in example 4

Example 4:

This example creates a steep slope which you can climb down but not up.

76) Press M - To modify game.
Response: Centre editing menu

77) Move the cursor to the centre of cell 2,2 - The starting room of the game.
Response: Centre editing menu

78) Press D - To alter the description
Response: Enter description:
You are in an entrance hall, a passage leads south. ■

79) Add to this text the words:
A steep slope leads east, which you can climb down but not up.
Response: Centre editing menu

80) Press → once - To move to the east wall of cell
Response: Edge editing menu (2,2.)

81) Press letter O - To open a wall for east-west passage.
Response: The cursor turns bright green.

82) Press → twice - to move to the centre of cell 3,2.
(Note that the west wall of cell 3,2 is **not** opened. West-east passage is not allowed through this wall.)
Response: Centre editing menu.

83) Press D
Response: Enter description:

84) Type in the text: *You are at the bottom of a steep cliff, a path leads southwest.*
Response: Centre editing menu

85) Press ↓ once and ← once.
Response: Cursor on SW edge of cell 3,2.

86) Press letter O - To open the SW wall of cell 3,2
Response: Cursor turns bright green.

87) Press ↓ once and ← once
Response: Cursor on NW edge of cell (2,3)

88) Press letter O - To open a 2-way passage from cell 3,2 to cell 2,3
Response: Cursor turns bright green.

89) Press E - To exit
Play the new game. Notice that travel is only possible one way up the slope.

Creating a door is more difficult. First the door must be created as an object, and then movement made conditional on the door being open.

The following example creates a cupboard with a door in it.

Example 5

This continues from example 4

89) Press M
Response: edge editing menu

90) Move the cursor to the centre of cell 2,3
Response: centre editing menu

91) Press R - To create an object, the door.
Response: Enter object name:

92) Type in *door* ENTER
Response: Edge editing menu

93) Press D - To edit the description list.
Response: Add description? Y/N

94) Press Y - You want to add a description
Response: New status? Y/N

95) Press Y - To select a status for the description
Response: Enter new status: (0 =Any)

96) Type in 1 ENTER
Response: Edit description Y/N

97) Press Y - to enter the description text
Response: Enter description:

98) Type in *A closed door is to the south.* ENTER - will be printed for status 1
Response: Add description Y/N

99) Press Y - For the status 2 description
Response: New status? Y/N

100) Press Y
Response: Enter new status: (0 =Any)

101) Type in 2 ENTER - Descriptor for status 2
Response: Edit description? Y/N

102) Press Y
Response: Enter description:

103) Type in *To your south a door is open* - will be printed for status 2.
Response: Add description Y/N

104) Press N
Response: Object editing menu

105) Press C - To alter verb list.
Response: Add verb? Y/N

106) Press Y
Response: Enter command verb:

107) Type in *open* ENTER
Response: Verb editing menu.

108) Press A - to enter action list.
Response: Add command? Y/N

109) Press Y
Response: Command menu

110) Press S - to select SAY
Response: Enter text to SAY:

111) Type in *The door swings open.* ENTER
Response: Add command? Y/N

112) Press Y
Response: Command menu

113) Press T - To select sTatus command
Response: New object? Y/N

114) Press Y - to enter an object to have it's status changed.
Response:

1	.player
2	jewel
3	notice
4	door

115) Press 4 - to select the door as the object to change the status of.
Response: New status? Y/N

116) Press Y - To select a new status to assign.
Response: Enter new status: (0 = Any)

117) Type in 2 ENTER - This command changes the status of object 'door' to 2.
Response: Add command? Y/N

118) Press N
Response: Verb editing menu

119) Press E
Response: Add verb? Y/N

120) Press N
Response: Object editing menu

121) Press E
Response: Centre editing menu

122) Press ↓ - to move to south edge of room 2,3.
Response: Edge editing menu

123) Press C - to open the wall conditionally.
Response: Cursor turns flashing bright red
New object? Y/N

124) Press Y - to select an object to test for status.
Response:

1.	.player
2.	jewel
3.	notice
4.	door

125) Press 4 - to select door as the object
Response: New status? Y/N

126) Press Y - to select a new status to test.
Response: Enter new status:
(0 = Any)

127) Type in: 2 ENTER - Door must have status 2 to allow movement.
Response: New position? Y/N

128) Press Y
Response: Position menu

129) Press A - to select Any position as ok for the door.
Response: Edge editing menu

130) Press ↓ - To move to the north edge of room 2,4.
Response: Edge editing menu

131) Press letter O - to open a gap for reverse travel.
Response: Cursor turns bright green.

132) Press ↓ - to move to centre of room.
Response: Centre editing menu

133) Press D - to describe room
Response: Enter description:

134) Type in: *You are in a cupboard. A door hangs open to the north.* ENTER
Response: Centre editing menu

135) Press E - to exit, then press P to play the new game. Notice that you have to open the door to get into the cupboard.

There now follows a more detailed discussion of doors.

