

I NCENTI VE

Home Computer Software

The Adventure Creator

BBC & Electron



THE ADVENTURE CREATOR

For the Acorn Electron

CONTENTS :

SECTION ONE

1.1	Introduction	PAGE	3
1.2	Loading Instructions	PAGE	4

SECTION TWO - "AC"

2.1	Writing Adventures	PAGE	5
2.2	Verbs	PAGE	5
2.3	Room Descriptions	PAGE	6
2.4	Messages	PAGE	6
2.5	Nouns And Objects	PAGE	7
2.6	Adverbs	PAGE	8
2.7	Conditions	PAGE	8
2.8	Begin Where?	PAGE	12
2.9	Save And Load	PAGE	12
2.10	Delete Data	PAGE	13
2.11	Printer Menu	PAGE	13
2.12	Creating Multi-Part Adventures	PAGE	14

SECTION THREE

3.1	Test Adventure	PAGE	15
3.2	Player Commands	PAGE	15

APPENDIX A

APP A	Definitions and Conditions for the Demonstration Adventure	PAGE	17
-------	---	------	----

APPENDIX B

APP B	Tables and Charts	PAGE	21
-------	-------------------	------	----

(1.1) INTRODUCTION

=====

Welcome to the ADVENTURE CREATOR!

This manual is intended as a simple introduction to the writing of adventures using AC. The ADVENTURE CREATOR is best thought of as a small programming language specifically designed for the writing of adventure games. Although it is smaller than other programming languages such as Basic or Pascal. It can perform all of the complex tasks needed to write an adventure program; and although the instructions may seem a little complicated at first, a little time spent studying this manual will enable you to write your very own large scale adventures, far more easily and efficiently than in basic or machine code.

Although the ADVENTURE CREATOR was written with the Electron in mind, it may also be run on the BBC, (including the BBC Plus and Master). When this is done, an extra 7000 bytes are available for the adventure (since it will automatically use a Mode 7 screen). The resulting (runnable) adventure will still run on the Electron, since it reclaims the memory used by the editor.

The datafiles created by AC are compatible with those of the GRAPHIC ADVENTURE CREATOR so long as the memory limitations of either are not exceeded. Hence, an adventure created on the Electron may be transferred to GAC and have pictures added to run on the BBC. The only exceptions are the conditions 'CHN' which is ignored by GAC, and 'TEXT' and 'PICT' which are ignored by AC.

Any adventures that you write using AC are your own work, and may be sold as such without any prior permission being sought or payment to us being made. However in this case, you should include with your adventure something to the effect that it was written using the ADVENTURE CREATOR (c) 1986 by Sean Ellis/Incentive Software.

And now on with AC!

CREDITS

DESIGN	by Sean Ellis
PROGRAMMED	by Cornucopia Software
COVER	by Pete Carter

Thanks also to Lesley and John.

COPYRIGHT 1986 INCENTIVE SOFTWARE LTD.
54 LONDON STREET, READING RG1 4SQ

All rights of the producer, and of the owner of the work being produced, are reserved. Unauthorised copying, hiring, lending, public performance and broadcasting of this program is prohibited. The publisher assumes no responsibility for errors, nor liability for damage arising from its use.

(1.2) LOADING INSTRUCTIONS
=====

The ADVENTURE CREATOR, (AC), files supplied are as follows:

- | | |
|-------------|---------------------------------------|
| 1. "AC" | THE ADVENTURE CREATOR - See section 2 |
| 2. "QS" | The Quickstart File |
| 3. "ADVMAN" | The Adventure in the Manual |
| 4. "RANSOM" | A Demonstration Adventure |

Loading

To load "AC" type CHAIN "AC" <RETURN>

To load "RANSOM" type CHAIN "RANSOM" <RETURN>

To load "QS" and "ADVMAN" see below

"QS" and "ADVMAN"

Files "QS" and "ADVMAN" can be loaded by selecting T from the main menu in "AC".

QS - Quickstart, sets you up with the most commonly used verbs etc and common messages. A complete listing of its contents is contained in Appendix B, along with several other useful tables and charts.

ADVMAN - Sets up the small adventure in the manual to work through and even edit! (See Appendix A)

(2.1) SECTION TWO - WRITING ADVENTURES

=====

There are several elements necessary to the writing of adventures using AC. These are displayed on the Main Menu which is the first thing you will see after loading up.

For convenience, the menu is arranged in alphabetical order, to make it easier for you to remember which keys to press, to call up any part of AC.

You may write the elements of your adventure in any order you like, although you may find that some orders may make things easier than others.

What follows is a brief description of what each element of the Main Menu does, and detailed instructions of how to enter each one, in an order that you might choose to enter them.

(2.2) VERBS

=====

Verbs are the words with which you will actually tell the computer what to do when you are playing an adventure. For example EAT or DROP. Directions such as WEST or UP are also included. AC allows you to enter and edit your own verbs when writing an adventure. Some of the most commonly used verbs are already in the Quickstart File; however since AC allows you to enter up to 255 verbs, your imagination is the only limit to the commands you can give your adventure.

After pressing V on the Main Menu, you should see a prompt on the screen *EDIT VERBS* and a pointer >.

To enter a verb, simply type in a number then a space followed by your verb. When you press <RETURN>, the verb will be entered on the screen in alphabetical order.

If two or more verbs have the same meaning and you wish both to be accepted i.e. "Get" and "Take", simply allocate them the same verb number. Try to think of as many similar words as you can in this way, as this will make your adventure more friendly.