THIS DIAGRAM SHOWS THE OBJECT 'DOOR' IN EXAMPLE 5

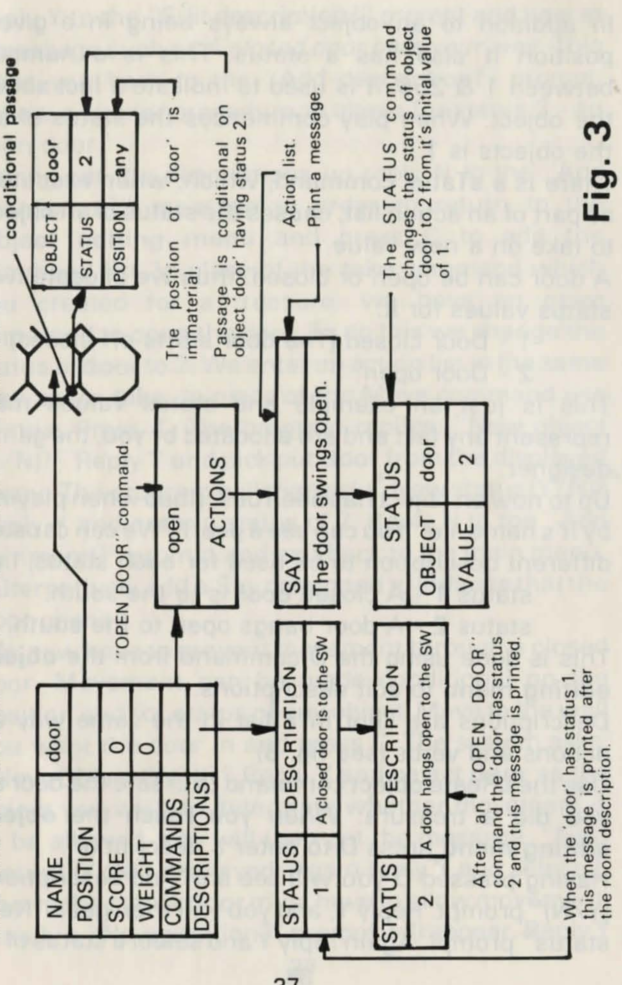

NAME
door
POSITION
SCORE
0
WEIGHT
0
COMMANDS
DESCRIPTIONS
"OPEN DOOR" command.
open
ACTIONS
conditional passage
OBJECT
door
STATUS
2
POSITION
any
The position of 'door' is immaterial.
Passage is conditional on object 'door' having status 2.
STATUS
DESCRIPTION
1
A closed door is to the SW
SAY
The door swings open.
Action list.
Print a message.
STATUS
DESCRIPTION
2
A door hangs open to the SW
STATUS
door
OBJECT
VALUE
2
The STATUS command changes the status of object 'door' to 2 from it's initial value of 1.
After an "OPEN DOOR" command the 'door' has status 2 and this message is printed.
When the 'door' has status 1, this message is printed after the room description.

Fig. 3

In addition to an object always being in a given position it also has a status. This is a number between 1 & 256. It is used to indicate a fact about the object. When play commences the status of all the objects is 1.

There is a sTatus command, which, when executed as part of an action list, causes the status of an object to take on a new value.

A door can be open or closed, thus we allocate two status values for it:

1 - Door closed (The door starts off closed)
2 - Door open.

This is just an example and status values may represent any fact and are allocated by you, the game designer.

Up to now an object has been described when playing by it's name, i.e. 'You can see a pearl'. We can cause a different description to be used for each status, i.e.

status 1 - A closed door is to the south.
status 2 - A door hangs open to the south.

This is done using the D command from the **object editing menu** to edit descriptions.

Descriptions are kept in a list in the same way as actions and verbs (see fig. 3)

Use the cReate object command to create the door as you did a treasure. When you reach the **object editing menu** press D to enter a description.

Having pressed D you will see an "Add description? (Y/N)" prompt. Reply Y, and you will then get a "New status" prompt. Again reply Y and select a status of 1.

Reply Y to the "Edit description?" prompt and type in a message such as *A closed door bars your way.* This takes you back to the "Add description?" prompt. Follow a similar procedure to above for status 2 - an open door.

Having set the descriptions up reply N to the "Add description?" message in order to return to the **Object editing menu** and press C to edit the command list. In place of the **take** command which you created for a treasure, we have an **open** command to open the door. To do this we change the status of **door** to 2. We enter an action list in the same way as for **take.** In place of the Move command use sTatus. Press T. The program replies: "New object (Y/N)?" Reply Y and pick out **door** from the displayed menu. The program will then ask:" New status (Y/N)?" Reply Y and enter a status of 2. Reply N to the "Add command?" prompt and exit back to the main menu. (Alternatively add a Say command to indicate that the door opens.)

We now have to prevent movement through a closed door. Movement can be made conditional on the position and/or status of any object. Move to the wall you want the door in and press C. The system asks: "New Object (Y/N)?" Reply Y and select **door** as the object you wish to determine whether movement is to be allowed. You will then get the message: "New status? (Y/N)" displayed. Again reply Y and select 2. This is the status **door** must have to allow movement. Finally a "New position?" prompt will appear. Reply Y and select Any as the position of the door is immaterial.

Play your improved game and notice that you cannot move through the door without opening it first. Try creating a few more doors.

Summary:

1) It is easy to make a one way passage by only opening one half of a wall.
2) Every object has an arbitrary status value
3) The status starts off at 1 and is altered by the sTatus command in an Action list
4) A door is created as an object like any other
5) Passage through a wall can be made conditional on the status and/or position of any one object.

6) Locking doors

Doors that open when pushed offer little challenge to the adventurer, so it is necessary to lock them and make the player find a key, magic spell or means of battering down the door.

We have set up conditions for moving through a wall. In the same way it is possible to have a condition which must be fulfilled for a verb to be executed. For a locked door the condition would be that the object **key** must be in position **carried.**

A key is generated in exactly the same way as a treasure, although as it is useful in it's own right it doesn't necessarily need a score.

In the following example the cupboard door created in example 5 is made locked and cannot be opened without a key.

Example 6:

136) Press M - To modify the stored game.
Response: Screen with cursor in NW of cell (1,1)
Edge editing menu

137) Use the cursor keys to move to centre of cell X=2, Y=2
Response: Screen with cursor in centre of cell (2,2)
Centre editing menu

138) Press R - To cReate an object
Response: Enter object name:

139) Type in *key* ENTER - the name of the object.
Response: Object editing menu

140) Press C
Response: Add verb? Y/N

141) Press Y - You want to add a new verb
Response: Enter command verb:

142) Type in *take* ENTER - command is to be "TAKE KEY"
Response: Verb editing menu

143) Press A - To change the actions which are executed by the "TAKE KEY" command
Response: Add command ? Y/N

144) Press Y - Add an action
Response: Command menu

145) Press S - SAY command
Response: Enter text to SAY:

146) Type in: *You pick up the key.* ENTER - This text will be printed when the "TAKE KEY" command is used.
Response: Add command? Y/N

147) Press Y - another action
Response: Command menu

148) Press M - MOVE command
Response: New position? Y/N

149) Press Y - To give a position to move the key to
Response: Position menu

150) Press C - To select Carried
Response: New object? Y/N

151) Press Y - To select an object to move.
Response:

```
1   player
2   jewel
3   notice
4   door
5   key
```

152) Press 5 - To select the key as the object to be moved
Response: Add command? Y/N

153) Press N - You've entered all the actions for the TAKE verb
Response: Verb editing menu

154) Press E - To Exit, you've created a verb "TAKE".
Response: Add verb? Y/N

155) Press Y - To add another verb
Response: Enter command verb:

156) Type in *drop* ENTER - to create command "DROP KEY"
Response: Verb editing menu

157) Press A - To enter actions
Response: Add command? Y/N.

158) Press Y
Response: Action menu

159) Press M - Move command
Response: New position? Y/N

160) Press Y
Response: Position menu

161) Press S - select position by coordinates
Response: Enter Y position:

162) Type in:0 ENTER - Room 3,0 – The floor
Response: Enter X position:

163) Type in: 3 ENTER
Response: New object? Y/N

164) Press Y - To select the object to be dropped
Response:

1	player
2	jewel
3	notice
4	door
5	key

165) Press 5 - to select the key
Response: Add command? Y/N

166) Press N - No more actions
Response: Verb editing menu

167) Press E - To exit
Response: Add verb? Y/N

168) Press N - No more verbs needed.
Response: Object editing menu

169) Press E - To exit
Response: Centre editing menu

170) Press M - to modify an object.
Response:

1	player
2	jewel
3	notice
4	door
5	key

171) Press 4 - To select the door to be modified
Response: Object editing menu

172) Press C - To modify the command list
Response: Edit verb open? Y/N

173) Press Y - To alter the "OPEN DOOR" command to require possession of a key.
Response: Verb editing menu

174) Press C - To set a condition for verb execution
Response: New object? Y/N

175) Press Y - To select an object
Response:

1	.player
2	jewel
3	notice
4	door
5	key

176) Press 5 - the key as the object to test
Response: New status? Y/N

177) Press N - The status of 'key' is immaterial
Respone: New position? Y/N

178) Press Y - To set a position
Response: Position menu

179) Press C - The key must be carried to open the door
Response: Verb editing menu

180) Press E - To exit
Response: Add verb? Y/N

181) Press N
Response: Object editing menu

182) Press E - to exit
Response: Centre editing menu

183) Press E - to exit and notice that a key is now needed to open a door.

The general method to lock a door is as follows:
Modify the door you wish to be locked and go to the **verb editing menu** for the **open** command. Press C to select a condition. Select object **key.** The status is 0, matching any value and need not be changed. The position should be set to Carried

Summary:

1) Doors may be locked so that a key is needed to open them

2) A key is created like any other object

3) A condition can be put on a verb, in the same way as on a wall, that must be fulfilled for it to be executed.

7) What to do when you make mistakes

It is inevitable that you will make a few errors when designing a game.

As already mentioned, if you open a wall by accident pressing B will Block it off

If you want to change a score or description etc, proceed in the same way as for a new entry. The program will display the previous text for you to modify.

To modify an object press M and select that object from the menu. This puts you in the same position as you were when creating that object. You can correct a misspelt name with the N command from the **object editing menu.**

Pressing C from the **object editing menu** allows you to edit the objects verb list.

It will display:

Edit verb take Y/N

Pressing Y will take you to the **verb editing menu** which allows you to alter the verb, whilst pressing N moves you along the list.

To edit an action list press A from the **verb editing menu,** or move to a yellow wall of a room and press Q.

If the action list has any commands in it a message such as:

```
Command:    Say,

D elete, E dit, N ext
```

will appear.

Press E to alter the command
Press D to remove the command from the list
Press N to move on to the next command
If you wish to destroy a verb or object the best way is to change its name to a rubbish code such as WXY$//%. To destroy an object, move it to an unused room.

8) Scary monsters

It is possible to create aggressive lifeforms to people your game. These are objects which, when present and alive, prevent movement past or away from a point. The police roadblock prevented movement away until they were neutralised, while it was not possible to get past the dragon until it was dead. This is done in the following example which creates a werewolf

Example 7:

184) Press M
Response: Edge editing menu

185) Move the cursor to the centre of cell 2,4.
Response: Centre editing menu

186) Press R - To create an object, the werewolf
Response: Enter object name:

187) Type in *werewolf* ENTER
Response: Edge editing menu

188) Press D - to edit the description list
Response: Add description? Y/N

189) Press Y - You want to add a description
Response: New status? Y/N

190) Press Y - To select a status for the description.
Response: Enter new status: (0=Any)

191) Type in 1 ENTER
Response: Edit description? Y/N

192) Press Y - to enter the description text.
Response: Enter description:

193) Type in: *A huge werewolf is in the room blocking your way* ENTER - will be printed for status 1
Response: Add description? Y/N

194) Reply Y - For the status 2 description.
Response: New status? Y/N

195) Reply Y
Response: Enter new status: (0=Any)

196) Type in 2 ENTER - description for status 2
Response: Edit description? Y/N

197) Press Y
Enter description:

198) Type in *A large dead werewolf lies in a corner* ENTER - will be printed for status 2
Response: Add description? Y/N

199) Press N
Response: Object editing menu

200) Press C - To alter verb list
Response: Add verb? Y/N

201) Reply Y
Response: Enter verb name? Y/N

202) Type in *kill* ENTER
Response: Verb editing menu

203) Press A - to enter action list
Response: Add command? Y/N

204) Press Y
Response: Command menu

205) Press S - to select SAY
Response: Enter text to SAY:

206) Type in *The werewolf collapses slain.* ENTER
Response: Add command? Y/N

207) Press Y
Response: Command menu

208) Press T - To select sTatus command
Response: New object? Y/N

209) Press Y - To enter an object to have it's status changed
Response:

```
1   player
2   jewel
3   notice
4   door
5   key
6   werewolf
```

210) Press 6 - To select the werewolf as the object to change the status of
Response: New status? Y/N

211) Press Y - To select a new status to assign.
Response: Enter new status: (0=Any)

212) Type in 2 ENTER - This command changes the status of object 'werewolf' to 2
Response: Add command? Y/N

213) Press N
Response: Verb editing menu

214) Press E
Response: Add verb? Y/N

215) Press Y
Response: Enter verb name:

216) Type in *eat* ENTER
Response: Verb editing menu

217) Press A - To enter action list
Response: Add command? Y/N

218) Press Y
Response: Action menu

219) Press S - SAY command
Response: Enter text to SAY:

220) Type in: *The werewolf eats you first.* ENTER
Response: Add command? Y/N

221) Press Y
Response: Action menu

222) Press D - DIE command to kill the player off
Response: Add command? Y/N

223) Press N
Response: Verb editing menu

224) Press E
Response: Add verb? Y/N

225) Press N
Response: Object editing menu

226) Press E
Response: Centre editing menu

227) Press ↑ - to move to north edge of room 2,4
Response: Edge editing menu

228) Press C - to open the wall conditionally
Response: Cursor turns flashing bright red
Response: New object? Y/N

229) Press Y - to select an object to test for status
Response:

1	player
2	jewel
3	notice
4	door
5	key
6	werewolf

230) Press 6 - to select werewolf as the object.
Response: New status? Y/N

231) Press Y - to select a new status to test.
Response: Enter new status: (0=Any)

232) Type in: 2 ENTER - Werewolf must have status to allow movement.
Response: New position? Y/N

233) Press Y
Position menu

234) Press A - to select Any position as ok for the werewolf
Response: Edge editing menu

235) Press E - to exit, then press P to play the new game. Try to kill and/or eat the werewolf

Use the cReate command to create your creature as for any other object. A monster that blocks your way until it is dead is very like a door, it will typically have two statuses and descriptions: i.e.