To edit a verb you can do one of two things. If you press <RETURN>, this frees a small cursor which can be directed by means of the arrow keys to the verb you wish to edit. Characters may then be copied by use of the <COPY> key to the editing line at the bottom and changed. More simply perhaps, the offending verb can simply be retyped and entered in the normal way. The arrow keys may then be used to move the pointer to the previous version for deletion, using <DELETE>. If you have deleted a verb by mistake, <COPY> will retrieve the last work deleted.

There is a brief summary of these instructions at the bottom of the screen.

More information about the use of verbs may be found in section (2.7) CONDITIONS.

PRESSING ESCAPE WILL RETURN YOU TO THE MAIN MENU

(2.3) ROOM DESCRIPTIONS

=====

The term room can be used to refer to any location in your adventure. For example, "a forest", "a car", "a strange alien world" and so on. AC allows you, memory permitting, to use up to 9999 of these rooms, and to specify the connections to and from each room. You will probably find it easier if you have decided in advance, what each room is and how they are connected.

On pressing R on the main menu, the prompt WHICH ROOM NO? should appear. At this point you can either, enter a room number followed by <RETURN> which will give the prompt ROOM NUMBER # IS... , or you can simply press <RETURN> which will give the same prompt, but stepping through each time you do it, from number 1. You should then either, enter your description which can be up to 255 characters followed by <RETURN>, or simply press <RETURN> if there is no description.

You will then see the prompt CONNECTIONS ARE? These connections are specified in terms of a VERB, which must be from the list already specified, followed by a space, followed by the room number it connects with. For example EAST 20.

You may have as many connections from a single room as you can fit into 255 characters, all on a single line and followed by <RETURN>. i.e. EAST 20 WEST 18 NORTH 19 SOUTH 21 JUMP 49 (<RETURN>).

All of the above may be edited at any time by using the arrow keys to separate the cursors, and moving the smaller one to the part to be edited. The <COPY> key copies characters to the editing line at the bottom of the screen, where they may be edited in the usual way. <RETURN> enters the edited version. Pressing f0 will delete the entire entry for that particular room. <ESCAPE> at any point will return you to the WHICH ROOM? prompt, and <ESCAPE> at that point will return you to the Main Menu.

(2.4) MESSAGES

=====

Messages are the pieces of information, prompts and so on, that appear on the screen when you play an adventure. For example WHAT NOW? or YOU CAN'T DO THAT. Many of the most common system messages, numbers 238 to 255, such as these are contained in the Quickstart file. AC allows you to enter up to 255 messages that can appear at any point in your adventure.

When you press M on the Main Menu, the prompt asking for a message number will appear. You should then enter a number followed by your message, this may be up to 255 characters long. Pressing <RETURN> will enter your message and prompt for the next message number.

To edit a message once it has been entered, simply recall the message and edit it in the usual way. (Refer to 2.3 Room Descriptions).

Please note that messages 238 to 255 should all be defined, since these are essential system messages without which, your adventure will not run. You can edit these messages, but you should be careful to preserve the meaning. For example, message 242 PARDON? could be

changed to YOU WHAT? (All these messages are set up in the Quickstart file).

Pressing <ESCAPE> will return you to the prompt EDIT MESSAGE? and <ESCAPE> from there will return you to the Main Menu.

(2.5) NOUNS AND OBJECTS

=====

Nouns are the "things" that your adventure will recognise. For example BOOK, SWORD and so on. You should not be too confused by the fact that AC frequently requires you to define things both as nouns and as objects: once as a noun so that the adventure will recognise it, and then as an object so that the program can move it around and do things with it. For example, SWORD is defined as a noun, but since you may wish to pick it up during your adventure, you must also define it as an object. Another example makes clear another point. If your adventure requires you to light a lamp, LAMP must be defined as a noun, and you must also define two objects, AN UNLIT LAMP and A LIT LAMP.

When you press N on the Main Menu, the prompt *EDIT NOUNS* will appear, Apart from this, the procedure for entering and editing nouns is exactly the same as that for editing verbs. (2.2)

When you press O on the Main Menu, the prompt WHICH OBJECT NUMBER? will appear. You should then enter a number from 1 to 255 followed by <RETURN>. You will then see the prompt "OBJECT # IS....".

You can then enter the object description of up to 255 characters followed by <RETURN>. This will produce the prompt "Starts in room #". Enter the location number of the room you wish this object to start off in. The prompt asking for the weight will then appear. Enter the weight you wish to allocate this object and press <RETURN>.

If the start room number remains at zero, the object is assigned to ROOM ZERO ; a special room which cannot be travelled to, which contains "dead" and "unborn" objects. For example, a piece of cake once eaten will be assigned to room zero.

<ESCAPE> will return you to the Main Menu.

More information about the use of nouns and objects can be found in section (2.7) CONDITIONS.

(2.6) ADVERBS

=====

In AC adverbs are used for two sorts of things. Firstly to describe the precise way a verb is used. For example, MOVE SLOWLY or PUT DOWN GENTLY.

Secondly they are used to differentiate between similar nouns. For example if your adventure has three differently coloured boxes, the adverbs will specify the colours, RED, GREEN, etcetera.

On pressing A on the Main Menu, the prompt *EDIT ADVERBS* will appear. Apart from this the procedure for entering and editing adverbs is exactly the same as that for editing verbs and nouns.

More information on the use of adverbs can be found in section (2.7) CONDITIONS.

(2.7) CONDITIONS

=====

When using AC the conditions form the internal commands that simulate decision making processes within the game.

All the terms you have specified and the data you have entered is now drawn together to form the structure of your adventure.

Conditions tell the adventure that if certain conditions are met, then it is to do something. i.e. "If x is true then do y".