Status 1 - monster alive
Status 2 - monster dead

Enter these using the D command as you did for a door. A method of getting rid of the monster should be available. At it's simplest the command "KILL monster" could alter it's status to 2, thus disposing of it. This command has much the same form as the OPEN command for a door. There may be a condition on this command requiring possession of an object, i.e. the chainsaw.

If the object is chased away it will have only a status 1 descriptor, if any, such as 'there is a police roadblock here' and will be moved to another room by a command. The command "CHASE monster" could move the monster away. Moving an object to a room not otherwise used, i.e. (1,1) can be used to remove it from the game. This method can also be used to introduce an object into the game. The police in the demo game are moved from this room when the "TAKE COINS" command is executed, as this verb has an extra Move command in it's action list.

If an attempt to kill a monster fails, death usually results. Use the Die command to kill off your player. This is activated by pressing D from the action menu. It needs no further information. You should also give an explanation for your players demise with the SAY command. Thus the actions for 'EAT WEREWOLF' would be to SAY 'The werewolf eats you first' and DIE.

To prevent movement put conditional blocks on all the directions through which the player cannot move unless the monster has been neutralised. (Note that if more than one wall of a room has a condition on it, i.e. is shown in red, the same condition will be used for all of them.) Set the object to be the monster. Either set the Position condition to Any and the Status condition to 2, or else set the Position to the room the monster is sent to when chased away, and the Status condition to 0 for Any room.

Summary:

1) A monster that blocks your way is similar to a door

2) Monsters can be chased away by moving them to a room which doesn't form part of the game.

3) If an attempt to kill a monster fails the DIE command kills off the player.

9) Adding pictures

So far our adventure has been purely textual. A picture creator is provided enabling pictures to be drawn. To use this, press P from the **centre editing menu.**

The program will ask for a background colour. The colour is entered as a number. On the Sinclair keyboard a colour is above each number key. Press the number matching the colour you wish to use as a background. The screen will clear and a cross will display along with a status line.

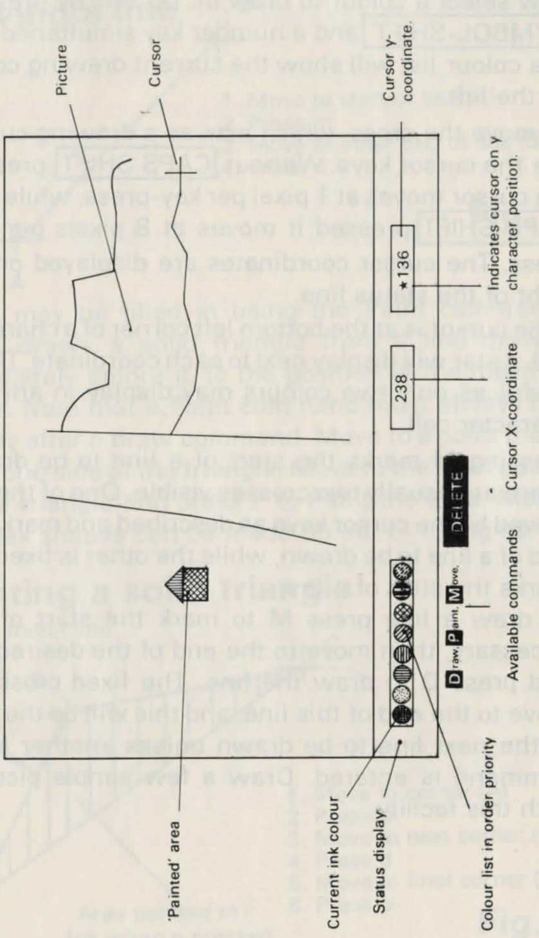


Fig. 4 Picture Creator Display

Now select a colour to draw in. Do this by pressing SYMBOL-SHIFT and a number key simultaneously. The colour list will show the current drawing colour on the left.

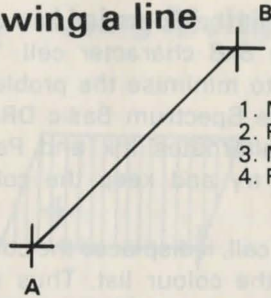
To move the cross, which acts as a drawing cursor, use the cursor keys. Without CAPS SHIFT pressed, the cursor moves at 1 pixel per key-press, while with CAPS-SHIFT pressed it moves at 8 pixels per key-press. The cursor coordinates are displayed on the right of the status line.

If the cursor is at the bottom left corner of a character cell, a star will display next to each coordinate. This is useful as only two colours may display in any one character cell.

Pressing M marks the start of a line to be drawn. There are usually two crosses visible. One of these is moved by the cursor keys as described and marks the end of a line to be drawn, while the other is fixed and marks the start of a line.

To draw a line press M to mark the start of it if necessary, then move to the end of the desired line and press D to draw the line. The fixed cross will move to the end of this line, and this will be the start of the next line to be drawn unless another Move command is entered. Draw a few simple pictures with this facility.

Drawing a line

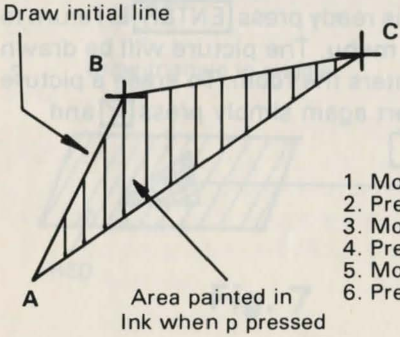


1. Move to start of line (A)
2. Press m
3. Move to other end of line (B)
4. Press d

Fig, 5

Areas may be filled in using the Paint command which draws a solid triangle from a line drawn immediately before it, to the position of the moving cursor. Note that a Paint command must always be directly after a draw command. Move to a point then Draw one side of the triangle. Move to the third point on the triangle and press P to Paint the area. More complex shapes can be made up out of triangles.

Painting a solid triangle



1. Move to corner (A)
2. Press m
3. Move to next corner (B)
4. Press d
5. Move to final corner (C)
6. Press p

Fig. 6

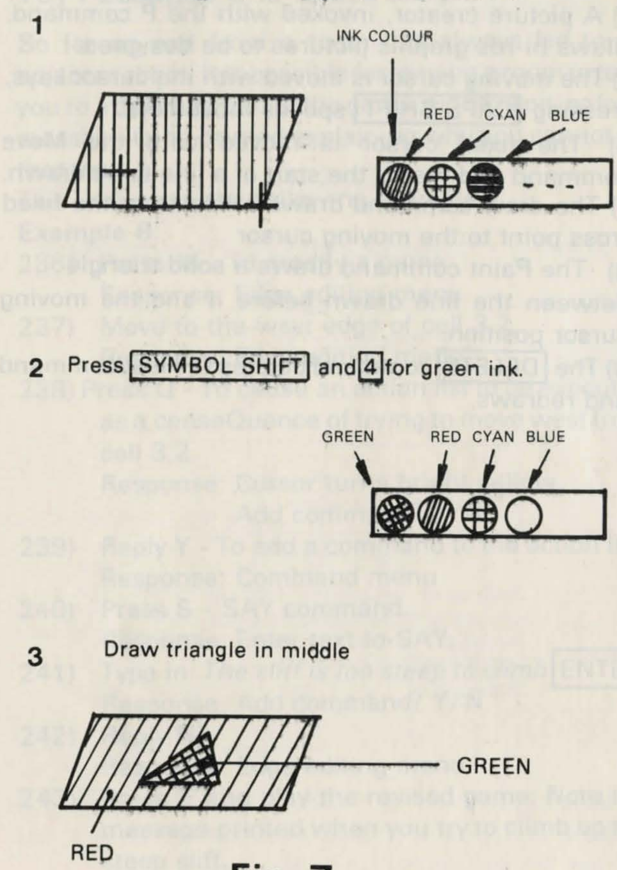
It was indicated earlier that the Spectrum can only display 2 colours in any 8x8 character cell. The picture creator attempts to minimise the problems caused by this. Unlike the Spectrum Basic DRAW command the program alternates Ink and Paper while drawing a line to try and keep the colour constant.

If it has to add colour to a cell, it displaces the colour furthest to the right of the colour list. Thus it is advisable to have colours that you wish to remain displayed on the left of the list. Do this by selecting them as the drawing colour, which moves the list to the right and puts the selected colour on the extreme left.

As an example, if you want to draw in black over a green area; select green ink, then black ink. Black and green will be on the left of the colour list.

If you make a mistake pressing CAPS-SHIFT and 0 deletes the last command and redraws without it. When your picture is ready press ENTER to return to the **centre editing menu**. The picture will be drawn when the player enters the room. To erase a picture so that you can start again simply press X and SYMBOL SHIFT

Using Multiple Colours


1

2 Press SYMBOL SHIFT and 4 for green ink.

3 Draw triangle in middle

Fig. 7

Summary:
1) A picture creator, invoked with the P command, allows hi-res graphic pictures to be designed.
2) The moving cursor is moved with the cursor keys, pressing CAPS SHIFT speeds movement.
3) The fixed cursor is moved with the Move command and marks the start of a line to be drawn.
4) The draw command draws a line from the fixed cross point to the moving cursor
5) The Paint command draws a solid triangle between the line drawn before it and the moving cursor position.
6) The DELETE command deletes the last command and redraws.

10) Passages and spacewarps

So far an exit from a room has always led to an adjacent room. It is possible for leaving a room to take you to a different part of the game, or into a pit, or for a message to be printed explaining why you cannot go that way.
This is done in the following example.

Example 8:

236) Press M - To modify a game.
Response: Edge editing menu

237) Move to the west edge of cell 3,2.
Response: Edge editing menu

238) Press Q - To cause an action list to be executed as a conseQuence of trying to move west from cell 3,2.
Response: Cursor turns bright yellow.
Add command? Y/N

239) Reply Y - To add a command to the action list.
Response: Command menu

240) Press S - SAY command.
Response: Enter text to SAY:

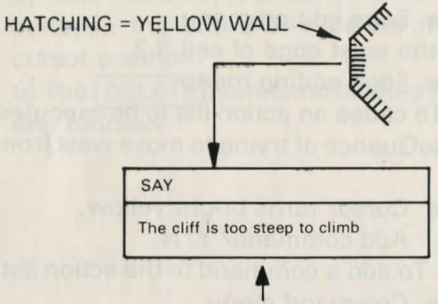
241) Type in: *The cliff is too steep to climb* ENTER.
Response: Add command? Y/N

242) Reply N
Response: Edge editing menu

243) Press E and play the revised game. Note the message printed when you try to climb up the steep cliff.

An action list was set up to be executed by the conseQuential command, which was accessed from the **edge editing menu.**
This causes an action list (as for a verb) to be executed on movement to that edge. At it's simplest this can consist of a SAY command to print 'The cliff is too steep to climb'. (See figure 8)

Using conseQuential movement



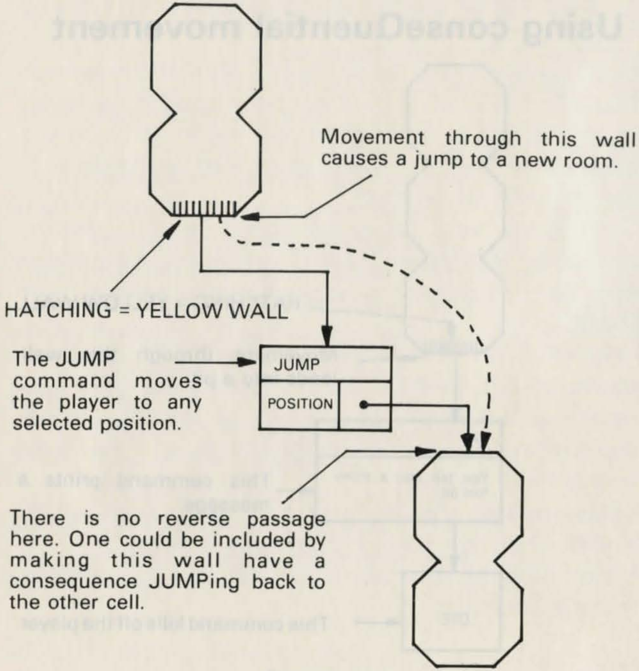
Attempted movement through any of the three walls prints this message , which is more helpful to the player than, 'You can't go that way'.

Fig. 8

Pressing Q here will change the wall colour to yellow and give an *"Add command?"* prompt. An action list can be set up as for a verb.
Normally passage through a yellow wall is prevented.

To cause movement to a different room we can use the Jump command. This will ask for a position to move to. Use Select to enter the coordinates of the room you wish the player to be transported to. As for a condition the same action !ist is used for every wall of a room.

Using conseQuential movement



Movement through this wall causes a jump to a new room.

The JUMP command moves the player to any selected position.

There is no reverse passage here. One could be included by making this wall have a consequence JUMPing back to the other cell.

Fig. 9

If a two way passage is to be implemented you will need to set up conseQuential walls with Jump commands to the other end at both ends of the passage.

The maze of twisty little passages in the demonstration game was implemented this way with some routes causing a Jump whilst others cause you to fall into a pit, (executing a Die command.)