Or you can build up conditions to a more complex level.

i.e. "If x and y are true then do t",
or "If x and q are true then do y and x", and so on.

AC has a special format for writing these conditions. For example, if you have defined a verb number sixteen to mean "Examine" and a noun number 2 to mean "Room", then a condition using these terms might read:

"IF(VERB 16 AND NOUN 2) LOOK WAIT END"

Translated this means if "Examine Room" is typed in, describe the current room and wait for a new command.

The Conditional list below separates the words used for the "Conditions Part" e.g. If (Verb 16 and Noun 2) and those used for the "Action Part" after the first close bracket), i.e. LOOK WAIT END Subscripts used (a, c, m, ms, o, n, r, v, x and y) - See Appendix B.

i. CONDITION PART

VERB v	Is verb v typed ? (v is a number) VERB 7 will give a TRUE answer if verb number 7 was typed, otherwise it will give a false answer.
NOUN n	Is noun n typed ? (similar to VERB)
ADVE a	Is adverb a typed ? (similar to VERB)
HERE o	Is object o here ? (i.e. in the same room as you?) If you are in room 3 and object 1 is also in room 3, then HERE 1 will give a TRUE answer because object 1 is here.
CARR o	Is object o being carried ?
AVAI o	Is object o available for use (i.e. here or being carried)
r IN o	Is object o in room r ? If object 1 is in room 3, as above, then 1 IN 3 will be TRUE, but 1 IN 4 will be FALSE.

WEIG o Gets the weight of object o. In our adventure, WEIG 2 would give the value 20, which is the weight of the rat.

SET? m Is marker m set ?

RES? m Is marker m reset ?
 (There are 256 markers, numbered 0 to 255. They are used to store information that can be in one of two states, like doors which are open or shut, lights which are on or off, etc. There are three which are important for the adventure program :
 Mkr 0, if set, means that a room has been described since it was last reset.
 Mkr 1, if set, means you are in a light room. Otherwise you are in a dark room.
 Mkr 2, if set, means you have a lamp or some other source of light.
 If markers 1 and 2 are both reset, then the program will refuse to describe rooms, coming up with the "It's dark" message instead, since you are in a dark room without a lamp.
 Mkr 3, if set, disables the scoring mechanism when you exit from the game.

CTR c Gives you the value of counter c.

x EQU? c Is x equal to the value of counter c ?
 There are 128 counters, numbered from 0 to 127. They are most frequently used to store the number of moves since a particular event, (e.g. in the dark). Counter 0 holds the score and counters 126 and 127 hold the turns count since the start of the game.

TURN Gives the number of turns since the start of the game.

ROOM Gives you the room number of the room you're currently in.

AT r Gives a TRUE answer if you are at room number r.

condition AND condition will give a TRUE answer if both of the conditions give TRUE answers.
 e.g. VERB 1 and NOUN 2 will be TRUE only if verb 1 and noun 2 are both typed.

condition OR condition will give TRUE if either condition is TRUE, or both.

condition XOR condition will give TRUE if one of the conditions is TRUE and the other FALSE.

NOT condition will give a TRUE answer if the condition was FALSE, and vice versa.
 So NOT VERB 1 will be TRUE if verb 1 is NOT typed.

x <= y Gives TRUE if number x is less than or equal to number y.

x >= y Gives TRUE if x is greater than or equal to y.

x = y Gives TRUE if x is equal to y.

RAND x Gives a random number between 0 and (x-1). So RAND 10 will give any number at random in the range 0 to 9.

VBNO Gets the number of the verb in this command.

NO1 Gets the number of the first noun in this command.

NO2	Gets the number of the second noun in this command. These are used to check word order, and to GET and DROP objects without having to have a condition for each one.
ii. ACTION PART -----	
LOOK	Describe the room you are in the moment.
DESC r	Describe room number r
PICT	{These do nothing on the ADVENTURE CREATOR but {retained for compatibility with GAC.
TEXT	
GET o	Get object number o. If it isn't here, or you've already got it, then the appropriate message is printed.
DROP o	Drop object number o . If you haven't got it, then the appropriate message is printed.
x SWAP y	Exchange objects x and y. In our little adventure, 1 SWAP 5 will exchange the lamp and the lit lamp.
OBJ o	Describe object number o.
LIST r	List all the objects in room number r.
LIST WITH	Lists all the objects carried with you.
o TO r	Move object number o to room r. To destroy an object, move it to room 0.
SET m	Set marker m.
RESE m	Reset marker m. See SET? and RES? for an explanation of markers.
x CSET c	Set x to be the value of counter number c.
INCR c	Increase counter c by one. The maximum is 255.
DECR c	Decrease counter c by one. The minimum is 0.
	Trying to increase past 255 or decrease past 0 is ignored.
GOTO r	Go to room r and describe the new room.
CONN v	This checks through the connection table for a connection from the current room using verb v. If one is found, this gives the room number of the room you would move to if you took the connection. If not, then it gives zero. As an example, if you were in room 3 in our adventure, CONN 4 would give 2 because in the connections from room 3, verb 4 (W) would take you to room 2.
STRE x	Set the maximum weight you can carry to x. STRE is short for STRENGTH.
BRIN o	Brings object o here (if it exists)
FIND o	Find object o and move to it (if it exists). This does not acknowledge any restrictions (such as connections), so it could be useful in a magic spell or something.
SAVE	Saves the current game position.
LOAD	Loads the current game position.
	These are useful for continuing a game after tea or after you get killed!
WAIT	Waits for a new command.
OKAY	Prints "Okay" and awaits a new command.
EXIT	Stops the game. The player is NOT asked, so this is for use when the player gets killed or wins.