Using conseQuential movement

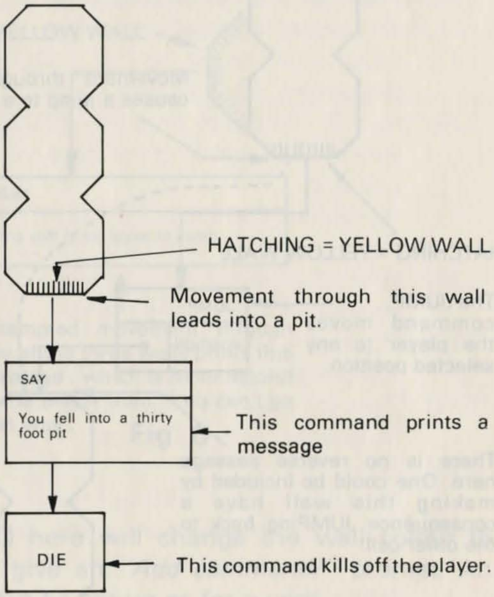

HATCHING = YELLOW WALL
Movement through this wall leads into a pit.
SAY
You fell into a thirty foot pit
This command prints a message
DIE
This command kills off the player.

Fig. 10

Summary:

1) The conseQuential command from an edge editing menu causes an action list to be executed on attempted movement through a wall.

2) The JUMP action moves the player to a new room.

3) Movement is prevented through a yellow wall unless otherwise specified.

11) Rising and falling

We have up to now had totally 'flat' adventures, where Up and Down movement is not possible. You can allow such movement with the aid of pseudo-objects. These start with a dot, i.e. ".player", ".stair1". The effects of the dot are:

1) In the absence of a specific description the name will not be printed in an inventory or on entry to a room.

Thus if a room contains an object '.stair' the message "You can see a .stair" will **not** be printed. If however the object has a description defined for it's current status value such as "Stone steps lead up the dome." then this will be printed.

2) Any verb using the object needs no noun, i.e. saying "UP" When ".stair" is present in the same room as the player is equivalent to saying "UP .STAIR".

If you want to make an object invisible do not use a pseudo-object. Instead set the descriptions for any status values where the object is invisible to an empty string, i.e. press ENTER in reply to the "Edit description:" prompt.

Pseudo objects are very useful for verbs such as UP, DOWN which do not normally take a noun.

In the following example a pseudo-object is used to create a staircase.

Example 9:

244) Press M - To modify the stored game.
Response: Edge editing menu

245) Move to the centre of room 2,2
Response: Centre editing menu

246) Press D - To alter a description.
Response: Enter description:
You are in an entrance hall, a passage leads south. A steep slope leads east, which you can climb down but not up. ■

247) Append to this text:
A spiral staircase leads up. ENTER
Response: Centre editing menu.

248) Press R - to create an object
Response: Enter object name:

249) Type in the name: *.stair1* ENTER
Response: Object editing menu

250) Press C - To edit commands.
Response: Add verb? Y/N

251) Press Y
Response: Enter command verb:

252) Type in: *up* ENTER
Response: Verb editing menu

253) Press A - To edit actions
Response: Add command? Y/N

254) Press Y
Response: Command menu

255) Press J - JUMP action
Response: Position menu

256) Press S - To Select a room to jump to.
Response: Enter Y position:

257) Type in 10 ENTER - To jump to room 2,10.
Response: Enter X position:

258) Type in 2 ENTER
Response: Add command? Y/N

259) Press N
Response: Verb editing menu

260) Press E
Response: Add verb? Y/N

261) Press N
Response: Object editing menu.

262) Press E
Response: Centre editing menu

263) Move to centre of room 2,10
Response: Centre editing menu

264) Press D
Response: Enter description:

265) Type in: *You are at the top of a spiral staircase.* ENTER
Response: Centre editing menu

266) Press R - to create an object.
Response: Enter object name:

267) Type in the name: *.stair2* ENTER
Response: Object editing menu

268) Press C - To edit commands
Response: Add verb? Y/N

269) Press Y
Response: Enter command verb:

270) Type in: *down* ENTER
Response: Verb editing menu

271) Press A - To edit actions.
Response: Add command? Y/N

272) Press Y
Response: Command menu

273) Press J - JUMP action.
Response: Position menu

274) Press S - To Select a room to jump to.
Response: Enter Y position:

275) Type in 2 ENTER - To jump back to room 2,2
Response: Enter X position:

276) Type in: 2 ENTER
Response: Add command? Y/N

277) Press N
Response: Verb editing menu

278) Press E
Response: Add verb? Y/N

279) Press N
Response: Object editing menu

280) Press E
Response: Centre editing menu

281) Press E and play the new game. Notice that you can climb the staircase to the room above.

To implement a staircase create objects such as ".stair1" and ".stair2" at the top and bottom of it. ".stair1" at the top will have the verb "UP", which will cause a jump to the room at the top, while ".stair2" will have verbs "D" and "DOWN" which both cause a jump back to the room below. It is

possible to put conditions on these as on any other verbs, thus allowing for locked trapdoors and the like. Alternatively, by moving the dotted object in from a hidden room, a staircase can be made to appear only at a certain point in a game.
Pseudo-objects can be used for a variety of other purposes, for example a magic word such as "XYZZY" can be made to work on a pseudo-object to cause a teleport to another region of the game.

Summary:
1) A pseudo-object has a name starting with a dot.
2) Pseudo-objects names are not printed by an invent or look command.
3) The name of a pseudo-object is not given in a verb which uses it.

12) Multiple actions

When you play the game a section of the program called the command processor tries to make sense of your input. It runs down the list of all the objects you have defined, looking to see if the object is the same as the last word in the command, i.e. "TAKE KNIFE" will cause it to look for an occurence of the object "KNIFE".
When it finds a match it then scans along the verb list of the object looking for a match for the first word in the command. i.e. "TAKE". Having succeeded in matching the command it will check that the object referred to is present in the same room as the player. If not it will continue trying.
If the object is present it will then check the verbs condition, if any. Remember that this is that the status and position of a given object match those specified in the condition. If this is not fulfilled it will continue looking. Only when all possibilities have been tried will the "I don't understand" message be printed.
The upshot of this is that it is possible to have objects in a game which have defined a number of verbs with the same name but different conditions. At it's simplest this allows a more friendly message than "I don't understand" to be produced using the Say command.
This is done in the following example which improves the monster fight routines.