QUIT	The player is asked if he wants to continue, and if he typed "N" for "NO" when asked if he is sure, the game continues, otherwise the game is abandoned. On abandoning the game, the score and number of moves taken is displayed, if not disabled by setting marker three.
CHN	Load next part of the adventure from tape. See section 2.12.
MESS ms	Print message number ms.
PRIN x	Print number x.
LF	Prints a LineFeed. Everything from here is printed on a new line.
WITH	is equal to the room number of where things are out when you are carrying them.
HOLD x	Holds up the game for x hundredths of a second. For example, to freeze for 10 seconds, do HOLD 1000. (Or until a key is pressed).
x + y	As you may expect, this returns the value of x added to y.
x - y	And this gives the value of x - y.

There are several things to note when writing conditions. Firstly all operations are performed from left to right. The second thing to note is that AC requires spaces both inside and outside brackets, and also between words such as 'verb' and 'noun' and their numbers and finally that Conditions should be entered in Upper Case.

Thirdly, x and y need not be simply numbers, they can be more complex expressions. However you should note that since GAC performs all its operations from left to right, you should be careful about the order in which you enter things. Put any COMPARISONS first, i.e. IF (VERB >= 5 AND NOUN 9) QUIT END

The conditions are checked by AC at three different points, and are entered as High Priority, Low Priority and Local Conditions. See the flow chart at the back of the manual to see the order in which these are actioned.

HIGH PRIORITY conditions are checked for BEFORE the adventure prompts for a player command. It checks for such things as whether you are still "alive", whether a light is flashing, etc.

Pressing H on the Main Menu will get the prompt "Line Number?" You should enter a number in order, i.e. 1, 2, 3, etc, etc.

AC will not recognise a line number such as '5' when you have only entered two conditions so far, although it will insert a condition earlier in your list and remember accordingly.

<RETURN> will give you the prompt "Line number _ is....", at which point you should enter your condition and press <RETURN> which will give you the "Line Number" prompt again.

<ESCAPE> returns you to the Main Menu.

LOCAL CONDITIONS are checked for after the player has entered a command. They deal with things local to a particular room, for example whether a particular object is in a room, or it might execute a player command to go to another room, or open a door at that location.

Pressing C on the Main Menu will give the prompt "Room Number?" Entering a number followed by <RETURN> will give the prompt "Line Number?" Thereafter the procedure is exactly the same as for High Priority Conditions, except that each Room has its own set of line numbers (1, 2, 3, etc).

LOW PRIORITY CONDITIONS are also checked for after a player command is entered, but are not associated with any particular room. For example, they might check whether you are carrying a particular object, independent of the location.

After pressing L on the Main Menu, the procedure for entering and editing Low Priority Conditions is exactly the same as that for High Priority Conditions.

Further examples of High, Local and Low Priority Conditions can be found in Appendix A.

(2.8) BEGIN WHERE? =====

The last element of your adventure, at least as far as the text is concerned is to specify which room your adventure starts in.

Pressing B on the Main Menu will give the prompt to enter start location number.

Entering a room number followed by <RETURN> means that your adventure will start at this room. To change the start room, simply enter a new number.

<ESCAPE> will return you to the Main Menu.

(2.9) SAVE AND LOAD =====

SAVING -----

Pressing S on the Main Menu, you will be asked if you wish to save a "Data File" or a "Runnable Adventure".

If you press D for 'Data File', it will prompt for a file name. Enter your file name followed by <RETURN>, this will save your Data which can be loaded into AC at a later date for further development.

If you press E for runnable adventure followed by <RETURN>, your data will be saved as a Runnable Adventure, which can no longer be loaded into AC for editing but will run on its own.

LOADING -----

Pressing T on the Main Menu followed by a file name will load a Data file into AC for editing.

To play your runnable adventure, enter *RUN filename

NOTE : Do not attempt to load a runnable adventure into AC. It
 will run on its own!

(2.10) DELETE DATA

=====

Pressing X on the Main Menu will delete all your data.
N.B. Loading a new data file will have the same effect.

(2.11) PRINTER MENU

=====

Pressing P on the Main Menu will give you the Printer Menu. Select section for printing by pressing the appropriate key. To print to your printer, press P at the appropriate prompt, or print to the screen, press S. (The <SHIFT> key will scroll the screen if necessary).

<ESCAPE> will return you to the Main Menu.

(2.12) CREATING MULTI-PART ADVENTURES

=====

The AC uses extensive text compression which enables commercial sized adventures to be written. However, for large adventures AC allows multi-part adventures to be created using the 'CHN' command.

When CHN is executed in a line of conditions, the program will look for a load a new datafile. The information on the state of play is preserved, (i.e. the current room, player's strength, weight carried, all flag and counter settings and the position of all objects, including those being carried). The rest of the condition line is ignored (since it has now been written over), and play continues from the start of a new turn.

The CHN command should be used with caution and the following points noted:

- 1) The objects are NOT initialised to their start rooms after a CHN. Any objects that exist in the second part of the adventure that were not in the first part will start off in room 0. (i.e. not created). Objects and their matching nouns that you want to start in a specific room should be defined in the first part AS WELL AS the second.
- 2) Going to a room that is not defined in the current module will give a "Room not found" error. This can cause problems if a player SAVES his or her position whilst in the second part of the adventure and later tries to LOAD it into the first part. There are two methods of avoiding this:
 - a) Giving the player the option of loading the second part before loading his position (perhaps by giving him a password).
 - b) If, for example, the first part consists of rooms 1-100 and the second of rooms 100-200, then a HIGH priority condition in the first part of: IF (ROOM >= 100) CHN END will automatically chain the second part of the adventure when a game position is loaded.

To create a runnable multi-part adventure, save the first part with the R (runnable) option and the following part(s) with the D (data) options consecutively on tape.

EXAMPLE: A two-part adventure consists of rooms 1-100 in the first part and rooms 100-200 in the second. Room 100 is common to both parts and is used to link the two sections.

DATA FOR FIRST SECTION

Room 100:

As you step onto the pentagram, the Wizard says "You have done well, but far greater perils await before you complete your quest." Your surrounding fade to be replaced by a strange alien landscape...

CONNECTIONS none

MESSAGE 99: Loading next section. Insert tape and press PLAY.

HIGH PRIORITY CONDITIONS: IF (ROOM >= 100) MESS 99 CHN END

DATA FOR SECOND SECTION

Room 100: You are on an open plain of black crystal. The sunless sky is of constantly shifting colours. The crystal throbs gently beneath you.

CONNECTIONS E102 W105

The loading message and condition line are not needed in section 2, (unless a third section is to be chained).

In the above example, when the player reaches room 100 in the first section the room description is followed by the loading message. When the second section has loaded the new description is printed.

(3.1) SECTION THREE - TEST ADVENTURE

=====

To test your adventure, get back to the Main Menu and simply press <RETURN> to enter the adventure.

When you are testing an adventure, the way to return to the Main Menu is to press <ESCAPE> as usual. This gives you the message "Press D for diagnostics or "ESC" to ESCAPE...". If you press the D key, then all the makers and counters will be displayed on the screen. For the markers, a one indicates "SET" and a zero indicates "RESET". The values of the counters are given as you would expect. To get back to the game, press <ESCAPE>.

If you press any other key again then you will get back to the Main Menu.

The error routine prints out a message indicating the nature of the error, plus a line of conditions if the error occurred in one of the condition tables.

Message not found	means you have referred to a message that does not exist. If you get this immediately, check that you have entered the system's special messages at numbers 240 and above.
Room not found	means you have tried to describe, or move to, a room that doesn't exist.
Object not found	means you have tried to pick up, drop or describe an object that does not exist.
Marker not found	means you have tried to access a marker which does not exist.
Counter not found	means you have tried to access a counter which does not exist.
Illegal value	means you have tried to look for a verb, noun or adverb with a number greater than 255, or you have tried to load a counter with a number greater than 255/

Here are a few hints for solving errors :

- i. Suspect typing error. Check that you really mean what you have said.
- ii. Have you forgotten to enter the room/object/message being referred to? If so, enter it.
- iii. Have you forgotten to delete this condition which you didn't need, having deleted the objects/messages used by it?

(3.2) PLAYER COMMANDS

=====

This section looks at the commands a player gives when he plays the adventure. It is intended to give an idea of the complexity of the commands the adventure will understand.

Each command line consists of one or more simple commands, which in turn consist of a verb, and maybe an adverb and one or two nouns. Any word which the command interpreter does not understand as being a noun, verb or adverb is ignored.

Let us take a typical line and see how the program looks at it.

Get the gold, examine it, put it in the box then go north

Let us assume that the vocabulary includes :

NOUNS :	3 GOLD	VERBS :	1 NORTH	ADVERBS :	1 IN
	7 BOX		7 GET		
	255 IT		8 PUT	16 EXAMINE	

The command line is split into separate commands by the following :
 ".", ",", "!", "?", ":", ";", "and" and "then", so it becomes :

```

Get the gold      ","
Examine it       ","
Put it in the box "then"
Go north

```

Each of these is scanned for verbs, adverbs and nouns in that order, and any found have their number stored.

	VERB	ADVE	NOUN1	NOUN2
Get the gold	7	0	3	0

Examine it	16	0	255	0

Put it in the box	8	1	255	7
--- --				
Go north	1	0	0	0

Then all occurrences of noun 255 ("it") are replaced by the last noun typed before that, giving

	VERB	ADVE	NOUN1	NOUN2
Get the gold	7	0	3	0
Examine it	16	0	3	0
Put it in the box	8	1	3	7
Go north	1	0	0	0

These values are then passed to the connection table, and then to the conditions. See diagram in Appendix B.

Note that all the letters in the vocabulary entries are significant - there is no truncation to only four or five letters. Thus RIVER and RIVET, TROUT and TROUSERS, and others like these are distinguished between.

APPENDIX A

=====

Definitions and Conditions for "ADVMAN"

i. NOUNS

1 TORCH 5 SERPENT
1 LAMP 6 DOOR
2 RAT 255 IT
3 KEY
4 GOLD
4 BAR
4 TREASURE
5 SNAKE

ii. VERBS

1 NORTH 9 LOOK 17 ON
2 SOUTH 10 LIST 18 EXTINGUISH
3 EAST 10 INVENTORY 18 OFF
4 WEST 11 QUIT 19 UNLOCK
5 UP 14 SAVE 20 SCORE
6 DOWN 15 LOAD 21 EAT
7 GET 15 RESTORE
7 TAKE 16 EXAMINE
8 DROP 17 LIGHT
1 N 2 S 3 E 4 W
5 U 6 D

iii. ROOM DESCRIPTION

Room No.	Description (as it would appear)	Connections
1.	You are above the ground. There is a cave entrance to the east.	(Press <RETURN>)
2.	You are in a large cavern. Passages lead east, west and south.	E 3
3.	You are in a cave. A snake is asleep in a corner and exits lead east and west.	W 3
4.	You are by a small lake. The only exit is west.	(Press <RETURN>)
5.	You are outside a castle. A tunnel leads to the north and a large door can be seen to the east.	(Press <RETURN>)
6.	You are in the castle strongroom. A door stands open to the west.	W 5

iv. OBJECTS

No.	Description	Starts in room	Weight
1.	a lamp	1	10
2.	a dead rat	5	20
3.	a key	4	1
4.	a gold bar	6	100
5.	a lit lamp	0	100

v. MESSAGES

238 nothing
239 You are carrying
240 What now ?....
241 You can't
242 Pardon ?
243 Press a key for another game...
244 Are you sure (Y or N) ?
245 You've already got that.
246 You haven't got that.
247 You can't see that.
248 You're carrying too much to pick that up.
249 Your score was
250 and you took
251 It's dark. You can't see a thing.

```

252 I can't find that anywhere.
253 You can also see
254 Okay
255 turns