Example 10:

282) Press M - To modify the game.
Response: Edge editing menu

283) Move to the centre of cell 2,2.
Response: Centre editing menu

284) Press M - to modify an object.
Response:

```
1    .player
2    jewel
3    notice
4    door
5    key
6    werewolf
7    .stair1
8    .stair2
```

285) Press 6 - To select the werewolf.
Response: Object editing menu

286) Press C - To modify the verbs.
Response: edit verb kill? Y/N

287) Press Y - To alter the verb "KILL WEREWOLF"
Response: Verb editing menu

288) Press C - To give a condition to the verb
Response: New object? Y/N

289) Press Y
Response:

```
1    .player
2    jewel
3    notice
4    door
5    key
6    werewolf
7    .stair1
8    .stair2
```

290) Press 2 - to select the jewel
Response: new status? Y/N

291) Press N
Response: new position? Y/N

292) Press Y
Response: Position menu

293) Press C - The jewel must be carried.
Response: Verb editing menu

294) Press E
Response: edit verb eat? Y/N

295) Press N
Response: Add verb? Y/N

296) Press Y
Response: Enter command verb:

297) Type in: *kill* ENTER - For an attempt to kill the werewolf without the jewel being carried.
Response: Verb editing menu

298) Press A - To enter actions.
Response: Add command? Y/N

299) Press Y
Response: Command menu
300) Press S - SAY command.
Response: Enter text to SAY:
301) Type in: *The monster is too strong for you.* ENTER
Response: Add command? Y/N
302) Press N
Response: Verb editing menu
303) Press E
Response: Add verb? Y/N
304) Press N
Response: Object editing menu
305) Press E
Response: Centre editing menu
306) Press E and play the new game. Notice that the "KILL MONSTER" command now requires that you possess the jewel to succeed.

This is the way the "SPIN KNIFE" command works to help you find your way through the maze. The object KNIFE has several SPIN verbs defined, each one having as it's condition KNIFE being in a particular room in the maze. The verbs Say a message indicating to the adventurer which way to go.

A monster can have 2 or more KILL verbs, with some conditional on possession of a weapon and succeeding, while the last results in the players termination.

Multiple actions and pseudo-objects together allow adventures to be made more exciting and realistic. It

may be helpful to plan what you want to happen and how to achieve it before trying to key in that part of your game.

Summary:

1) An object can have more than one verb with the same name.

2) The first verb with a valid condition in any verb list will be the one that is executed.

13) Generating stand alone programs

A facility is provided to allow the generation of machine code games which can be loaded and run without 'The Dungeon Builder' being present: thus you can give your friends games to play, even if they do not have 'the Dungeon Builder'.

First Save your game in the usual way as described in Chapter 3, then disconnect your Spectrum briefly and load the 'Make' utility which is stored on Side 2 of the program tape.

To do this, type:

LOAD "" ENTER this time with the tape at the start of side 2. The program will load and prompt you with:

Enter saved game name:

Type in the name you saved your game under and press ENTER. Put the tape with the saved game in the recorder, rewind and start the tape. The program will then prompt

Enter your own name:

Type it in and press ENTER

Finally it will prompt

Enter game name:

Type in the name you want to call your game and press ENTER. The program will then ask you to start the tape. Put in the cassette you want to store the game on, rewind it and press PLAY and RECORD on the tape recorder.

When saving is complete it will ask

Another copy Y/N?.

Replying Y allows further copies to be made.

The generated game can be loaded with LOAD "" ENTER and will run automatically.

The Dungeon Builder

GRAPHIC ADVENTURE SYSTEM

REFERENCE MANUAL

Loading From Tape:

To load from tape type:

LOAD " " ENTER

The program will autorun

Outer menu.

This displays the "The Dungeon Builder" banner and the options:

Load, Save, Verify, Modify,
Create, Play game

Load:

This command loads a game file from tape. It will prompt for a filename.

Save:

This command allows a game file to be written to tape. It will prompt for a filename.

Verify:

This command verifys a game file was saved correctly. It prompts for a filename.

To abort any of the above press BREAK and press ENTER to return to the outer menu.

Create:

This command prompts for confirmation, then clears game memory and goes to the **Edit screen** to allow a new game to be created.

Modify:

This command goes to the **Edit screen** to allow the game in memory to be modified.

Play:

This command causes the game to commence.

Edit Screen

This displays a window onto the grid of hexagonal cells which represent the rooms in the scenario. A display of bytes free is given at the top of the screen.

Cursor keys.

These move the flashing cursor around the screen. The screen will scroll allowing the cursor to be moved anywhere on a 40x40 grid. If the cursor is moved to the edge of the grid the computer will bleep.

Menu displays.

The menu displayed varies depending on whether the cursor is at the edge of a cell or in the centre of it.

Centre Menu

This is displayed when the cursor is at the centre of a cell and has the options:

Edit **S**core, **D**escription, **P**icture

Modify, c**R**eate object **E**xit

Centre Editing Menu

Score:

This allows a score to be assigned for discovering the selected room (that with the cursor in it.). The score can be edited using ←, → and DELETE.

Description:

This allows a room to be described. The description should be typed in between the quotes. Editing keys may be used as specified above.

Picture:

This calls the **Picture Creator** to define a hi-res picture of the selected room.

Exit:

This exits to the **Outer Menu**.

Edge menu.

This is displayed when the cursor is on the edge of a cell and has the following options, mainly concerned with changing the walls of the cell:

Open, **B**lock, **C**ond, conse**Q**uential,

Modify, c**R**eate object, **E**xit

Open:

This command opens a gap in the wall of the the room at the position pointed to by the cursor. For a two way corridor the walls of both rooms should be opened.

Block:

This is the inverse of the above command, i.e. it blocks up an open gap.

Cond:

This conditionally opens up a gap, i.e. it builds in a door which can be locked. The command will cause entry to the **Select Condition Sequence.** The conditionally open gap will show up in red.

conseQuential:

This will result in attempted movement through the marked wall causing a series of commands to be obeyed. The command causes entry to the **Select Command List Sequence.** The wall will show up in yellow.

Exit:

This will cause an exit to the **Outer menu.**

Creating and Modifying objects.

Objects can be created and modified from an edit menu using the *cReate* and *Modify* commands.

cReate:

The cReate command creates a new object. It will prompt for the name of the object. Names longer than 8 characters will be truncated. If a name starts with "." it will not be displayed on any inventory or room description and any commands acting on it will be accepted without a noun being specified. One object ".player", which represents your character in the game has already been defined. After a name has been entered the **Object Editing Menu** will appear.

Modify:

The modify command allows an already created object to be modified. It will display a list of all defined objects together with a number by each object in reverse video. An object is selected by pressing the number beside it. The system then goes to the **Object Editing Menu** to allow the object to be modified.