```

i. Conditions - Local Don't type in the comments !

Room 1

```

IF ( VERB 3 ) RESE 1 GOTO 2 WAIT END
    If you typed "EAST", reset the dark/light market, goto room 2
    and wait for a new command.

```

Room 2

```

IF ( VERB 4 ) SET 1 GOTO 1 WAIT END
    If you typed "WEST", set the dark/light marker, goto room 1 and
    wait for a new command.
IF ( VERB 2 ) SET 1 GOTO 5 WAIT END
    And a similar condition for going south too.

```

Room 3

```

IF ( VERB 7 AND NOUN 5 ) MESS 14 HOLD 200 EXIT END
    If you typed "GET SNAKE" then print message fourteen to say how
    it reacts. Freeze for four seconds and end the game.
IF ( VERB 3 ) SET 1 GOTO 4 WAIT END
    Again another move-from-dark-room-to-light-room condition, this
    time east.

```

Room 4

```

IF ( VERB 4 ) RESE 1 GOTO 3 WAIT END
    I'll let you work this one out (hint - look at rooms 1, 2, 3)
IF ( VERB 7 AND NOUN 3 ) CTR 0 + 20 CSET 0 END
    If you typed "GET KEY" then add twenty to counter 0 (the score)
IF ( VERB 8 AND NOUN 3 ) CTR 0 - 20 CSET 0 END
    If you typed "DROP KEY" then subtract 20 from ctr 0 (the score)

```

Room 5

```

IF ( VERB 3 AND SET? 3 ) GOTO 6 WAIT END
    If you typed "EAST" and marker 3 is set, (i.e. if the door has
    been opened), goto room 6 and wait for a new command.
IF ( VERB 3 ) MESS 7 WAIT END
    If you typed "EAST" then print message 7 and wait for a new
    command. Note that the door cannot be opened since if it was,
    the last line would have worked and we would be waiting for a
    new command by now.
IF ( VERB 19 AND NOUN 6 AND CARR 3 ) SET 3 MESS 10 WAIT END
    If you typed "UNLOCK DOOR" then set marker 3, (mark the door as
    open), print message 10 and wait for a new command.
IF ( VERB 1 ) RESE 1 GOTO 2 WAIT END
    It is another light-to-dark movement, this time north to room 2

```

ii. Conditions - Low Priority

Those marked "*" are included in the Quickstart file.

```

IF ( VERB 20 ) MESS 249 PRIN CTR 0 MESS 250 PRIN TURN MESS 255
WAIT END
    If you typed "SCORE", print message 249, your score, message
    250, the number of turns you had and message 255. Then wait for
    a new command.
IF ( NO1 = 0 AND VERB 7 ) MESS 19 WAIT END
    If you typed "GET" by itself or with an unrecognised word,
    print message 19 "I'm sorry, but I don't know what one of those
    is..."
IF ( NO1 = 0 AND VERB 8 ) MESS 19 WAIT END
    And similarly for "DROP".

```

```

IF ( NO1 = 0 AND VERB 16 ) MESS 18 WAIT END
    And again for "EXAMINE".
IF ( VERB 7 AND NOUN 1 AND HERE 5 ) GET 5 OKAY END
    If you typed "GET LAMP" and there is a lit lamp here, then get
    it, print "Okay" and wait for a new command.
IF ( NO1 <=5 AND VERB 7 ) GET NO1 OKAY END
    If you typed "GET" and a noun with a number less than 5, then
    get the object with that noun's number. This only works because
    the objects and the nouns which refer to them have the same
    number. (A very useful trick!)
IF ( VERB 8 AND NOUN 1 AND CARR 5 ) DROP 5 OKAY END
    If you typed "DROP LAMP" and you've got a lit lamp, then drop
    it, print "Okay" and wait for a new command.
IF ( NO1 <= 5 AND VERB 8 ) DROP NO1 OKAY END
    If you typed "DROP" and a noun whose number is less than 5,
    then drop the object with that noun number.
IF ( VERB 16 AND NOUN 1 AND AVAI 5 ) MESS 5 WAIT END
    If you typed "EXAMINE LAMP" and you have a lit lamp available
    then print out its more detailed description and wait for a new
    command.
IF ( NO1 <= 5 AND VERB 16 AND AVAI NO1 ) MESS NO1 WAIT END
    If you typed "EXAMINE" and a noun whose number is less than 5,
    then print out the message with that noun number, it being the
    more detailed description of that object, then wait for a new
    command.
* IF ( VERB 11 ) QUIT END
    If you typed "QUIT" then quit.
* IF ( VERB 9 ) LOOK WAIT END
    If you typed "LOOK" then describe this room and wait for a new
    command.
IF ( VERB 21 AND NOUN 2 AND CARR 2 ) DROP 2 2 TO 0 MESS 17 WAIT
END
    If you typed "EAT RAT" and you are carrying it, drop it, move
    to room 0, thus destroying it, print message 17 ("Yum, yum")
    and wait for a new command.
* IF ( VERB 10 ) MESS 239 LIST WITH END
    If you typed "INVENTORY" then list the objects with you and
    wait for a new command.
* IF ( VERB 13 ) PICT OKAY END
    If you typed "PICTURES" then turn them on, print "Okay" and
    wait for a new command.
* IF ( VERB 12 ) TEXT OKAY END
    And similarly for "TEXT".
IF ( VERB 17 AND NOUN 1 AND AVAI 1 ) 1 SWAP 5 CTR 0 + 20 CSET 0
MESS 15 SET 2 WAIT END
    Type this all on one line. If you typed "LIGHT LAMP" and you
    have an unlit lamp, then exchange the lit lamp for the unlit
    one, tell the user that he has lit the lamp, increase the score
    by 20 and await a new command.
IF ( VERB 18 AND NOUN 1 AND AVAI 5 ) 1 SWAP 5 CTR 0 - 20 CSET 0
MESS 16 WAIT END
    And similarly for "LAMP OFF" ("EXTINGUISH LAMP").
* IF ( VERB 14 ) SAVE OKAY END
    If you typed "SAVE" then save the game position.
* IF ( VERB 15 ) LOAD LOOK WAIT END
    If you typed "LOAD" then load in a previously saved game
    position.

```

iii. Conditions - High Priority

```
-----
IF ( RES? 6 ) SET 6 STRE 111 3 CSET 1 END
    If marker 6 is reset (if this is the first move) then set
    marker 6 to say that it isn't the first move any more, set the
    strength and counter one to hold value three.
IF ( RES? 1 AND RES? 2 ) DECR 1 END
    If you are in total darkness, then decrease counter number one.
IF ( 1 EQU? 1 AND RES? 1 AND RES? 1 ) MESS 20 END
    If counter one has reached value 1 then print that you can hear
    footsteps.
IF ( 0 EQU? 1 ) MESS 21 EXIT END
    If counter 1 has reached zero then tell the player he has been
    "got" by the spider, and end the game.
IF ( AT 3 AND RES? 4 AND CARR 2 ) SET 4 MESS 13 DROP 2 2 TO 0
WAIT END
    If you are in room three with the rat and the snake hasn't been
    fed yet, mark the snake as fed, print the message to say so,
    and move the rat to room 0, destroying it.
IF ( AT 3 AND RES? 4 ) MESS 8 EXIT END
    If you are in room three without the rat and the snake hasn't
    been fed yet, then say that it kills you and exit from the
    game.
IF ( AT 1 AND CARR 4 ) MESS 9 EXIT END
    If you are back at the start carrying the gold then you win!
```

GET AND DROP

```
-----
Here is another useful feature! To save putting a Get and Drop
condition in for each and every object, here are two very useful
condition lines that will enable you to get and drop any object (up
to No.10 in our example) at any location. To do this simply ensure
the Noun numbers match the Object numbers. i.e.
```

NOUN 1 = Hat OBJECT 1 = a hat

and enter Low Priority Condition lines :

```
IF ( NO1 <=10 AND VERB 7 ) GET NO1 OKAY END
IF ( NO1 <=10 AND VERB 8 ) DROP NO1 OKAY END
```

APPENDIX B
=====
Tables and Charts

THE QUICKSTART DATA FILE

The "Quickstart" data file contains all the system messages, many useful verbs and several of the common low priority conditions associated with them. This is to allow you to get straight in to writing the adventure that you want to write, without having to worry about things that are included in all adventures.

The full contents are:

VERBS -----	NOUNS 255 IT -----
6 D	
6 DOWN	MESSAGES
8 DROP	-----
3 E	238 nothing
3 EAST	239 You are carrying
16 EXAMINE	240 What now?
7 GET	241 You can't.
	242 Pardon?
10 INVENTORY	243 Press a key for another game...
9 L	244 Are you sure? (Y/N)
10 LIST	245 You've already got that.
15 LOAD	246 You haven't got that.
9 LOOK	247 You can't see that.
1 N	248 You're carrying too much to pick that
1 NORTH	up.
	249 Your score was
11 QUIT	250 and you took
15 RESTORE	251 It is dark. You can't see.
2 S	252 I can't find that anywhere.
14 SAVE	253 You can also see
2 SOUTH	254 Okay
7 TAKE	255 turns
5 U	
5 UP	
4 W	
4 WEST	

LOW PRIORITY CONDITIONS

```

IF ( VERB 9 ) LOOK WAIT END
If you typed "LOOK", re-describe the room you're in and wait for a
new command.
IF ( VERB 10 ) MESS 239 LIST WITH WAIT END
If you typed "INVENTORY" then print "You are carrying" and list
all the objects that are with you.
IF ( VERB 11 ) QUIT OKAY END
If you typed "QUIT" then ask the adventurer if he is sure, and if
he responds Y (for YES) then quit, otherwise print "Okay" and wait
for a new command.
IF ( VERB 14 ) SAVE OKAY END
If you typed "SAVE" then save the game position.
IF ( VERB 15 ) LOAD LOOK WAIT END
If you typed "LOAD" then load a previously saved game position and
describe the room you are in.
```

ii. SUBSCRIPTS USED WITH CONDITION WORDS

a - adverb number
 c - counter number
 m - marker number
 ms - message number
 o - object number
 n - noun number
 r - room number
 v - verb number
 x,y - any numbers

iii. WORDS USED IN CONDITIONS

Words	Markers	Counters	Rooms	Commands
-----	-----	-----	-----	-----
VERB v	SET m	CSET c	GOTO r	OKAY
NOUN n	RESE m	INCR c	CONN r	WAIT
ADVE a	SET? m	DECR c	ROOM	EXIT
NO1	RES? m	CTR c	AT r	QUIT
NO2		x EQU? c	DESC r	
VBNO		TURN	LOOK	

Objects	Decisions	Tape
-----	-----	-----
GET o	OBJ o	IF
DROP o	LIST o	END
o SWAP o	WEIG o	SAVE
HERE o	STRE o	LOAD
CARR o	o TO r	
AVAI o	BRIN o	
r IN o	FIND o	

Other

MESS ms	x = y
x + y	WITH
x - y	RAND x
condition AND condition	PRIN x
condition OR condition	HOLD x
condition XOR condition	LF
NOT condition	CHN
x >= y	
x <= y	

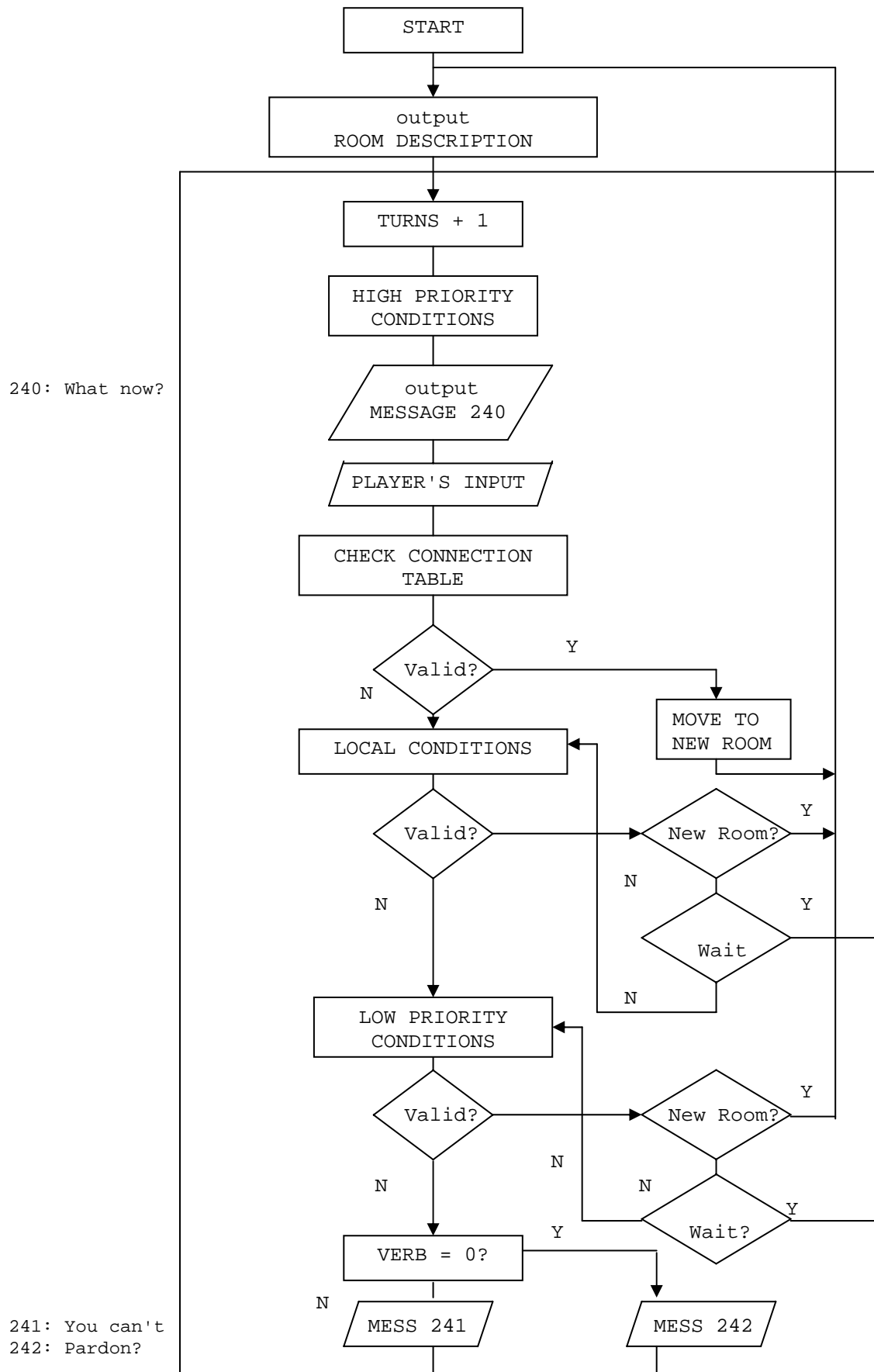
iv. COUNTERS AND MARKERS USED BY THE SYSTEM

Counter 0 - Holds the score
 126 x 1 } Count the number of turns since the beginning
 127 x 256 } of the game.

Marker 0 - If set means that a room has been described since
 last reset.
 1 - If set, you are in a light room.
 2 - If set, you have a source of light.
 3 - If set, disables the scoring mechanism.

v. RANGES OF NUMBERS

	No. allowed	No. of characters
Rooms	1..9999	255
Objects	1..255	255
Messages	1..255	255
Verbs	1..255	35
Nouns	1..255	35
Adverbs	1..255	35
Markers	0..255	
Counters	0..127	
[Which store]	0..255	



The internal commands QUIT and EXIT return the player to the first room with the program initialised. There is no end as such.