Object Editing Menu

The object editing menu allows all the parameters of an object to be altered. It displays the commands:

Name, Description, Score
Commands, Position, Weight, Exit

Name:

This allows the name of an object to be edited; i.e. to correct a spelling error.

Description:

Each object has a number of alternate descriptions which are displayed in preference to the name of an object in an inventory. Each description corresponds to a status value. Each object has a status which is initially 1 and is modified by commands. When this command is used the system will ask:
Add description Y/N?
If Y is answered the question
new status Y/N?
will be displayed. The default status is 0 (any value). The system will then prompt:
edit description Y/N?
A Y reply allows the description to be altered in a similar way to room descriptions. The system

will then return to the *add description?* prompt. Replying N takes you back to the **Object Editing Menu.** Editing an existing description gives a message such as:
Edit description Status 002 Y/N
N moves on to the next in the list but if you are at the end it goes to *Add description?* Y allows the description to be altered as described above.

Score:

This allows the score given for picking up an object to be edited in the same way as the score for entering a room.

Commands:

Each object can have associated with it a number of command verbs forming the verb list. This command allows these verbs to be modified. The command will prompt:
Add verb Y/N?
A reply of Y will result in a request to enter a verb. The name of the verb you wish to define should be entered. A name of more than 8 characters will be truncated. Having done this the **Verb menu** will appear. If there are already one or more verbs associated with an object the message:
edit verb VERBNAME Y/N?
will appear. Replying Y goes to the **Verb menu** to allow this verb to be modified. Replying N moves you along the list of verbs.

Position:

This command allows the initial position of an object to be selected using a **Position Menu**. Altering the position of the predefined object ".player" allows the starting room of the game to be selected. The default position of any object is the room where it was created.

Weight:

Each object has a weight which is selected using this command. The player may carry objects with a total weight of 500 at any one time.

Exit:

This returns to the **Edit screen**.

Position menus

A position menu allows the entry of a position. It displays the following:

Position? Any, Carried, Here Select.

Any:

This selects position (1,0) which matches any position in a condition check.

Carried:

This selects position (0,0) which indicates that an object is being carried by the player.

Here:

This selects the room marked by the cursor.

Select:

This allows selection of an arbitrary room. The system will prompt:

Enter Y position

and then

Enter X position

Room (3,0) corresponds to the room with the player in, i.e. moving an object to room (3,0) is equivalent to dropping it.

The Verb Menu

This menu allows a verb to be defined or altered. It produces the message:

Edit Name, Condition, Actions Exit

Name:

This allows the name of the verb to be modified.

Condition:

This allows a condition to be specified for the verb to be executed. It causes entry to the **Select Condition Sequence.** If at play-time a verbs condition is not satisfied the game will continue to look for another occurence of the verb. It is not neccesary to check for the presence of the object being operated upon as this is carried out automatically.

Actions:

This command allows the actions to be performed by a verb to be defined or modified. It causes entry to the **Select Command List sequence**

Exit:

This command causes a return to the calling menu.

The Select Condition Sequence.

This sequence is called by a number of menus and allows a condition to be set up for an action to happen.

The system will first ask:

New object Y/N?

Replying Y will display a list of all defined objects together with a number by each object in reverse video. An object is selected by pressing the number beside it. Replying N caused the object previously selected to be used. The system will then ask:

new status Y/N?

If Y is replied the system will ask for a status. The default value is 0, which will cause any status value to be accepted as satisfying the condition. If another value is entered the object selected will be required to have this status to satisfy condition checking.

Finally the system will ask:

new position Y/N?

Replying Y will cause the **Position menu** to appear. If position *Any* is selected any position will satisfy the condition, otherwise the object has to be in the position selected.

The Select Command List Sequence.

This sequence is called by a number of menus and allows a list of commands to be created. Commands are provided to print messages, alter the dynamic characteristics of objects etc.

On entry with a new command list the system will ask:

Add command Y/N?

Replying Y causes the menu:

S ay, D ie, M ove, s T atus, J ump

Command Menu

to appear.

Say:

This allows a message to be output on executing the list.

The system will reply:

enter text to SAY:

Type in a text, just as for a room or object description

Die:

Execution of this command results in the demise of the character immediately after the command list has been executed.

Move:

This command causes an object to be moved to a new position.

The system will prompt:

new position Y/N?

If Y is entered the system will go to the **Position menu** to allow a new position to be selected. The system will now reply:

new object Y/N?

If Y is replied an object may be selected and this object will be moved to the position previously entered.

sTatus:

This allows the status of an object to be set to a given value when this command is executed.

The system asks:

new object Y/N?

If Y is replied an object may be selected to have it's status altered.

The system will then ask:

new status Y/N?

If you reply Y to this the status can be modified to the value you wish the object to take on.

Jump:

This causes the player to move to the position specified. The command causes entry to a **Position menu** to allow a position to be entered.

Having defined a command the *Add command?* message will be redisplayed to allow more commands to be added to the list. If N is replied here you will return to the calling menu. When a command list already exists a message such as:

Command: Say,

D elete, E dit, N ext

will appear, where Say is the command being looked at. **Delete** removes the command from the list. **Edit** allows the command to be modified. **Next** moves along to the next command.

If you move to the end of the list using *Next* the *Add command?* message will reappear.

The Picture Creator

This is called from the **Centre menu** and is a powerful facility for defining hi-res multi-coloured pictures which are displayed when you enter a room.

On entry to the picture creator it will ask:

Select background colour:

The background colour is selected using the colour keys, 1-blue, 2-red etc.

If a picture already exists this message will be skipped and the system will go straight to the editing screen.

The screen display shows an area where the picture may be drawn and a status line. A cross is shown marking the current pen position. The coordinates of the cross are displayed on the status line. If a star appears by a coordinate this means that the cross is on a character boundary. The cross may be moved around the screen by using the cursor keys.

The Spectrum can only display two colours in one character cell(8x8 pixels). The system attempts to minimise the difficulty caused by this by maintaining a priority list of colours. If an attempt is made to draw in a colour not already present in a cell the colour with the lowest priority will be displaced. The colours in the priority list are displayed as round blobs on the left of the status line. The ink colour is the leftmost blob. To select a new ink colour SYMBOL SHIFT and the appropriate number key are used.

The **m** key marks the cross position with a second cross. The **d** key draws a line from this marked point to the current pen position and marks that position. The **p** key paints a triangle with one side the last line drawn, and the other two sides lines joining the ends of this line to the current pen position.

The ENTER key exits back to the centre menu. Pressing CAPS SHIFT and O DELETE deletes the last operation thus allowing correction of errors. Pressing SYMBOL SHIFT and X deletes the entire picture and returns to the **centre editing menu**.

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES