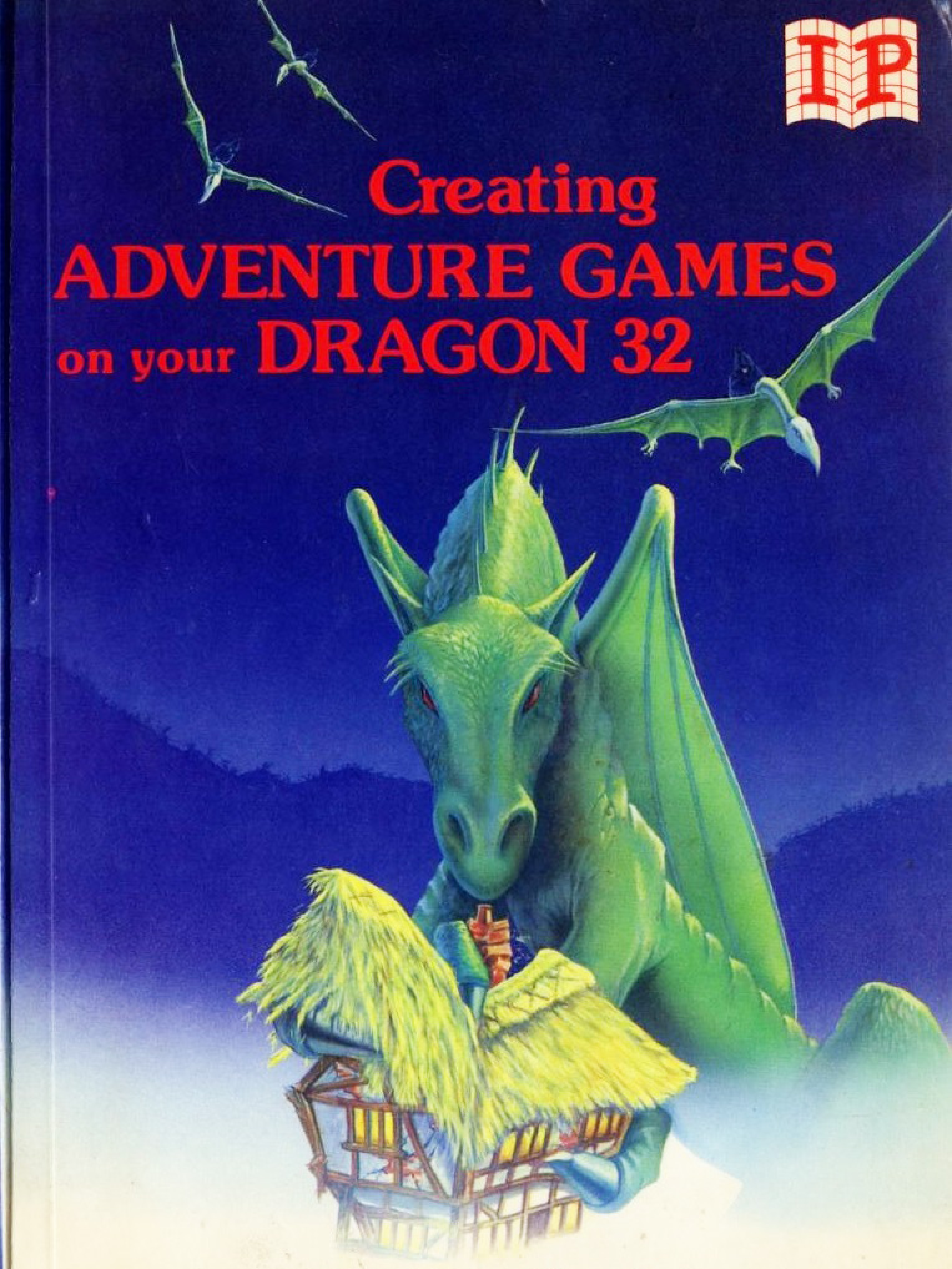


Creating ADVENTURE GAMES on your DRAGON 32

Clive Gifford

CREATING ADVENTURE GAMES ON YOUR DRAGON 32

BY CLIVE GIFFORD

INTERFACE
PUBLICATIONS

To Sally and Peter, who have helped me through many a crisis.

First published Interface Publications, 1984

First printing February 1984

The programs presented in this book have been included for their instructional value. They have been tested with care but they are not guaranteed for any particular purpose. While every care has been taken the publishers cannot be held responsible for running mistakes that may occur.

Any enquiries regarding the contents of this book should be directed by mail to Interface Publications, 9/11 Kensington High Street, London W8 5NP

ISBN 0 907563 57 0

Cover illustration: David John Rowe

Typeset in England by Twickenham Printers, Albion Works, Albion Road, Twickenham, Middlesex.

Printed and bound by Short Run Press Ltd., Exeter

CONTENTS

The author wishes to thank Scott Vincent and Ian Watt for their contributions, Richard for his constant advice and friendship, and 'Mum' for the many hours of frustration playing the adventures.

INTRODUCTION

There can be no doubt that playingadventuregames on home computers is increasing in popularity all the time. After all, what better way to relax than to forget the 'real' world and plunge headlong into a world of make-believe. Only an adventure game can take you to the heady heights of Mount Everest, pit you against Elfin Gods in the bowels of the earth or transport you into a darkened maze of terrifying complexity - all at a touch of a key.

To many people, the real enjoyment of adventure programs is in trying to win the game. They are a challenge to play - to find all the objects needed and reach your goal is a task often requiring some thought. Certainly, adventures are more stimulating and fulfilling than zapping aliens all the time.

For many adventure-gamers, there comes a time when they want to write their own adventure. "Where do I start?" is an often-heard question. For when you sit down and actually try to write a good adventure program, you realise how complex the actual task is.

Primarily, this book is a collection of programs for you to type in and enjoy and/or use in your own adventure games. There are some five adventures of one type or another in this book, as well as a whole host of adventure utilities and four non-adventure games.

The whole aim of this project is to get you, the reader, up and writing your own adventure games as quickly as possible.

There is little more to say, save that reading through this book in its entirety will provide you with maximum understanding. Finally, good luck and happy adventuring!

Clive Gifford,
Ashford, Middlesex.
October 1983.

CHAPTER 1

AN INTRODUCTION TO ADVENTURES AND ADVENTURE PROGRAMMING

Adventures and adventure programming is a huge area of computer entertainment that is only now being exploited to its full capabilities by the commercial software market. The ability to be locked in mental combat with unreal obstructions and fantastic creatures in a mythical world where anything can happen is a pastime that thousands of computer owners and users are flocking to.

Adventure games can offer a *real* challenge to those who want more from computer gaming than blasting spaceships out of existence. You will find that the range of adventures commercially available for your computer is very comprehensive and will contain something to interest and amuse everyone.

From a programming point of view, adventures are ideal projects for the less-experienced programmer. After fiddling around with a few very simple programs, you tend to find that you would like to embark on a bigger project. Adventures provide an excellent option. There is no need to complicate matters with graphics, speed (as for action games) is not essential. Most areas of BASIC programming you'll need for such a project (such as array manipulation and string handling) are fairly easy to master, and the program can be made as personal as the writer requires.

If anyone can claim to be inventors of the adventure game as we know it then it must be two mainframe programmers from the United States, Crowther and Woods. Their amusing adventure was set in what today is termed as the 'classic mould' with trolls, dragons and serpents to defeat in order to capture the treasure. This adventure became a cult with many computer users of the time, particularly on University campuses and in big busineses where there was access to large computer systems. Once microcomputers with sufficiently large memories came along, the adventure programs were scaled down to fit them and so the story continued.

If you ask many people what they recognise as a 'computer adventure', they will invariably answer that it is a game with monsters, trolls, wizards and treasure. You must pick up various objects to do battle with the beasts and obtain as much treasure as possible to take back to your home base. This is the 'classic' type of game following Woods's and Crowther's model but there are, of course, a much wider range of adventure games with no mythical beasts, no treasure to collect and no objects to even contemplate picking up. Perhaps the beauty of adventure writing is that almost any scenario can be used and any method of play adopted to suit the programmer's whim.

What then distinguishes adventure games from any other form of computer game? I considered this matter and detailed what I believe to be the basic points that a game must include to become an adventure. An adventure program must have:

(a) locations (ie. rooms).
(b) usually some form of logical connection between rooms (though the logic is sometimes far from perfect!).
(c) some form of reward or aim to work for, eg. escape, or the accumulation of treasure. Without any objective, an adventure game might become pointless and boring.
(d) some form of obstacle to hinder your progress, eg. monsters, locked doors, etc.
(e) some element of discovery.
(f) some form of command with which adventurers can direct themselves around the various locations.

These different factors would all seem essential if you were to create a playable and entertaining adventure game.

How does one go about writing an adventure then? Well, the chapters following on from this one will attempt to answer this with some practical examples, but let's consider a little theory first.

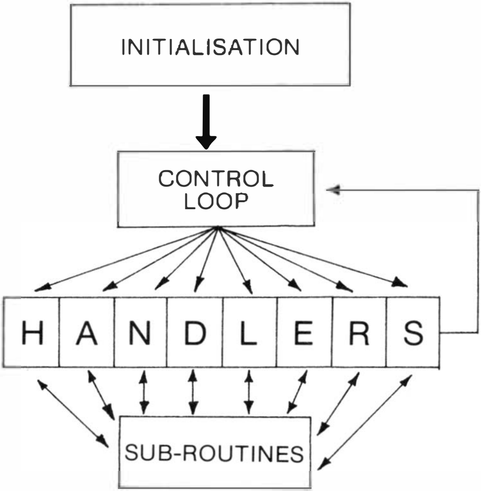
The most important factor that you must take into consideration is structured programming. Before you turn the page in horror, it must be said that structuring your programs is not that difficult. Everyone to an extent already structures their programs and it only takes a little extra thought and planning beforehand to give your programs enough structure to increase their performance and save you, in the long run, time.

Why bother structuring programs at all? There are many good reasons to do so. BASIC is quite an unstructured language when compared to something like FORTH and so the programmer must add his or her own framework. Structuring a program sensibly speeds the program up without it having to look through masses of irrelevant code every cycle. Structuring a program allows room for expanding the program at a later date and saves memory space.

For adventure programmers, structuring is essential. If you have written a program without structure, returning to it after a gap of several weeks will be difficult. Looking at the program listing it will be hard to tell where the 'room display' routine was or where the score increment is situated. For budding adventure writers, knowing where each routine and every piece of code is situated is vital, often simply because adventure programs are larger and more complex than other pieces of software. Without structuring, a large adventure will be a mess and future attempts at modification may well prove futile.

I have laboured this point a little, but I think if there is a secret to writing quality adventures of any length for any machine then it must be the use of structured programming.

Adventure programs need to be split into areas so that the programmer can concentrate on one area at a time. The diagram below shows one possible way.



Initialisation simply sets all the variables, arrays and other parameters to their starting values. The control loop is the vital part; the program constantly cycles through this awaiting a player's input and then directing the program flow to one of the many handlers. The handlers cover all the varying actions and tasks that will arise in the adventure, such as movement, a battle, finding and taking items and the like. These handlers make use of a clutch of subroutines, such as message display and pause loop. The handlers, once they have finished their task, go back to the control loop and so the process starts all over again.

It should be noted that this is just one method of structuring your adventure, there are lots of other ways. For instance, instead of having a bank of subroutines you could incorporate them into the handlers.

With that discussion of structured programming under our belts, we can now turn to planning out our adventure program. It must be said that planning out the game on paper is almost essential to good adventure writing. You must first try and decide the specifications of your adventure, ie. how many locations, the number of commands, types of obstacle, scenario setting and so on.

Once you have written all the parameters down, then you can start to work out which operation goes where. Having a few pages of information is very useful to turn to when you are in the process of completing your adventure. If you find a snag in the program, you do not really want to look through 16K of code for the elusive bug - it would be much simpler to check your papers for any possible problems. In later chapters, you will learn how to construct travel tables, a map of your adventure, an object table and all the other necessary information blocks to make your adventure work, and work smoothly.

CHAPTER 2

A SAMPLE ADVENTURE THE DARK FOREST

Now to our first adventure, a medieval-style scenario with traditional 'baddies' to overcome. The Dark Forest places you in the centre of a huge, foreboding forest around the time of King Arthur. You have been sent by the King to search for and destroy the creature, a strange beast that has never been seen by the local villagers and yet rules the county of Ramshire by terror. Once at the forest, your adventure has begun - you must find enough treasure and magic influence (while retaining plenty of strength) so that when you eventually meet with this loathsome brute, you can finish him off once and for all. The pathway to your goal is not an easy one as many fierce animals and the infamous Black Knight live in this area, and they will no doubt do their best to overcome you.

I do not want to give any of the secrets of the program away as without them, the adventure would lose much of its interest. This adventure is not of the 'sentence input' type where the computer must accept and decode a phrase or sentence command from the adventurer. You move around the forest by entering the compass heading, N, S, E or W. Occasionally, you will be asked questions; usually these are answered with a 'Y' or a 'N', but you will find the program is very user-friendly so you should not have any problems in this department.

One other command at your disposal is the 'rest' command. By typing 'R', you can choose a number of hours for your character to relax and increase his or her strength. However, each hour's peace costs a certain amount of cash, so you must also take this into consideration.

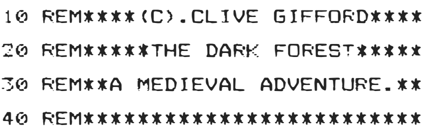
There will come a time when you cross paths with a fearsome monster of some ugly kind or another. You have the options of either fighting or running. If you choose fighting, you then have to choose an attribute to battle with. Certain monsters are more powerful than others. I will not give too many details of each monster's abilities away but suffice to say that if you come across the Black Knight, run as fast as you can!

This adventure uses many random elements in contrast to the adventure game which follows in the next chapter – the Nielson Papers. The locations and the logic linking them up, however, is always the same. This adventure could be a little more structured than it is but as it stands, the game runs smoothly. I have included REM statements indicating where all the routines lie, and from these you should easily be able to glean all that you need to know about the programming from the listing.

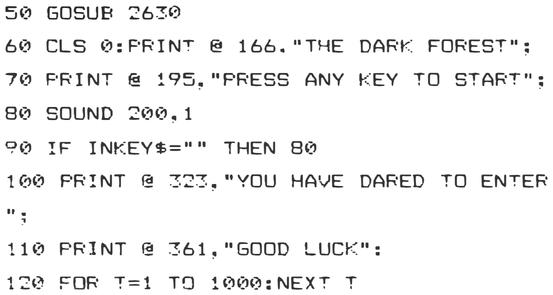
At the back of this chapter, there is a complete ready-to-run listing of the game. The rest of this chapter breaks the game down into various sections and analyses them in detail. Without structuring, it would have been impossible to split the program up into neat blocks at all. The program sections have been given the following labels: Initialisation; Control Loop; Monsters and Fight; Attributes and Possessions; Locations; The Final Confrontation; and finally, Win or Lose. This is the order that the sections of program will be looked at.

INITIALISATION

Initialisation in this program is in two blocks: the 'initialisation' subroutine and the 'start game' sequence. The 'start game' sequence is shown below.



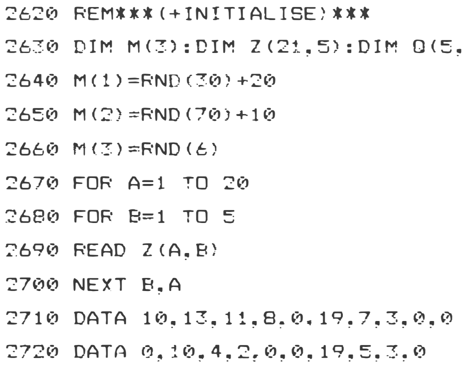
```
10 REM****(C).CLIVE GIFFORD****
20 REM*****THE DARK FOREST*****
30 REM**A MEDIEVAL ADVENTURE.**
40 REM*************************
```



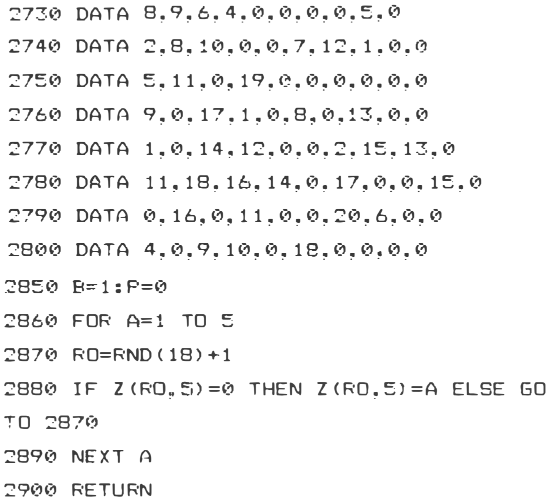
```
50 GOSUB 263Ø
6Ø CLS Ø:PRINT @ 166,"THE DARK FOREST";
7Ø PRINT @ 195,"PRESS ANY KEY TO START";
8Ø SOUND 2ØØ,1
9Ø IF INKEY$="" THEN 8Ø
1ØØ PRINT @ 323,"YOU HAVE DARED TO ENTER
";
11Ø PRINT @ 361,"GOOD LUCK":
12Ø FOR T=1 TO 1ØØØ:NEXT T
```

The first genuine program line, line 5Ø, gos to the 'initialisation' subroutine situated at line 263Ø onwards. The next few lines clear the screen, set up the title page and wait for a key to be pressed. Once a key has been pressed (the INKEY$ value is something other than "") the game displays the 'good luck' message and flows into the 'control loop' section of the program.

The 'initialisation' loop is shown below.



```
262Ø REM***(+INITIALISE)***
263Ø DIM M(3):DIM Z(21,5):DIM O(5,5)
264Ø M(1)=RND(3Ø)+2Ø
265Ø M(2)=RND(7Ø)+1Ø
266Ø M(3)=RND(6)
267Ø FOR A=1 TO 2Ø
268Ø FOR B=1 TO 5
269Ø READ Z(A,B)
27ØØ NEXT B,A
271Ø DATA 1Ø,13,11,8,Ø,19,7,3,Ø,Ø
272Ø DATA Ø,1Ø,4,2,Ø,Ø,19,5,3,Ø
```



```
2730 DATA 8,9,6,4,0,0,0,0,5,0
2740 DATA 2,8,10,0,0,7,12,1,0,0
2750 DATA 5,11,0,19,0,0,0,0,0,0
2760 DATA 9,0,17,1,0,8,0,13,0,0
2770 DATA 1,0,14,12,0,0,2,15,13,0
2780 DATA 11,18,16,14,0,17,0,0,15,0
2790 DATA 0,16,0,11,0,0,20,6,0,0
2800 DATA 4,0,9,10,0,18,0,0,0,0
2850 B=1:P=0
2860 FOR A=1 TO 5
2870 RO=RND(18)+1
2880 IF Z(RO,5)=0 THEN Z(RO,5)=A ELSE GO
TO 2870
2890 NEXT A
2900 RETURN
```

Line 263Ø dimensions all the well-used arrays. M is the attribute array and Z is a 'two-dimensional room' array. The attributes, strength, money and magic spells, are fixed at a value by lines 264Ø, 265Ø and 266Ø. The section of code between lines 267Ø and 28ØØ reads the values of the room data into the room array, Z. There are 21 rooms, each room having five numbers assigned to it. The actual workings of this location logic system is shown in the locations section of this chapter.

B, the variable holding the player's location number, is set to one, and the variable used for the special idol, P, is set to zero.

Finally in this section, the starting positions of the five monsters is determined between lines 286Ø and 289Ø. The fifth number in the room array data is set to a value other than zero if one of the five monsters is present in that room. The program then returns.

CONTROL LOOP

The most often used part of the program, the control loop, is the vital component of this game. The functions that it performs are quite wide-ranging and are detailed below.

The first two lines perform the presentation 'niceties'; that is clearing the screen (in one of the nine colours), pausing the program, playing a quick note or two of music and printing up the attribute values (the GOSUB 73Ø handles that).

```
130 FOR X=1 TO 1500:NEXT X:CLS RND(9)-1
140 SOUND 180,1:GOSUB 730:PRINT
```

The player's response is handled by line 28Ø which then sends the program back to the beginning of the control loop.

```
280 PRINT:INPUT"WHAT NOW";N$:N$=LEFT$(N$
,1)
290 GOTO 130
```

The player's input is cut down to the first letter and then acted upon by the control loop. Lines 15Ø to 18Ø process the player's movement.

```
150 IF N$="N" AND Z(B,1)<>0 THEN B=Z(B,1)


160 IF N$="S" AND Z(B,2)<>0 THEN B=Z(B,2 )


170 IF N$="E" AND Z(B,3)<>0 THEN B=Z(B,3)


180 IF N$="W" AND Z(B,4)<>0 THEN B=Z(B,4)
```

If the player's input was not 'R' (for rest), the program jumps to line 230. The lines between 190 and 230 handle the 'rest' command. A price for each hour's uninterrupted sleep is displayed and an input waited for. There then follows a pause in the game that is roughly proportional to the number of hours of rest chosen. Obviously, the money is deducted from the array, and the amount left is displayed together with the new strength level.

```
190 IF N$<>"R" THEN 230
200 CS=INT((B+10)/10):PRINT"  O.K. YOU C
AN SLEEP HERE BUT ITWILL COST YOU":CS;"C
OIN/S FOR EVERY HOURS UNINTERRUPTED REST
.":PRINT
210 INPUT"HOW MANY HOURS";H:IF H<1 OR M(
2)-(H*CS)<0 THEN PRINT"TRY AGAIN, SLEEPY
HEAD!":GOTO 210
220 M(2)=M(2)-(H*CS):M(1)=M(1)+H:CLS:PRI
NT @ 236,"ZZZZ":FOR T=1 TO 1500*H:NEXT T
:CLS:GOSUB 730:GOTO 250
```

The final lines in the control loop are particularly concerned with locations. Line 250 directs the program to the correct 'location' subroutine. Line 260 checks to see if there is a monster in the same room as you; if there is, the computer jumps out of the control loop and into the 'monster/fight' routine. Lines 230 and 240 provide a check to see if you have made a valid move since your previous go. If CHK equals B after going through the movement routine, it means that the player has not been able to move even though requested to do so. Therefore, the player attempted to make an invalid move and is told so.

```
230 IF CHK=B THEN PRINT"cannot move in t
hat direction":PLAY"L1O1C":GOTO 260
```

```
240 CHK=B
250 ON B GOSUB 980,1000,1020,1170,1200,1
250,1270,1290,1310,1380,1520,1630,1660,1
680,1700,1760,1880,1910,1930,1950
260 IF Z(B,5)<>0 THEN 300
270 GOSUB 790
```

MONSTERS AND FIGHT

Once the program has detected that there is a monster in the same location as the adventurer, the program moves to the 'monster/fight' routine. Lines 300 - 360 check to see which monster is in the location and prints a short description of that monster. The monster's fight level is given by the variable E.

```
300 REM***MONSTER/FIGHT***
310 MN=Z(B,5):PRINT
320 IF MN=1 THEN PRINT"A BAND OF 10 ROBB
ERS SEIZE YOU.":E=40
330 IF MN=2 THEN PRINT"YOU HAVE COME ACR
OSS GRENDAL, THE HERMIT.YOU HAVE AWOKEN
HIM HE IS NOT PLEASED.":E=40+RND(15)340
IF MN=3 THEN PRINT"SIR RUFUS OF GASCONY
ATTACKS...":E=20
350 IF MN=4 THEN PRINT"A FEROCIOUS
BEAR,RAVENOUS FOR MEAT POUNCES. ": E=2()1
36(;1 1F )1N=5 THEN PRINT" YOU HEAR THE WAR
CRY OF THE FEARLESS BLACK KNIGHT. ": E=60
```

The program waits for the player's decision to fight or run. If the player chooses to run and the random element in line 390 is favourable, then the player is asked for which direction and the answer is handled in the same way as any normal direction command. Four points are taken from the player's strength level for this base cowardice.

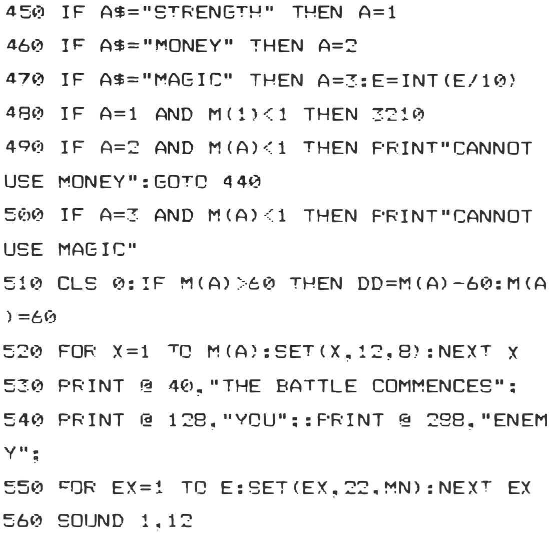
If the random element is not so kind, the player loses ten strength points, *and* must stay and fight the enemy!

```
370 INPUT"WILL YOU FIGHT OR RUN":F$:F$=L
EFT$(F$,1)
380 PRINT:IF F$="F" THEN 41Ø
390 IF RND(1Ø)>4 THEN INPUT"WHICH DIRECT
ION COWARD":N$:M(1)=M(1)-4:PRINT:GOTO 13Ø
```

```
46Ø PRINT "NO YOU MU$T $TAND & FIGHT A$ YOU LO$E 1Ø
$TRENGTH":MN=MN-1Ø
```

The fight now begins with the computer accepting the name of the attribute that the player wishes to fight with. Several lines are taken up with checks to see that the player can use that attribute. The fight display is now constructed. A simple bar graph, for want of a better name, is displayed on-screen with a bar for the player's attribute and a bar for the monster's attribute. The size of the bars is in proportion to the actual level of the respective attribute. Line 51Ø handles any attribute that is larger than 6Ø.

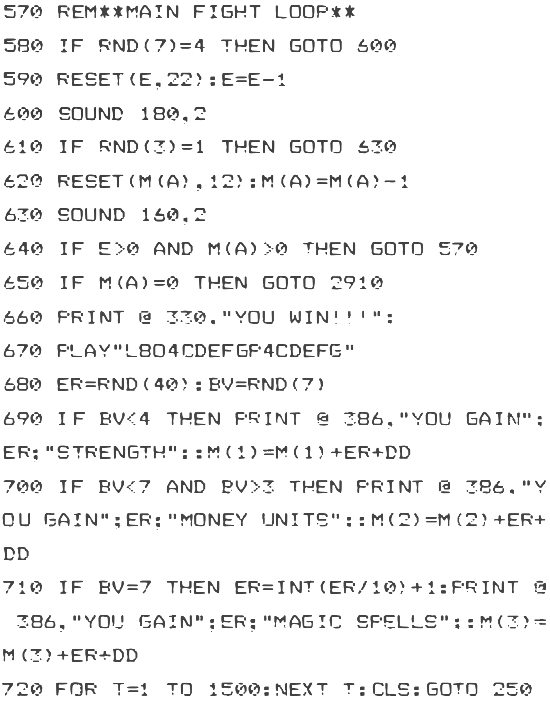
```
41Ø REM**FIGHT**
42Ø Z(B,5)=Ø:RO=RND(19)
43Ø IF Z(RO,5)=Ø THEN Z(RO,5)=MN ELSE GO
TO 42Ø
44Ø INPUT"WHICH ATTRIBUTE DO YOU WISH
TO FIGHT WITH":A$
```



```
450 IF A$="STRENGTH" THEN A=1
460 IF A$="MONEY" THEN A=2
470 IF A$="MAGIC" THEN A=3:E=INT(E/10)
480 IF A=1 AND M(1)<1 THEN 3210
490 IF A=2 AND M(A)<1 THEN PRINT"CANNOT
USE MONEY":GOTO 440
500 IF A=3 AND M(A)<1 THEN PRINT"CANNOT
USE MAGIC"
510 CLS 0:IF M(A)>60 THEN DD=M(A)-60:M(A
)=60
520 FOR X=1 TO M(A):SET(X,12,8):NEXT X
530 PRINT @ 40,"THE BATTLE COMMENCES";
540 PRINT @ 128,"YOU";:PRINT @ 298,"ENEM
Y";
550 FOR EX=1 TO E:SET(EX,22,MN):NEXT EX
560 SOUND 1,12
```

The fight continues with points being docked off until one of the attributes reaches zero. If the player has won then he or she will be granted some extra points for one of his or her attributes; if the player loses the battle, of course, the game is over. The bar chart is updated every cycle of the loop, with the bars getting gradually smaller.

One point of interest with this part of the program is how the fight is weighted in favour of the player. If you notice line 610, the computer is given a one in three chance of jumping the line which deducts one from the player's score. The monster, in contrast, has only a one in seven chance of the same happening. Therefore, if the adventurer starts a fight with four or five points less than the monster, he or she will not be resigned to defeat but could well have a good chance of overcoming the monster's advantage. Little weightings here and there in the program may decrease the logical and 'certain' elements in the program, but if used correctly will enhance the actual game.



```
570 REM**MAIN FIGHT LOOP**
580 IF RND(7)=4 THEN GOTO 600
590 RESET(E,22):E=E-1
600 SOUND 180,2
610 IF RND(3)=1 THEN GOTO 630
620 RESET(M(A),12):M(A)=M(A)-1
630 SOUND 160,2
640 IF E>0 AND M(A)>0 THEN GOTO 570
650 IF M(A)=0 THEN GOTO 2910
660 PRINT @ 330,"YOU WIN!!!":
670 PLAY"L8O4CDEFGP4CDEFG"
680 ER=RND(40):BV=RND(7)
690 IF BV<4 THEN PRINT @ 386,"YOU GAIN";
ER;"STRENGTH"::M(1)=M(1)+ER+DD
700 IF BV<7 AND BV>3 THEN PRINT @ 386,"Y
OU GAIN";ER;"MONEY UNITS"::M(2)=M(2)+ER+
DD
710 IF BV=7 THEN ER=INT(ER/10)+1:PRINT @
 386,"YOU GAIN";ER;"MAGIC SPELLS"::M(3)=
M(3)+ER+DD
720 FOR T=1 TO 1500:NEXT T:CLS:GOTO 250
```

ATTRIBUTES AND POSSESSIONS

Lines 730 to 780 encompass the 'attributes' routine. They have been defined in the 'initialisation' section and are added to and taken away from in the rest of the adventure. These attributes take the place of 'possessable' objects which are often found in other adventures. A quick glance at the attributes will give you an idea of how well you are doing and which areas, if any, you are weak on.

The routine simply takes one away from the strength attribute and then displays the attributes. If the strength is equal to zero then the program jumps to the 'lose' routine situated at 2910 onwards.

```
730 REM**ATTRIBUTES**
740 M(1)=M(1)-1:IF M(1)=0 THEN 2910
750 PRINT"STRENGTH":INT(M(1))
760 PRINT"MONEY";M(2)
770 PRINT"MAGIC";M(3)
780 RETURN
```

The 'possessions' routine is a little more involved. Line 790 decides if an object is going to appear and if so, chooses one of the five available objects. The lines from 810 to 850 describe the particular object chosen.

```
790 IF RND(10)>8 AND B<>1 AND B<>15 AND
B<>20 THEN PS=RND(5):GOTO 810
800 RETURN
810 IF PS=1 THEN PRINT"  AN OLD GLASS BO
TTLE IS WITHIN REACH. IT IS COVERED IN C
OBWEBS"
820 IF PS=2 THEN PRINT"  HERE IS A LARGE
OAKEN CHEST. A WEIRD SIGN IS CARVED ON ITS
LID."
830 IF PS=3 THEN PRINT" A VELLUM PARCHMENT
BODI(LIES BY YOUR FEET."
84Ø IF PS=4 THEN PRINT"  A GOLD LOCKET O
N A CHAIN IS   CLOSE TO YOU."
85Ø IF PS=5 THEN PRINT"  BEFORE YOU IS A
 BOX OF EXOTIC FRUIT"
86Ø REM****POSSESSIONS****
```

The next two program lines accept the player's decision to open the particular object or not. If you're brave and decide to open the object, then there is a three in ten chance of an Orc jumping out and severely mauling you. This chance element is handled by line 89Ø.

The rest of the routine is concerned with printing out and altering the attributes of the player according to which object has been safely opened. The descriptions are quite detailed and add a little extra realism to the game. As opposed to having a large pause loop, I chose to let the player press a key to continue. This is handled by lines 95Ø and 96Ø.

```
87Ø INPUT"DO YOU WANT TO OPEN IT";OP$
88Ø PRINT:IF LEFT$(OP$,1)<>"Y" THEN RETU
RN
89Ø IF RND(1Ø)>7 THEN PRINT"AN ORC,THE S
ERVANTS OF THE BLACKENID, JUMPS OUT AND
ATTACKS YOU.IT DISAPPEARS AFTER A STRUGG
LE  LEAVING YOU WEAK AND TIRED.":M(1)=M(
1)-(M(1)/4):PS=Ø
9ØØ IF PS=1 THEN PRINT"A LARGE CLOUD OF
SMOKE APPEARS, IT'S SULPHUROUS ESSENCE G
IVES   YOU NEW STRENGTH AND LIFE":M(1)=M
(1)+8
91Ø IF PS=2 THEN PRINT"OH JOY OF JOYS! M
```

```
UCH TREASURE   IS CONTAINED IN THE CHEST
. YOU  ARE A RICH MAN":M(2)=M(2)+35+RND(
15)
920 IF PS=3 THEN PRINT"IT CONTAINS 6 POW
ERFUL SPELLS.":M(3)=M(3)+6
930 IF PS=4 THEN PRINT" IT HAS A SMALL P
ORTRAIT OF THE BEAUTIFUL PRINCESS CAROLI
NE DE  MOSELLE. IT IS QUITE VALUABLE.":M
(2)=M(2)+10
940 IF PS=5 THEN PRINT"YOU ARE SO TEMPTE
D THAT YOU EAT SEVERAL. THEY ARE DELICIO
US AND DO NO HARM BUT ONLY ADD TO YOUR S
TRENGTH.":M(1)=M(1)+12
950 PRINT:PRINT"****PRESS A KEY TO CONTI
NUE****"
960 IF INKEY$="" THEN 960 ELSE CLS:RETUR
N
```

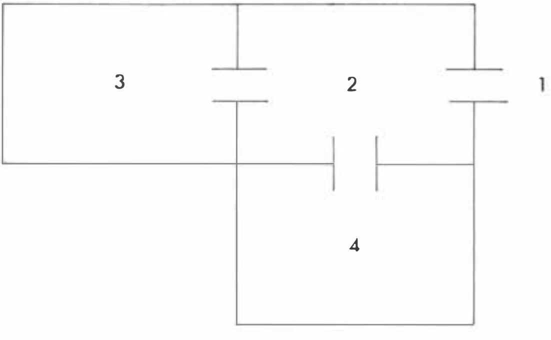
LOCATIONS

The most important part of any adventure is to include a number of different locations, for without locations to move between there would be no exploration and little discovery (two elements that are vital to an entertaining adventure).

More difficult than thinking up location descriptions, is formulating some system of logic between each location and its exits. The adventures in this book show you several particularly versatile ways of constructing room logic. 'The Dark Forest' uses one of these systems.

If you look through the 'initialisation' section, you will notice a large amount of data that is being read into an array, Z. There are five numbers for each location. Let's examine the third set of numbers, ie. the date for location number '3'. The data is as follows: Ø, 1Ø, 4, 2, Ø. The last number is used later for indicating whether a monster is present in the location or not, so at this stage in the program every fifth number is a zero. The first four numbers relate to location numbers. The first piece of data gives the number of the location if the player goes north from location '3'. In the location we are at present discussing, location '3', the player's way in that direction is blocked, ie. there is no exit, hence the first location number is zero. If the player wishes to move south, then the second piece of data gives the location he or she would end up in, location '1Ø'. The third item in the data list gives the location if the player goes east and the final piece is for when the player goes west. If you look at lines 15Ø to 18Ø in the control loop, you will see how this information is interpreted when the player wishes to move. Line 15Ø simply means 'If the player requests to go north and the first item of data in the present location is not equal to zero (is not a blocked exit) then the first piece of data in the player's present location will now become the number of the player's new location. The player's location is always stored in the variable, B'.

If you find this a little involved, then consider the simple example below. As you can see, there are three rooms and an 'outside' (labelled location '1').



The data for the four locations would be as follows:

Location '1'	Ø, Ø, Ø, 2
Location '2'	Ø, 4, 1, 3
Location '3'	Ø, Ø, 2, Ø
Location '4'	2, Ø, Ø, Ø

Location '1' has a single exit to the west, location '2' has exits to the south, east and west, location '3' has an exit to the east and location '4' has an exit to the north.

From this it can be seen how a full logic system can be developed. To help you, it is a good idea to remember the locations as simple rooms with open doors in certain directions. When designing your adventure, try to map out your locations on a piece of paper. You will then find it easier to code into a data list and you will also have at your disposal, a complete map of your adventure which you will find useful when testing the various parts of the adventure or when a friend is playing the game and comes across a troublesome problem.

Using this system, many extra options can be attached. Up and down movement could be allowed with two extra items of data for each location needed. The extra data would simply signify which location the player would end up in if he or she went up, in one case, or down.

Using a similar feature to the monster feature in this game, it would be possible to assign an extra figure that places the objects in fixed positions at the start of the game so that each item would only appear in one location. The actual possibilities with this logic system are very wide indeed.

Once the player has moved to a new location, the computer ON. .GOSUBs to the line number dealing with that location. In this adventure, the computer does not just print the name of the location but if necessary, sets up a high resolution screen picture or waits for a specific piece of action that can only happen in that particular location. This is a good format for the first time adventure programmer to adopt as it allows plenty of flexibility. Also, if you wish to add an extra feature at some time, then you will have little of the program to change to do so.

Below is the section of the program which covers the detail of each location. Most are self-explanatory, so it is only necessary for me to give details if the location is particularly complex.

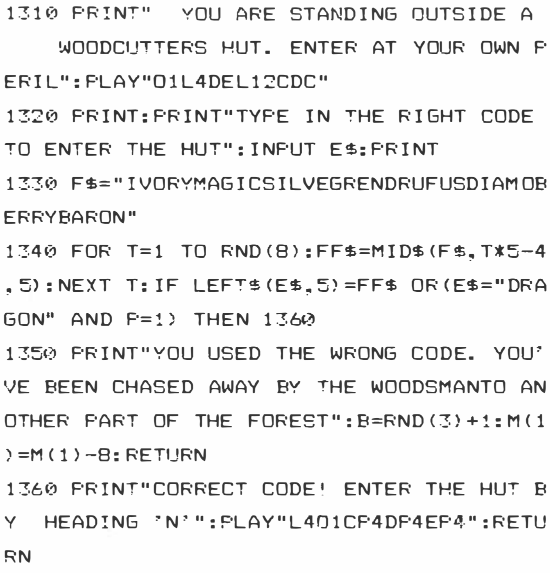
```
970 REM**LOCATIONS**
980 PRINT"     YOU ARE IN A SMALL CLEARIN
G WITHIN THE DARK FOREST. YOU CAN PROCEE
D IN ANY DIRECTION THAT   YOU WISH"
990 RETURN
1000 PRINT"  YOUR VIEW FROM OAKRIDGE HIL
L  IS MAGNIFICENT. IN THE DISTANCE YOU C
AN SEE GRENDAL THE          HERMIT'S LODG
E"
1010 RETURN
```

This location was really an afterthought that I could not resist adding. The memento at the bottom of the well can come in extremely useful. This part of the adventure involving the code on the memento is the hardest piece for the adventurer to solve. The graphics are quite amusing as well!

```
1020 PRINT"  YOU HAVE FOUND YOURSELF IN
A    PEACEFUL, LUSH FIELD."
1030 PRINT:PRINT"THERE IS A SMALL EMPTY
WELL NEXTTO YOU. DO YOU WANT TO GO DOWN
 IT.(Y/N)":INPUT Q$
1040 IF Q$="N" THEN PRINT"  ALRIGHT, STA
Y IN THE PEACEFUL FIELD THEN.":RETURN
1050 IF RND(3)<2 THEN 1120
```

```
1060 PMODE 3,1:PCLS:SCREEN 1,0:CIRCLE(12
8,40),20
1070 CIRCLE(119,32),2:CIRCLE(137,32),2:C
IRCLE(128,40),3:CIRCLE(128,55),9,4,1,.6,
.9
1080 COLOR 2,2:LINE(0,0)-(40,192),PSET,B
F
1090 LINE(215,0)-(255,192),PSET,BF:PAINT
(45,190),3,2
1100 FOR T=1 TO 1000:CLS:PRINT:PRINT" AH
,WELL,I THOUGHT IT WAS EMPTY":PRINT"HOW
SILLY OF ME!":M(1)=10:M(2)=10:M(3)=1:B=1
1110 PRINT:PRINT"  YOU DO MANAGE TO GET
OUT, BUT IN DOING SO LOSE MUCH MONEY AND
 STRENGTH.":GOTO 1150
1120 PLAY"L18O4GDFBBACEBB":PRINT:PRINT"A
 SMALL NICHE IN THE WELL WALL  HIDES A G
OLDEN FIGURE,THE RAM OFANAG."
1130 PRINT"   INSCRIBED ON THE UNDERSIDE
 IS THE MYTHICAL WORD grandos. THE FIGUR
E IS QUITE VALUABLE, LOOK  AFTER IT."
1140 P=1:M(3)=M(3)+1:M(2)=M(2)+5
1150 PRINT:PRINT"****PRESS A KEY TO CONT
INUE****"
1160 IF INKEY$="" THEN 1160 ELSE GOSUB 7
30:RETURN
1170 PRINT"  THE BARON'S RESIDENCE IS
   NEARBY. YOU CAN HEAR BEAUTIFUL  MUSIC
 BEING PLAYED...."
1180 PLAY"O3L8EL6GL8AL7BL12O4C#L8O3BL6AL
8F#L7DL12EL8F#"
1190 RETURN
1200 PRINT"  YOU ARE IN A DAMP HUT WITH
A  POORLY THATCHED ROOF. A MUSTY   SMELL
 LINGERS AND THERE IS AN   ADJOINING ROO
M TO THE RIGHT."
1210 FOR T=1 TO 2000:NEXT T
1220 PLAY"L8O4CO2CO4CO2CO4CO2C"
1230 PRINT"WHATS THIS? A SMALL HIDDEN DO
OR IN FRONT OF YOU (N)."
1240 RETURN
1250 PRINT"AROUND YOU IS A SPARSELY FILL
ED STOREROOM. THIS IS A DEAD-END."
1260 RETURN
1270 PRINT"  YOU STAND IN THE MIDDLE OF
   BERRY HAMLET. IT IS VERY QUIET. THE P
LACE SEEMS deserted"
1280 RETURN
1290 PRINT"  YOU ARE ON A GRASSY HILL
   SLOPING DOWNWARDS INTO A VALLEY. A SM
ALL HAMLET LIES AT THE      NORTHERN END.
"
1300 RETURN
```

The following location, the woodcutter's hut, waits for you to enter a code word. It takes the first five letters of whatever you enter and compares it with a string holding a number of keywords. If the random number is in your favour and you have entered one of the code words correctly, then you will be allowed to enter the hut. A much more sure way to enter the hut is to find the special treasure which will give you the code word which works every time. Each time you play the adventure, you will have to collect the treasure for the code word to work. If you get the word wrong or you are not carrying the treasure, you are chased away.



```
131Ø PRINT"  YOU ARE STANDING OUTSIDE A
     WOODCUTTERS HUT. ENTER AT YOUR OWN P
ERIL":PLAY"O1L4DEL12CDC"
132Ø PRINT:PRINT"TYPE IN THE RIGHT CODE
TO ENTER THE HUT":INPUT E$:PRINT
133Ø F$="IVORYMAGICSILVEGRENDRUFUSDIAMOB
ERRYBARON"
134Ø FOR T=1 TO RND(8):FF$=MID$(F$,T*5-4
,5):NEXT T:IF LEFT$(E$,5)=FF$ OR(E$="DRA
GON" AND P=1) THEN 136Ø
135Ø PRINT"YOU USED THE WRONG CODE. YOU'
VE BEEN CHASED AWAY BY THE WOODSMANTO AN
OTHER PART OF THE FOREST":B=RND(3)+1:M(1
)=M(1)-8:RETURN
136Ø PRINT"CORRECT CODE! ENTER THE HUT B
Y  HEADING 'N'":PLAY"L4O1CP4DP4EP4":RETU
RN
```

The next location, location '1Ø', is the prison which the Black Knight throws you into. The first few lines up to line 142Ø draws a Hi-res picture, and then the rest of the routine gives you some textual details of your predicament and offers you a method of escape. I will not tell you any more about how to escape – that's up to you to find out.

```
1370 N$="N":PLAY"L4O1CF4DF4EF4":RETURN
1380 PMODE 4,1:PCLS:SCREEN 1,1
1390 LINE(90,30)-(130,70),PSET,BF:LINE(9
8,30)-(104,70),PRESET,BF
1400 LINE(114,30)-(120,70),PRESET,BF
1410 LINE(0,194)-(30,174),PSET:LINE(255,
194)-(225,174),PSET
1420 DRAW"BM30,0:D174R195U174"
1430 FOR T=1 TO 2500:NEXT T
1440 CLS 0:PRINT @ 162,"YOU FELL INTO A
BEAR TRAP AND WERE CAPTURED BY THE BLACK
     KNIGHT WHO HAS IMPRISONED YOU."
1450 PLAY"V31L6O3GFEEO2GFEEO1L4GFEDCCC":PRINT
1460 INPUT"WHAT NOW";N$:IF N$="GRANDOS"
THEN PRINT"CLOSE, BUT WRONG,YOU SHOULD H
AVETHOUGHT ABOUT THE FIGURE'S NAME":PRIN
T:PRINT"THE ram OF anag....THE anag ram"
1470 IF N$="DRAGONS" AND P=1 THEN PMODE
3,1:PCLS:SCREEN 1,0:FOR T=1 TO 8:CLS T:S
OUND 25*T,4:NEXT
1480 IF N$="DRAGONS" AND P=1 THEN PRINT
@ 232,"THE MAGIC WORD!":PRINT @416,"YOU'
RE FREE AGAIN, BUT YOU LOST THE GOLDEN F
IGURE":B=1:P=0:RETURN
1490 PRINT:PRINT"you failed and are impr
isoned   forever.":END
1500 GOTO 1380
1510 RETURN
```

The location described from line 1520, is just one of a number of routines that draw a high resolution display. A chapter later in this book explains how high resolution pictures can be incorporated in your own programs.

```
1520 PMODE 3,1:PCLS:SCREEN 1,0
1530 CIRCLE(70,110),80,2,0.6,0.5,1
1540 CIRCLE(200,110),100,2,0.7,0.6,3
1550 LINE(0,110)-(255,110),PSET
1560 PAINT(180,180),2,4
1570 PAINT(0,0),3,2
1580 CIRCLE(20,20),10,2:PAINT(20,20),2,2
1590 PAINT(190,100),4,2
1600 FOR T=1 TO 2000:NEXT T:CLS:GOSUB 73
0
1610 PRINT"YOU ARE ON THE BARON'S LAND.
BE CAREFUL. IN THE DISTANCE, LIE   THE S
ACRED HILLS OF ANGELSARK"
1620 RETURN
1630 PRINT"  THE DARK FOREST SURROUNDS Y
OU. IT'S GROTESQUE FORMS SEEM TO   MOVE
CLOSER. THIS PLACE IS evil"
1640 FOR T=45 TO 1 STEP-3:SOUND T,3:NEXT
 T
1650 RETURN
1660 PRINT"YOU ARE BY THE SIDE OF A SMAL
L  STREAM. YOU CAN FOLLOW THE RIVER BANK
 BOTH EAST AND WEST OR YOU CAN RETREAT N
ORTH"
```

```
1670 RETURN
1680 PRINT"  A TWISTING TRACK CUTS THROU
GH THE TANGLED BUSHES.THIS IS WILD BOAR
COUNTRY."
1690 RETURN
1700 PRINT"  YOU ARE BESIDE A FAST-FLOWI
NG RIVER. A BRIDGE GUARDED BY A    GREED
Y TROLL LIES TO THE SOUTH. YOU MUST GUES
S THE FEE THAT THE BEAST CHARGES TO CROS
S"  "
1710 INPUT F:IF F<RND(40)+20 THEN B=Z(B,
1):PRINT"HE SCORNS YOUR TINY SUM AND
 CHASES YOU AWAY":RETURN
1720 IF M(2)-F<0 THEN PRINT"YOU GUESSED
RIGHT,BUT CANNOT PAYTHE TOLL. BE ON YOUR
 WAY, FRIEND":RETURN
1730 PRINT:PRINT"THAT IS CLOSE ENOUGH, C
ROSS NOW."
1740 M(2)=M(2)-F
1750 RETURN
```

The above piece of code deals with a vital bridge location. If the adventurer wishes to complete the adventure, he or she must find some way across this bridge. If the player offers the wrong amount of payment, he or she is chased away by the troll that guards the bridge.

```
1760 PMODE 3,1:PCLS:SCREEN 1,0
1770 DRAW"BM0,100;C2R50U70R6D5R5U5R5D25R
30U10R40D10R30U25R6D5R5U5R5D70R75L245"
```

```
1780 DRAW"BM103,100;U20R2U2R2U2R2U2R15D2
R2D2R2D2R2D20"
1790 PAINT(90,90),4,2
1800 PAINT(0,0),3,2
1810 LINE(103,100)-(88,194),PSET
1820 LINE(133,100)-(148,194),PSET
1830 PAINT(0,120),2,2:PAINT(255,120),2,2
1840 FOR T=1 TO 2000:NEXT T:CLS:GOSUB 73
0
1850 PRINT"YOU SEE THE MYTHICAL CASTLE O
F  DIAMONDS. DO NOT BELIEVE YOUR   EYES,
 WHAT YOU SEE BEFORE YOU   IS AN ILLUSIO
N CAST BY THE MOST POWERFUL OF ALL MAGIC
IANS, THE  WARLOCK OF THE DARK FOREST."
1860 PLAY"P1O1L100CDEFGGGFFEEDDCCCCCC"
1870 RETURN
1880 PRINT"BEFORE YOU LIES THE MAGNIFICE
NT WATCH TOWER OF PENDRAGON, A     SPLEN
DID GOTHIC MONUMENT."
1890 PLAY"O4L4CEDFEGL12ABABBA"
1900 RETURN
1910 PRINT" YOU ARE AT THE ENTRANCE TO A
   MYSTERIOUS CAVE. THE SUN ALWAYS SETS
TO THE RIGHT OF THE CAVE."
1920 RETURN
1930 PRINT"  FROM YOUR POSITION IN A SMA
LL MARKET SQUARE, YOU CAN SEE SOME STOCK
S AND A PILLORY A REMINDER OF WHAT WILL
```

```
HAPPEN IF YOU FAIL.""
1940 RETURN
```

THE FINAL CONFRONTATION

The adventurer has crossed the bridge and has found the way into the cave. There the player faces the Creature, the final hurdle to overcome. This part of the program puts the adventurer in a final battle where his or her attributes are tested against the Creature's.

Lines 1960 to 2180 announce the Creature's arrival and draw a picture of it on the high resolution screen, Pmode 4,1. Line 2170 plays the high warning notes within a loop, eight times.

Lines 2200 to 2240 then give the adventurer with an attribute less than the random figure, R, a chance to make an escape. If the timer is less than 250 when a key is pressed then the player is taken back to location '14', over the other side of the bridge.

```
1950 REM***END GAME***
1960 PRINT:PRINT"IT'S VERY DARK INSIDE T
HE CAVE"
1970 FOR T=1 TO 1000:NEXT T:PRINT"YOU HE
AR A LOUD NOISE"
1980 PLAY"V31L255O3BAGFEDCO2BAGFEDCO1BAG
FEDC"
1990 PRINT"IT'S TH-TH-THE C-C-C-CREATURE
."
2000 FOR T=1 TO 1000:NEXT T
2010 PMODE 4,1:PCLS:SCREEN 1,1
2020 R=RND(8)
2030 CIRCLE(130,100),40,1,0.8,0,0.5
```

```
2040 LINE(90,100)-(170,100),PSET
2050 PAINT(100,110),1,1
2060 LINE(90,115)-(170,115),PRESET
2070 FOR X=100 TO 170 STEP 10
2080 LINE(X,100)-(X,140),PRESET
2090 NEXT X
2100 CIRCLE(90,50),15,1,0.6
2110 CIRCLE(165,50),15,1,0.6
2120 CIRCLE(90,50),5:CIRCLE(165,50),5:
2130 DRAW"BM120,70;D8BR20U8"
2140 PAINT(90,50),1,1:PAINT(165,50),1,1
2150 PRESET(90,50):PRESET(165,50)
2160 FOR M=1 TO 8
2170 PLAY"V31O5L240GABGABAGABAGABAGABAAB
AAGG"
2180 NEXT M
2190 CLS 0
2200 PRINT"IF ANY OF YOUR ATTRIBUTES ARE
   LESS THAN ";R;" THEN RUN AWAY!!      (
PRESS A KEY)"
2210 TIMER=0
2220 IF INKEY$<>"" AND TIMER<250 THEN PR
INT @ 330,"YOU JUST MADE IT! YOU RUN BAC
K  OVER THE BRIDGE.":S=14:GOTO 250
2230 IF TIMER<250 THEN GOTO 2220
2240 IF M(1)<R OR M(2)<R OR M(3)<R THEN
CLS:GOTO 2910
```

The battle now begins in earnest. Firstly, the Creature's attributes are given values. Notice how two of the attributes have a very good chance of being above the adventurer's starting values. The money that the adventurer starts with has a good chance of being higher than the Creature's but it has to be remembered that the adventurer has to pay a large proportion of this amount over to the greedy troll at the bridge. Therefore, the adventurer, after several attempts at defeating the Creature, will realise that he or she has to move around the adventure a good deal more to gain the extra strength, wealth and magic required to be a reasonable opponent for the final confrontation.

```
2250 PRINT"LET THE FIGHT BEGIN....."
2260 PLAY"L4O3EFC":CLS
2270 DIM K(3)
2280 K(1)=RND(50)+10
2290 K(2)=RND(45)+15
2300 K(3)=RND(12)
```

The battle itself consists of a loop where each of the three attributes of the Creature and the adventurer are tested. The screen is cleared, the value of the attributes under battle are displayed and the battle begins . . . .

One point is deducted from each of the attributes unless a certain number is generated by the RND function. If the number three is generated by the RND function, then the program jumps a line and nothing is deducted from the player's attribute. If the number seven occurs, the same applies to the Creature's attribute. Once this is done, the computer displays a simple message. The effect obtained by the several inverse letters in each line (on the printout, these are shown as lower case letters) is fine to use in any situation, not just the one illustrated here.

This process continues until either the player's or the Creature's attribute level reaches zero. When this happens the program goes to a routine that prints up who has won, pauses the program and adds one to the 'round win' counter. If this counter reaches two then the battle has ended and the program goes to the appropriate 'end game' routine. There are two of these short 'end game' routines, one for the player and one for the Creature.

```
2310 FOR Y=1 TO 3
2320 FOR T=1 TO 1000:NEXT T
2330 CLS:PRINT @ 115,"ROUND";Y
2340 PRINT @ 258,"human";M(Y)
2350 PRINT @ 274,"creature";K(Y)
2360 IF RND(10)=3 THEN 2380
2370 M(Y)=M(Y)-1
2380 IF RND(10)=7 THEN 2400
2390 K(Y)=K(Y)-1
2400 IF M(Y)<1 THEN 2560
2410 IF K(Y)<1 THEN 2520
2420 PLAY"L240O1CFBO2EAO3DGO4C"
2430 PRINT @ 390,"batTLE STARTS"
2440 PLAY"L255O3BAGFEDC"
2450 PRINT @ 390,"BATtle STARTS"
2460 PLAY"O2BAGFEDC"
2470 PRINT @ 390,"BATTLE staRTS"
2480 PLAY"O1BAGFEDC"
2490 PRINT @ 390,"BATTLE STArts"
2500 PLAY"O1DCDC"
2510 GOTO 2340
2520 FOR T=1 TO 1000:NEXT T
2530 CLS 2:PRINT @ 100,"YOU WIN THIS ROU
ND";
```

```
2540 PLAY"L4O3FGAEE"
2550 V=V+1:IF V=2 THEN 3190 ELSE NEXT Y
2560 FOR T=1 TO 1000:NEXT T
2570 CLS 4:PRINT @ 100,"THE CREATURE WIN
S";
2580 PLAY"L4O1FGAEE"
2590 W=W+1:IF W=2 THEN 2910
2600 NEXT Y
2610 END
```

WIN OR LOSE

When it comes down to it, one of these two endings will happen to you. An ending to a game is as important as any other part of the game. If the players fail, you must give them the enthusiasm to have another try. If they win, well, they have worked hard and deserve a little more than a 'Well Done' message and the game stopping. It would be unfair to end the game in such a way that the player finds it an anti-climax. At the other extreme, you must not get so involved with a 'winning' ending (that may be only seen once every nine or ten games) that it starts to assume giant proportions and starts to take up too much valuable memory space.

I like to think that my endings have struck a balance between the two extremes.

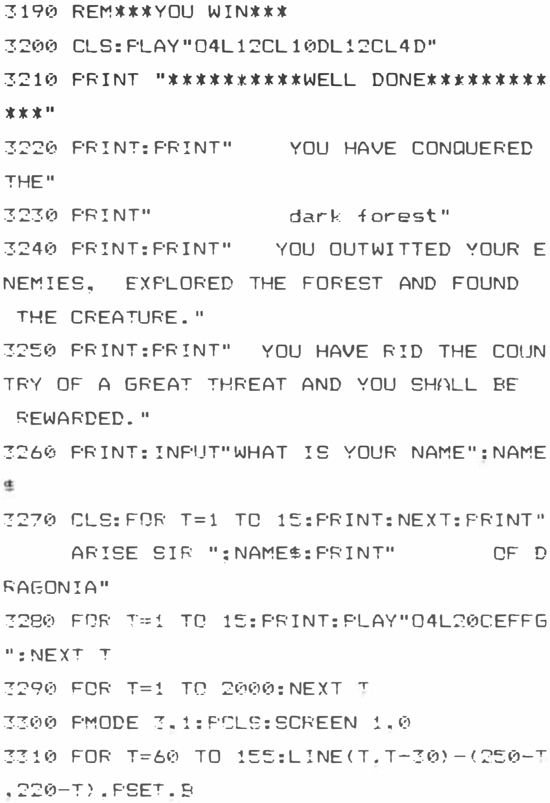
```
2910 REM***END OF GAME***
2920 FOR T=1 TO 15:PRINT" "
2930 NEXT T
2940 PRINT"    YOU HAVE BEEN BEATEN"
2950 GOSUB 3150
2960 PRINT"     YOUR STRENGTH COULD"
2970 GOSUB 3150
```

```
2980 PRINT"      NOT HOLD OUT AS YOU "
2990 GOSUB 3150
3000 PRINT"     TRIED IN VAIN TO FIND"
3010 GOSUB 3150
3020 PRINT"     THE SECRETS THAT THE"
3030 GOSUB 3150
3040 PRINT"     DARK FOREST HOLDS..."
3050 FOR C=1 TO 10
3060 GOSUB 3150
3070 NEXT C
3080 GOSUB 3150
3090 PRINT"   BUT DO NOT GIVE UP,MORTAL"
3100 GOSUB 3150
3110 PRINT"  FOR YOU MAY STILL SUCCEED."
3120 GOSUB 3150
3130 PRINT"   TO TRY AGAIN PRESS A KEY"
3140 IF INKEY$="" THEN GOTO 3140 ELSE RU
N
```

My ending if you lose the game is quite simple, but it does tempt the player into having another go. Having the screen scroll up with the crunching sound is performed by a separate subroutine within the main routine. This short subroutine is shown below.

```
3150 FOR T=1 TO 500:NEXT T
3160 PLAY"V31L255O1GEC"
3170 PRINT" "
3180 RETURN
```

My 'winning' ending is a little more complex. A text message is first printed up on the screen, followed by the player being given an 'official' title. The game finally ends with a continuous high resolution display using the Dragon 32's powerful LINE command.



```
3190 REM***YOU WIN***
3200 CLS:PLAY"O4L12CL10DL12CL4D"
3210 PRINT "**********WELL DONE*********
***"
3220 PRINT:PRINT"    YOU HAVE CONQUERED
THE"
3230 PRINT"          dark forest"
3240 PRINT:PRINT"   YOU OUTWITTED YOUR E
NEMIES,  EXPLORED THE FOREST AND FOUND
 THE CREATURE."
3250 PRINT:PRINT"  YOU HAVE RID THE COUN
TRY OF A GREAT THREAT AND YOU SHALL BE
 REWARDED."
3260 PRINT:INPUT"WHAT IS YOUR NAME";NAME
$
3270 CLS:FOR T=1 TO 15:PRINT:NEXT:PRINT"
     ARISE SIR ";NAME$:PRINT"       OF D
RAGONIA"
3280 FOR T=1 TO 15:PRINT:PLAY"O4L20CEFFG
":NEXT T
3290 FOR T=1 TO 2000:NEXT T
3300 PMODE 3,1:PCLS:SCREEN 1,0
3310 FOR T=60 TO 155:LINE(T,T-30)-(250-T
,220-T),PSET,B
```

```
3320 LINE(T,T-30)-(250-T,220-T),PRESET,B
3330 COLOR RND(4),RND(4):SOUND RND(100)+
120,1:NEXT T
3340 PCLS:SOUND RND(200)+20,1:GOTO 3310
```

We have now completed an analysis of a complete adventure, phew! As we have already discussed, there is no one definitive style of adventure so in the next chapter we consider a different but no less popular type of adventure which provides quite a marked contrast with 'The Dark Forest'.

Here now, for your delectation, is the complete listing of 'The Dark Forest'. Good luck and above all, have fun!

```
10 REM****(C),CLIVE GIFFORD****
20 REM*****THE DARK FOREST*****
30 REM**A MEDIEVAL ADVENTURE.**
40 REM***************************
50 GOSUB 2630
60 CLS 0:PRINT @ 166,"THE DARK FOREST";
70 PRINT @ 195,"PRESS ANY KEY TO START";
80 SOUND 200,1
90 IF INKEY$="" THEN 80
100 PRINT @ 323,"YOU HAVE DARED TO ENTER
";
110 PRINT @ 361,"GOOD LUCK";
120 FOR T=1 TO 1000:NEXT T
130 FOR X=1 TO 1500:NEXT X:CLS RND(9)-1
140 SOUND 190,1:GOSUB 730:PRINT
150 IF N$="N" AND Z(B,1)<>0 THEN B=Z(B,1
)
```

```
160 IF N$="S" AND Z(B,2)<>0 THEN B=Z(B,2
)
170 IF N$="E" AND Z(B,3)<>0 THEN B=Z(B,3
)
180 IF N$="W" AND Z(B,4)<>0 THEN B=Z(B,4
)
190 IF N$<>"R" THEN 230
200 CS=INT((B+10)/10):PRINT"  O.K. YOU C
AN SLEEP HERE BUT ITWILL COST YOU";CS;"C
OIN/S FOR EVERY HOURS UNINTERRUPTED REST
.":PRINT
210 INPUT"HOW MANY HOURS";H:IF H<1 OR M(
2)-(H*CS)<0 THEN PRINT"TRY AGAIN, SLEEPY
HEAD!":GOTO 210
220 M(2)=M(2)-(H*CS):M(1)=M(1)+H:CLS:PRI
NT @ 236,"ZZZZ":FOR T=1 TO 1500*H:NEXT T
:CLS:GOSUB 730:GOTO 250
230 IF CHK=B THEN PRINT"cannot move in t
hat direction":PLAY"L1O1C":GOTO 260
240 CHK=B
250 ON B GOSUB 980,1000,1020,1170,1200,1
250,1270,1290,1310,1380,1520,1630,1660,1
680,1700,1760,1880,1910,1930,1950
260 IF Z(B,5)<>0 THEN 300
270 GOSUB 790
280 PRINT:INPUT"WHAT NOW";N$:N$=LEFT$(N$
,1)
290 GOTO 130
```

```
300 REM**MONSTER/FIGHT**
310 MN=Z(B,5):PRINT
320 IF MN=1 THEN PRINT"A BAND OF 10 ROBB
ERS SEIZE YOU.":E=40
330 IF MN=2 THEN PRINT"YOU HAVE COME ACR
OSS GRENDAL,    THE HERMIT. YOU HAVE AWOK
EN HIM HE IS NOT PLEASED.":E=40+RND(15)
340 IF MN=3 THEN PRINT"SIR RUFUS OF GASC
ONY ATTACKS...":E=20
350 IF MN=4 THEN PRINT"A FEROCIOUS BEAR,
 RAVENOUS FOR  MEAT POUNCES.":E=20
360 IF MN=5 THEN PRINT" YOU HEAR THE WAR
CRY OF THE       FEARLESS BLACK KNIGHT.":E
=60
370 INPUT"WILL YOU FIGHT OR RUN";F$:F$=L
EFT$(F$,1)
380 PRINT:IF F$="F" THEN 410
390 IF RND(10)>4 THEN INPUT"WHICH DIRECT
ION COWARD";N$:M(1)=M(1)-4:PRINT:GOTO 13
0
400 PRINT"NO YOU MUST STAND & FIGHT' AS
A PUNISHMENT FOR YOUR COWARDICE   YOU LO
SE 10 STRENGTH":M(1)=M(1)-10
410 REM**FIGHT**
420 Z(B,5)=0:RO=RND(19)
430 IF Z(RO,5)=0 THEN Z(RO,5)=MN ELSE GO
TO 420
440 INPUT"WHICH ATTRIBUTE DO YOU WISH TO
```

```
   FIGHT WITH":A$
450 IF A$="STRENGTH" THEN A=1
460 IF A$="MONEY" THEN A=2
470 IF A$="MAGIC" THEN A=3:E=INT(E/10)
480 IF A=1 AND M(1)<1 THEN 3210
490 IF A=2 AND M(A)<1 THEN PRINT"CANNOT
USE MONEY":GOTO 440
500 IF A=3 AND M(A)<1 THEN PRINT"CANNOT
USE MAGIC"
510 CLS 0:IF M(A)>60 THEN DD=M(A)-60:M(A
)=60
520 FOR X=1 TO M(A):SET(X,12,8):NEXT X
530 PRINT @ 40,"THE BATTLE COMMENCES";
540 PRINT @ 128,"YOU";:PRINT @ 288,"ENEM
Y";
550 FOR EX=1 TO E:SET(EX,22,MN):NEXT EX
560 SOUND 1,12
570 REM**MAIN FIGHT LOOP**
580 IF RND(7)=4 THEN GOTO 600
590 RESET(E,22):E=E-1
600 SOUND 180,2
610 IF RND(3)=1 THEN GOTO 630
620 RESET(M(A),12):M(A)=M(A)-1
630 SOUND 160,2
640 IF E>0 AND M(A)>0 THEN GOTO 570
650 IF M(A)=0 THEN GOTO 2910
660 PRINT @ 330,"YOU WIN!!!";
670 PLAY"L8O4CDEFGP4CDEFG"
```

```
680 ER=RND(40):BV=RND(7)
690 IF BV<4 THEN PRINT @ 386,"YOU GAIN":
ER;"STRENGTH"::M(1)=M(1)+ER+DD
700 IF BV<7 AND BV>3 THEN PRINT @ 386,"Y
OU GAIN";ER;"MONEY UNITS";:M(2)=M(2)+ER+
DD
710 IF BV=7 THEN ER=INT(ER/10)+1:PRINT @
 386,"YOU GAIN";ER;"MAGIC SPELLS";:M(3)=
M(3)+ER+DD
720 FOR T=1 TO 1500:NEXT T:CLS:GOTO 250
730 REM**ATTRIBUTES**
740 M(1)=M(1)-1:IF M(1)=0 THEN 2910
750 PRINT"STRENGTH";INT(M(1))
760 PRINT"MONEY";M(2)
770 PRINT"MAGIC";M(3)
780 RETURN
790 IF RND(10)>8 AND B<>1 AND B<>15 AND
B<>20 THEN PS=RND(5):GOTO 810
900 RETURN
810 IF PS=1 THEN PRINT"  AN OLD GLASS BO
TTLE IS WITHIN REACH. IT IS COVERED IN C
OBWEBS"
820 IF PS=2 THEN PRINT"  HERE IS A LARGE
 OAKEN CHEST.  A WIERD SIGN IS CARVED ON
 IT'S  LID."
830 IF PS=3 THEN PRINT"  A VELLUM PARCHM
ENT BOOK LIES  BY YOUR FEET."
840 IF PS=4 THEN PRINT"  A GOLD LOCKET O
```

```
N A CHAIN IS   CLOSE TO YOU."
950 IF PS=5 THEN PRINT"  BEFORE YOU IS A
 BOX OF EXOTIC FRUIT"
960 REM****POSSESSIONS****
970 INPUT"DO YOU WANT TO OPEN IT":OP$
980 PRINT:IF LEFT$(OP$,1)<>"Y" THEN RETU
RN
990 IF RND(10)>7 THEN PRINT"AN ORC,THE S
ERVANTS OF THE BLACKENID, JUMPS OUT AND
ATTACKS YOU.IT DISAPPEARS AFTER A STRUGG
LE  LEAVING YOU WEAK AND TIRED.":M(1)=M(
1)-(M(1)/4):PS=0
900 IF PS=1 THEN PRINT"A LARGE CLOUD OF
SMOKE APPEARS. IT'S SULPHUROUS ESSENCE G
IVES   YOU NEW STRENGTH AND LIFE":M(1)=M
(1)+8
910 IF PS=2 THEN PRINT"OH JOY OF JOYS! M
UCH TREASURE   IS CONTAINED IN THE CHEST
. YOU  ARE A RICH MAN":M(2)=M(2)+35+RND(
15)
920 IF PS=3 THEN PRINT"IT CONTAINS 6 POW
ERFUL SPELLS.":M(3)=M(3)+6
930 IF PS=4 THEN PRINT" IT HAS A SMALL P
ORTRAIT OF THE BEAUTIFUL PRINCESS CAROLI
NE DE  MOSELLE. IT IS QUITE VALUABLE.":M
(2)=M(2)+10
940 IF PS=5 THEN PRINT"YOU ARE SO TEMPTE
D THAT YOU EAT SEVERAL. THEY ARE DELICIO
```

```
US AND DO NO HARM BUT ONLY ADD TO YOUR S
TRENGTH.":M(1)=M(1)+12
950 PRINT:PRINT"****PRESS A KEY TO CONTI
NUE****"
960 IF INKEY$="" THEN 960 ELSE CLS:RETUR
N
970 REM**LOCATIONS**
980 PRINT"     YOU ARE IN A SMALL CLEARIN
G WITHIN THE DARK FOREST. YOU CAN PROCEE
D IN ANY DIRECTION THAT   YOU WISH"
990 RETURN
1000 PRINT"  YOUR VIEW FROM OAKRIDGE HIL
L  IS MAGNIFICENT, IN THE DISTANCE YOU C
AN SEE GRENDAL THE          HERMIT'S LODG
E"
1010 RETURN
1020 PRINT"  YOU HAVE FOUND YOURSELF IN
A    PEACEFUL, LUSH FIELD."
1030 PRINT:PRINT"THERE IS A SMALL EMPTY
WELL NEXTTO YOU. DO YOU WANT TO GO DOWN
 IT.(Y/N)":INPUT O$
1040 IF O$="N" THEN PRINT"  ALRIGHT, STA
Y IN THE PEACEFUL FIELD THEN.":RETURN
1050 IF RND(3)<2 THEN 1120
1060 PMODE 3,1:PCLS:SCREEN 1,0:CIRCLE(12
8,40),20
1070 CIRCLE(119,32),2:CIRCLE(137,32),2:C
IRCLE(128,40),3:CIRCLE(128,55),9,4,1,.6,
.9
1080 COLOR 2,2:LINE(0,0)-(40,192),PSET,B
F
1090 LINE(215,0)-(255,192),PSET,BF:PAINT
(45,190),3,2
1100 FOR T=1 TO 1000:CLS:PRINT:PRINT" AH
,WELL,I THOUGHT IT WAS EMPTY":PRINT"HOW
SILLY OF ME!":M(1)=10:M(2)=10:M(3)=1:B=1
1110 PRINT:PRINT"  YOU DO MANAGE TO GET
OUT, BUT IN DOING SO LOSE MUCH MONEY AND
 STRENGTH.":GOTO 1150
1120 PLAY"L18O4GDFBBACEBB":PRINT:PRINT"A
 SMALL NICHE IN THE WELL WALL  HIDES A G
OLDEN FIGURE,THE RAM OFANAG."
1130 PRINT"   INSCRIBED ON THE UNDERSIDE
 IS THE MYTHICAL WORD grandos. THE FIGUR
E IS QUITE VALUABLE, LOOK  AFTER IT."
1140 P=1:M(3)=M(3)+1:M(2)=M(2)+5
1150 PRINT:PRINT"***PRESS A KEY TO CONT
INUE****"
1160 IF INKEY$="" THEN 1160 ELSE GOSUB 7
30:RETURN
1170 PRINT"  THE BARON'S RESIDENCE IS
   NEARBY. YOU CAN HEAR BEAUTIFUL  MUSIC
 BEING PLAYED...."
1180 PLAY"O3L8EL6GL8AL7BL12O4C#L8O3BL6AL
9F#L7DL12EL8F#"
1190 RETURN
```

```
1200 PRINT"  YOU ARE IN A DAMP HUT WITH
A  POORLY THATCHED ROOF. A MUSTY   SMELL
 LINGERS AND THERE IS AN   ADJOINING ROO
M TO THE RIGHT."
1210 FOR T=1 TO 2000:NEXT T
1220 PLAY"L8O4CO2CO4CO2CO4CO2C"
1230 PRINT"WHATS THIS? A SMALL HIDDEN DO
OR IN FRONT OF YOU (N)."
1240 RETURN
1250 PRINT"AROUND YOU IS A SPARSELY FILL
ED STOREROOM. THIS IS A DEAD-END."
1260 RETURN
1270 PRINT"  YOU STAND IN THE MIDDLE OF
   BERRY HAMLET. IT IS VERY QUIET, THE P
LACE SEEMS deserted"
1280 RETURN
1290 PRINT"  YOU ARE ON A GRASSY HILL
   SLOPING DOWNWARDS INTO A VALLEY. A SM
ALL HAMLET LIES AT THE     NORTHERN END.
"
1300 RETURN
1310 PRINT"  YOU ARE STANDING OUTSIDE A
    WOODCUTTERS HUT. ENTER AT YOUR OWN P
ERIL":PLAY"O1L4DEL12CDC"
1320 PRINT:PRINT"TYPE IN THE RIGHT CODE
TO ENTER THE HUT":INPUT E$:PRINT
1330 F$="IVORYMAGICSILVEGRENDRUFUSDIAMOB
ERRYBARON"
```

```
1340 FOR T=1 TO RND(8):FF$=MID$(F$,T*5-4
,5):NEXT T:IF LEFT$(E$,5)=FF$ OR(E$="DRA
GON" AND P=1) THEN 1360
1350 PRINT"YOU USED THE WRONG CODE. YOU'
VE BEEN CHASED AWAY BY THE WOODSMANTO AN
OTHER PART OF THE FOREST":B=RND(3)+1:M(1
)=M(1)-8:RETURN
1360 PRINT"CORRECT CODE! ENTER THE HUT B
Y  HEADING 'N'":PLAY"L4O1CP4DP4EP4":RETU
RN
1370 N$="N":PLAY"L4O1CP4DP4EP4":RETURN
1380 PMODE 4,1:PCLS:SCREEN 1,1
1390 LINE(90,30)-(130,70),PSET,BF:LINE(9
8,30)-(104,70),PRESET,BF
1400 LINE(114,30)-(120,70),PRESET,BF
1410 LINE(0,194)-(30,174),PSET:LINE(255,
194)-(225,174),PSET
1420 DRAW"BM30,0;D174R195U174"
1430 FOR T=1 TO 2500:NEXT T
1440 CLS 0:PRINT @ 162,"YOU FELL INTO A
BEAR TRAP AND WERE CAPTURED BY THE BLACK
      KNIGHT WHO HAS IMPRISONED YOU."
1450 PLAY"V31L6O3GFEEO2GFEEO1L4GFEDCCC":
PRINT
1460 INPUT"WHAT NOW";N$:IF N$="GRANDOS"
THEN PRINT"CLOSE, BUT WRONG,YOU SHOULD H
AVETHOUGHT ABOUT THE FIGURE'S NAME":PRIN
T:PRINT"THE ram OF anag....THE anag ram"
```

```
1470 IF N$="DRAGONS" AND P=1 THEN PMODE
3,1:PCLS:SCREEN 1,0:FOR T=1 TO 8:CLS T:S
OUND 25*T,4:NEXT
1480 IF N$="DRAGONS" AND P=1 THEN PRINT
@ 232,"THE MAGIC WORD!":PRINT @416,"YOU'
RE FREE AGAIN, BUT YOU LOST THE GOLDEN F
IGURE":B=1:P=0:RETURN
1490 PRINT:PRINT"you failed and are imor
isoned   forever.":END
1500 GOTO 1380
1510 RETURN
1520 PMODE 3,1:PCLS:SCREEN 1,0
1530 CIRCLE(70,110),80,2,0.6,0.5,1
1540 CIRCLE(200,110),100,2,0.7,0.6,3
1550 LINE(0,110)-(255,110),PSET
1560 PAINT(180,180),2,4
1570 PAINT(0,0),3,2
1580 CIRCLE(20,20),10,2:PAINT(20,20),2,2
1590 PAINT(190,100),4,2
1600 FOR T=1 TO 2000:NEXT T:CLS:GOSUB 73
0
1610 PRINT"YOU ARE ON THE BARON'S LAND,
BE CAREFUL. IN THE DISTANCE, LIE   THE S
ACRED HILLS OF ANGELSARK"
1620 RETURN
1630 PRINT"  THE DARK FOREST SURROUNDS Y
OU. IT'S GROTESQUE FORMS SEEM TO   MOVE
CLOSER. THIS PLACE IS evil"
```

```
1640 FOR T=45 TO 1 STEP-3:SOUND T,3:NEXT
 T
1650 RETURN
1660 PRINT"YOU ARE BY THE SIDE OF A SMAL
L  STREAM. YOU CAN FOLLOW THE RIVER BANK
 BOTH EAST AND WEST OR YOU CAN RETREAT N
ORTH"
1670 RETURN
1680 PRINT"  A TWISTING TRACK CUTS THROU
GH THE TANGLED BUSHES.THIS IS WILD BOAR
COUNTRY."
1690 RETURN
1700 PRINT"  YOU ARE BESIDE A FAST-FLOWI
NG RIVER. A BRIDGE GUARDED BY A    GREED
Y TROLL LIES TO THE SOUTH. YOU MUST GUES
S THE FEE THAT THE BEAST CHARGES TO CROS
S"  "
1710 INPUT F:IF F<RND(40)+20 THEN B=Z(B,
1):PRINT"HE SCORNS YOUR TINY SUM AND
 CHASES YOU AWAY":RETURN
1720 IF M(2)-F<0 THEN PRINT"YOU GUESSED
RIGHT,BUT CANNOT PAYTHE TOLL. BE ON YOUR
 WAY, FRIEND":RETURN
1730 PRINT:PRINT"THAT IS CLOSE ENOUGH, C
ROSS NOW."
1740 M(2)=M(2)-F
1750 RETURN
1760 PMODE 3,1:PCLS:SCREEN 1,0
```

```
1770 DRAW"BM0,100;C2R50U70R6D5R5U5R5D25R
30U10R40D10R30U25R6D5R5U5R5D70R75L245"
1780 DRAW"BM103,100;U20R2U2R2U2R2U2R15D2
R2D2R2D2R2D20"
1790 PAINT(90,90),4,2
1800 PAINT(0,0),3,2
1810 LINE(103,100)-(88,194),PSET
1820 LINE(133,100)-(148,194),PSET
1830 PAINT(0,120),2,2:PAINT(255,120),2,2
1840 FOR T=1 TO 2000:NEXT T:CLS:GOSUB 73
0
1850 PRINT"YOU SEE THE MYTHICAL CASTLE O
F  DIAMONDS. DO NOT BELIEVE YOUR   EYES,
 WHAT YOU SEE BEFORE YOU   IS AN ILLUSIO
N CAST BY THE MOST POWERFUL OF ALL MAGIC
IANS, THE  WARLOCK OF THE DARK FOREST."
1860 PLAY"P1O1L100CDEFGGGFFEEDDCCCCCC"
1870 RETURN
1880 PRINT"BEFORE YOU LIES THE MAGNIFICE
NT WATCH TOWER OF PENDRAGON, A     SPLEN
DID GOTHIC MONUMENT."
1890 PLAY"O4L4CEDFEGL12ABABBA"
1900 RETURN
1910 PRINT" YOU ARE AT THE ENTRANCE TO A
   MYSTERIOUS CAVE. THE SUN ALWAYS SETS
TO THE RIGHT OF THE CAVE."
1920 RETURN
1930 PRINT"  FROM YOUR POSITION IN A SMA
```

```
LL MARKET SQUARE. YOU CAN SEE SOME STOCK
S AND A PILLORY A REMINDER OF WHAT WILL
HAPPEN IF YOU FAIL.""
1940 RETURN
1950 REM***END GAME***
1960 PRINT:PRINT"IT'S VERY DARK INSIDE T
HE CAVE"
1970 FOR T=1 TO 1000:NEXT T:PRINT"YOU HE
AR A LOUD NOISE"
1980 PLAY"V31L255O3BAGFEDCO2BAGFEDCO1BAG
FEDC"
1990 PRINT"IT'S TH-TH-THE C-C-C-CREATURE
."
2000 FOR T=1 TO 1000:NEXT T
2010 PMODE 4,1:PCLS:SCREEN 1,1
2020 R=RND(8)
2030 CIRCLE(130,100),40,1,0.8,0,0.5
2040 LINE(90,100)-(170,100),PSET
2050 PAINT(100,110),1,1
2060 LINE(90,115)-(170,115),PRESET
2070 FOR X=100 TO 170 STEP 10
2080 LINE(X,100)-(X,140),PRESET
2090 NEXT X
2100 CIRCLE(90,50),15,1,0.6
2110 CIRCLE(165,50),15,1,0.6
2120 CIRCLE(90,50),5:CIRCLE(165,50),5:
2130 DRAW"BM120,70;D8BR20U8"
2140 PAINT(90,50),1,1:PAINT(165,50),1,1
```

```
2150 PRESET(90,50):PRESET(165,50)
2160 FOR M=1 TO 8
2170 PLAY"V31O5L240GABGABAGABAGABAGABAAB
AAGG"
2180 NEXT M
2190 CLS 0
2200 PRINT"IF ANY OF YOUR ATTRIBUTES ARE
   LESS THAN ";R;" THEN RUN AWAY!!      (
PRESS A KEY)"
2210 TIMER=0
2220 IF INKEY$<>"" AND TIMER<250 THEN PR
INT @ 330,"YOU JUST MADE IT! YOU RUN BAC
K  OVER THE BRIDGE.":B=14:GOTO 250
2230 IF TIMER<250 THEN GOTO 2220
2240 IF M(1)<R OR M(2)<R OR M(3)<R THEN
CLS:GOTO 2910
2250 PRINT"LET THE FIGHT BEGIN....."
2260 PLAY"L4O3EFC":CLS
2270 DIM K(3)
2280 K(1)=RND(50)+10
2290 K(2)=RND(45)+15
2300 K(3)=RND(12)
2310 FOR Y=1 TO 3
2320 FOR T=1 TO 1000:NEXT T
2330 CLS:PRINT @ 115,"ROUND";Y
2340 PRINT @ 258,"human";M(Y)
2350 PRINT @ 274,"creature";K(Y)
2360 IF RND(10)=3 THEN 2380
```

```
2370 M(Y)=M(Y)-1
2380 IF RND(10)=7 THEN 2400
2390 K(Y)=K(Y)-1
2400 IF M(Y)<1 THEN 2560
2410 IF K(Y)<1 THEN 2520
2420 PLAY"L240O1CFBO2EAO3DGO4C"
2430 PRINT @ 390,"batTLE STARTS"
2440 PLAY"L255O3BAGFEDC"
2450 PRINT @ 390,"BATtle STARTS"
2460 PLAY"O2BAGFEDC"
2470 PRINT @ 390,"BATTLE staRTS"
2480 PLAY"O1BAGFEDC"
2490 PRINT @ 390,"BATTLE STArts"
2500 PLAY"O1DCDC"
2510 GOTO 2340
2520 FOR T=1 TO 1000:NEXT T
2530 CLS 2:PRINT @ 100,"YOU WIN THIS ROU
ND";
2540 PLAY"L4O3FGAEE"
2550 V=V+1:IF V=2 THEN 3190 ELSE NEXT Y
2560 FOR T=1 TO 1000:NEXT T
2570 CLS 4:PRINT @ 100,"THE CREATURE WIN
S";
2580 PLAY"L4O1FGAEE"
2590 W=W+1:IF W=2 THEN 2910
2600 NEXT Y
2610 END
2620 REM***(+INITIALISE)***
```

```
2630 DIM M(3):DIM Z(21,5):DIM Q(5,5)
2640 M(1)=RND(30)+20
2650 M(2)=RND(70)+10
2660 M(3)=RND(6)
2670 FOR A=1 TO 20
2680 FOR B=1 TO 5
2690 READ Z(A,B)
2700 NEXT B,A
2710 DATA 10,13,11,8,0,19,7,3,0,0
2720 DATA 0,10,4,2,0,0,19,5,3,0
2730 DATA 8,9,6,4,0,0,0,0,5,0
2740 DATA 2,8,10,0,0,7,12,1,0,0
2750 DATA 5,11,0,19,0,0,0,0,0,0
2760 DATA 9,0,17,1,0,8,0,13,0,0
2770 DATA 1,0,14,12,0,0,2,15,13,0
2780 DATA 11,18,16,14,0,17,0,0,15,0
2790 DATA 0,16,0,11,0,0,20,6,0,0
2800 DATA 4,0,9,10,0,18,0,0,0,0
2850 B=1:P=0
2860 FOR A=1 TO 5
2870 RO=RND(18)+1
2880 IF Z(RO,5)=0 THEN Z(RO,5)=A ELSE GO
TO 2870
2890 NEXT A
2900 RETURN
2910 REM***END OF GAME***
2920 FOR T=1 TO 15:PRINT" "
2930 NEXT T
```

```
2940 PRINT"     YOU HAVE BEEN BEATEN"
2950 GOSUB 3150
2960 PRINT"      YOUR STRENGTH COULD"
2970 GOSUB 3150
2980 PRINT"      NOT HOLD OUT AS YOU "
2990 GOSUB 3150
3000 PRINT"     TRIED IN VAIN TO FIND"
3010 GOSUB 3150
3020 PRINT"     THE SECRETS THAT THE"
3030 GOSUB 3150
3040 PRINT"     DARK FOREST HOLDS..."
3050 FOR C=1 TO 10
3060 GOSUB 3150
3070 NEXT C
3080 GOSUB 3150
3090 PRINT"   BUT DO NOT GIVE UP,MORTAL"
3100 GOSUB 3150
3110 PRINT"  FOR YOU MAY STILL SUCCEED."
3120 GOSUB 3150
3130 PRINT"   TO TRY AGAIN PRESS A KEY"
3140 IF INKEY$="" THEN GOTO 3140 ELSE RU
N
3150 FOR T=1 TO 500:NEXT T
3160 PLAY"V31L255O1GEC"
3170 PRINT" "
3180 RETURN
3190 REM***YOU WIN***
3200 CLS:PLAY"O4L12CL10DL12CL4D"
```

```
3210 PRINT "**********WELL DONE*********
***"
3220 PRINT:PRINT"    YOU HAVE CONQUERED 
THE"
3230 PRINT"          dark forest"
3240 PRINT:PRINT"   YOU OUTWITTED YOUR E
NEMIES,  EXPLORED THE FOREST AND FOUND
 THE CREATURE."
3250 PRINT:PRINT"  YOU HAVE RID THE COUN
TRY OF A GREAT THREAT AND YOU SHALL BE
 REWARDED."
3260 PRINT:INPUT"WHAT IS YOUR NAME";NAME
$
3270 CLS:FOR T=1 TO 15:PRINT:NEXT:PRINT"
     ARISE SIR ";NAME$:PRINT"       OF D
RAGONIA"
3280 FOR T=1 TO 15:PRINT:PLAY"O4L20CEFFG
":NEXT T
3290 FOR T=1 TO 2000:NEXT T
3300 PMODE 3,1:PCLS:SCREEN 1,0
3310 FOR T=60 TO 155:LINE(T,T-30)-(250-T
,220-T),PSET,B
3320 LINE(T,T-30)-(250-T,220-T),PRESET,B
3330 COLOR RND(4),RND(4):SOUND RND(100)+
120,1:NEXT T
3340 PCLS:SOUND RND(200)+20,1:GOTO 3310
```

CHAPTER 3

A FULL SCALE TEXT ADVENTURE - THE NIELSON PAPERS

Moving on from our first program, we have here an adventure set around a less common scenario - a block of offices. The adventure requires you to collect the secret Nielson Papers, Nielson being the scientist who first managed to perfect the Uranium substitute, Syntheron 2. The company he works for are now making vast sums of money from this invention, and rival companies just cannot wait to get their hands on the formular and manufacturing techniques described in these papers. You are employed by the fuel giant, Trithon Ltd, and must steal the papers and escape safely from the back of the office block.

To complete your task, you will need to find various pieces of equipment, keep yourself out of the many traps that exist and solve the complex problems placed around the adventure. Of course, the building is patrolled by security guards who are trained to shoot on sight. You are fortunate in that you are wearing a bullet-proof vest which will take quite a few shots before finally allowing the fatal round through. You are given a gun, told of your mission and transported to the front of the office complex. From there on, the success of your mission is up to you. The Nielson Papers are your goal . . .

Stirring stuff. The adventure is rather different in design and actual game play to 'The Dark Forest'. You have a full range of commands that you must discover and use. I will not give these away but suffice to say, they allow you to acquire objects, battle with the guards, and move around the adventure scenario among other things.

The descriptions of the locations are much shorter than the previous adventure. Much of the adventure is taken up with handling commands, approximately 30 of them in all. The computer, in this game, allows quite a wide syntax to any commands. The best way is to find out which commands it will acceptand which it refuses to understand. In many situations, the computer gives more help than the simple 'I CAN'T DO THAT' message which can be so frustrating in adventures. If you find you are getting nowhere in this adventure, then have a look at chapter eight of this book.

Certain areas of this program have been put under scrutiny. The whole program is listed at the end of the chapter. It is a good idea to type the whole program in first and then return to the chapter for a discussion of the various routines within the listing.

The first area of the program to discuss is the 'initialisation' routine.

```
10 REM****THE NIELSON PAPERS****
20 REM*****A TEXT ADVENTURE*****
30 GOSUB 1730:GOSUB 1440
```

The program first displays the title page. As this program is all text, the title page uses some simple string slicing to give an effective display.

```
1730 CLS:NN$="*THE NIELSON PAPERS*"
1740 PRINT @ 130,"A FULL SCALE TEXT ADVE
NTURE":PRINT @ 364,"C.1983."
1750 FOR F=1 TO 3
1760 FOR T=1 TO 20:PRINT @ 228,MID$(NN$,
T,21-T):PRINT @ 249-T,LEFT$(NN$,T)
1770 PLAY"P10":NEXT:NEXT
```

```
178Ø PRINT @ 228," ";:FOR T=1 TO 1ØØØ:NE
XT T:RETURN
```

Line 176Ø performs the leftwards scroll feature. This line is in a loop which splits NN$ (the string holding the title) into two pieces. The piece from the loop pointing to the right is printed on the left part of the screen and the other part of the screen, ie. that which is to the left, is printed to the right of the other portion. The loop gradually increases the size of the left part of the string and decreases the right side of the string. When these are continually displayed on-screen in the same position (using PRINT @) it gives the effect of the title scrolling to the left, off the edge of the screen, and magically re-appearing on the other side of the screen.

Once the title page is displayed, the program then initialises, with the title page still displayed. All the arrays are dimensioned in line 145Ø. Most of the 'initialisation' routine is concerned with filling the arrays with the data necessary for the game. O$ is filled with the names of the 18 objects which can be found in the game. L$ is filled with the description of the 3Ø locations in the game, while the array, L, is filled with the starting positions of the 18 objects. Array A is given the monetary values of the 18 objects, and array P contains the room logic.

The other function performed in the 'initialisation' routine is the defining of the four large strings used in the game. C$ contains all the names of the commands available in the game, but only the first three letters of each. O$ contains the first three letters of each object, while D$ holds the direction string used when there is a locked door or blocked entrance in the adventure. P$ is what I have termed as the 'response' string. When you enter a command which the computer does not understand in any way, it will randomly choose one of the four responses to display on-screen. This removes the monotony of having the computer constantly repeat the same phrase every time you have trouble.

Now that we have looked at the 'initialisation' section in total, I will take just a couple of areas of the rest of the program and detail them more exactly.

The 'initialisation' routine ...

```
1440 REM***INITIALISATION***
1450 DIM O$(18),L$(30),P(30,4),L(18),A(1
8)
1460 FOR T=1 TO 18:READ O$(T):NEXT
1470 FOR T=1 TO 30:READ L$(T):NEXT
1480 FOR T=1 TO 18:READ L(T):NEXT
1490 DATA GUN,TORCH,COFFEE-MACHINE,SCRIB
BLED-MEMO,POCKET-COMPUTER,GLOVES,CUP,STO
OL,WIRECUTTERS
1500 DATA MONEY,COIN-MACHINE,SILENCER,LA
DDER,KEY,PAPER-CLIPS,ROPE,SECRET-PAPERS,
PAPERS
1510 DATA THE MAIN ENTRANCE,THE RECEPTIO
N AREA,THE DOCUMENT ROOM,THE STAFF CANTE
EN,THE DELIVERY ROOM
1520 DATA "A CORNER OF THE YARD NEXT TO 
AN ELECTROCUTED FENCE","THE ALLEYWAY BEH
IND THE COMPLEX, A CAR IS READY TO TAKE 
YOU AWAY"
1530 DATA THE COMPUTER ROOM,THE MAINTAIN
ANCE AREA,A DARK CORRIDOR,THE RECORDS DE
PT,AN UNTIDY OFFICE
1540 DATA THE TYPING POOL,THE RECREATION
 ROOM,A NARROW CORRIDOR,A PLUSH CONFEREN
CE    ROOM,AN UNLIGHTED OFFICE
1550 DATA THE STRONGROOM NEXT  TO THE VA
ULT,THE STEEL VAULT,THE ACCOUNTS DEPT,TH
E EXECUTIVE WASHROOMS
1560 DATA "A COLD, BARE ROOM",A DARK COR
RIDOR,"A CLEAN, NEW STOREROOM",A LARGE C
```

```
UPBOARD,A BRIGHT HALLWAY
1570 DATA THE TELEX ROOM,A LIFT,THE DIRE
CTORS        BOARDROOM,A SPACIOUS OFFICE
1580 DATA 0,2,4,27,8,33,25,13,9,19,20,30
,5,10,13,11,34,11
1590 D$="NORTHSOUTHEAST WEST "
1600 C$="GO MOVTAKGETSTEDROREACLICUTCHAO
PEUNLFIRSHOWAIDRISEALOOWEACOMCALDECHELCL
UQUIPUTHITFIGATTKIL"
1610 O$="GUNTORCOFMEMCOMGLOCUPSTOWIRMONC
OISILLADKEYCLIROPSECPAP"
1620 P$="PARDON? WHAT?   RUBBISH!REPHRAS
E"
1630 L=1:L(1)=0:K=0:ST=10000:KY=0:GS=0
1640 G1=RND(28)+2:IF G1=28 OR G=7 THEN 1
640
1650 DC=RND(6)+4
1660 FOR X=1 TO 30:FOR Y=1 TO 4:READ P(X
,Y):NEXT:NEXT
1670 DATA 2,0,0,0,-1,1,4,0,0,0,2,0,0,10,
9,2,0,15,6,0,0,0,5,0,0,0,0,0,15,24,0,21,
20,8,13,9
1680 DATA 4,0,9,1,0,27,14,12,17,0,11,16,
16,0,0,9,26,0,18,11,5,8,0,21,-1,23,12,26
,22,12,15,0
1690 DATA 19,20,18,14,0,-1,0,0,18,9,0,0,
2,17,8,-1,0,17,-1,28,16,23,0,17,8,0,26,1
2,0,0,0,27
1700 DATA 6,14,0,24,11,-1,25,13,29,16,0,
0,28,0,0,30,0,0,29,0
1710 DATA 80,10,0,1,200,8,3,10,22,0,0,10
```

```
,40,10,1,27,1000000,2
1720 RETURN
```

The following lines of the program comprise the routine that tells the adventurer which objects can be seen and those already in the player's possession.

```
110 S=0:PRINT:PRINT"YOU HAVE:";
120 FOR D=1 TO 18:IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" " ELSE A$=" A "
130 IF S>0 AND L(D)=0 THEN PRINT TAB(9);
140 IF L(D)=0 THEN PRINT A$;O$(D):S=S+1
150 NEXT D
160 PRINT
170 Z=0:PRINT"YOU CAN SEE:";
180 FOR D=1 TO 18:IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" " ELSE A$=" A "
190 IF L(D)=L AND Z>0 THEN PRINT TAB(11)
;
200 IF L(D)=L THEN PRINT A$;O$(D):Z=Z+1
210 NEXT D
220 IF Z=0 THEN PRINT" NOTHING MUCH"
```

If you win . . .

```
1220 PRINT:PRINT"YOU WERE 100% SUCCESSFU
L IN YOURMISSION AND THE ORGANISATION HA
SKINDLY LET YOU KEEP THE OBJECTS THAT YO
U COLLECTED..."
1230 FOR T=1 TO 19:READ A(T):NEXT T
1240 FOR T=1 TO 19:IF L(T)=0 THEN PRINT
O$(T);" WORTH":PRINT A(T);"POUNDS."
1250 NEXT T
```

```
1260 PRINT"except the secret papers"
1270 FOR T=1 TO 5000:NEXT T
1280 FOR T=1 TO 8:PRINT:PLAY"O3L8C":NEXT
 T:PRINT TAB(12);"WELL DONE":FOR T=1 TO
8:PLAY"O4L8C":PRINT:NEXT T:END
```

If you manage to escape but forget something . . .

```
1290 PRINT:PRINT"YOU'RE SAFELY AWAY BUT
YOU DID  NOT STEAL THE SECRET PAPERS. TH
EORGANISATION ARE NOT PLEASED... YOU HAV
E BEEN SENT BACK TO TRY  AGAIN.":FOR T=1
 TO 4000:NEXT T:RUN
```

If you fail miserably . . .

```
1300 FOR T=1 TO 4000:NEXT T:FOR T=1 TO 1
6:PRINT:NEXT
1310 PRINT"  OOOOO   OOOOO   OOOOO   O
  ":PLAY"L16O2G"
1320 PRINT"  O       O   O     O     O":
PLAY"D"
1330 PRINT"  OOOO    OOOOO     O     O":
PLAY"O1A"
1340 PRINT"  O       O   O     O     O":
PLAY"E"
1350 PRINT"  O       O   O   OOOOO   OOO
OO":PLAY"C"
1360 PRINT:PRINT:GOTO 1310
```

Line 1580 gives all the starting values of the objects. It can be seen that object '1', the gun, is given the value zero. This means that the player is carrying the gun. When the program later accepts the player's command to take an object, it will put the location starting position of the object (held in the L array) equal to zero. If the item is dropped, the computer will give the object the value of the location the object was dropped in. There are then two routines that display what is in the same location and also what you are carrying. These routines are situated at lines 11Ø to 22Ø. It now seems to be the best time to look closely at those routines which are displayed following the 'initialisation' section.

What basically happens is that the computer checks through the 'object location' array and any object with the value zero is displayed in the player's inventory. Any object with the value equal to the location number that the player is at (held in the variable L) is printed as being seen by the player. Lines 12Ø and 18Ø are there simply for grammatical reasons, ie. to put an 'A' in front of certain objects. There is also the counter, S, that checks how many objects you are carrying. There is a maximum number of objects which can be carried which means the player has to make decisions in the later stages of the game as to what to carry and what to drop.

A vital part of this adventure program are the routines which accept and process the player's command.

Firstly, the computer must accept the player's command and then go to a routine which will handle the response. The lines below do this.

```
24Ø PRINT:INPUT"WHAT NOW";N$
25Ø GOSUB 137Ø
```

The routine at line 137Ø is very useful and basically dissects the parts of the command phrase or sentence that are required. This routine is shown below.

```
137Ø REM****COMMAND HANDLING****
138Ø T=1:IF LEN(N$)<3 THEN T=1ØØ:RETURN
139Ø B$=MID$(C$,T,3):IF B$=LEFT$(N$,3) T
HEN T=(T+2)/3:GOTO 142Ø
14ØØ IF T>97 THEN 143Ø
```

```
1410 T=T+3:GOTO 139Ø
1420 FOR CT=LEN(N$) TO 1 STEP-1:IF MID$(
N$,CT,1)=" " THEN CT$=MID$(N$,CT+1,3) EL
SE NEXT CT
143Ø RETURN
```

It would be useful to look further into the above routine, so here follows a line-by-line explanation. First, it checks to see that the command entered is less than three letters long; if this is the case it goes back to the main loop as the command entered was an invalid one. Lines 139Ø, 14ØØ and 141Ø form a loop without using the FOR... NEXT commands. In this loop, the first three letters of the player's command are checked with the 'command' string. If there is a command in the 'command' string that is equal to the player's command then the position that the command comes in the 'command' string is recorded and that is used as the basis for the information when the program goes back to the main loop. If no command is found then the program indicates that it is an invalid command (by assigning a value of greater than 97 to the variable T) and then returns to the main loop. If the command is a valid one then there is further string handling to be done. The last word of the command is found and the first three letters of that word are recorded in CT$. This is primarily used for when manipulation of objects within the adventure is required.

The program now returns to the main loop where if it is an invalid command, the computer prints up one of the messages from the response string, P$. The program then ON... GOSUBs to the line starting the subroutine to operate that particular command. The variable T is used here.

```
27Ø IF T>96 OR T<1 THEN PRINT:PRINT TAB(
12);MID$(P$,((RND(4)-1)*8)+1,8):GOTO 29Ø
28Ø ON T GOSUB 34Ø,34Ø,42Ø,42Ø,42Ø,51Ø,5
7Ø,63Ø,65Ø,68Ø,7ØØ,7ØØ,83Ø,83Ø,92Ø,93Ø,9
6Ø,96Ø,1Ø2Ø,1Ø4Ø,1Ø4Ø,1Ø4Ø,1Ø6Ø,1Ø7Ø,1Ø7
Ø,11ØØ,113Ø,113Ø,113Ø
```

We will not discuss all the commands available in this adventure, but I have chosen a couple which are fairly representative of what is in store for the budding adventurer. We will first have a look at the movement commands, MOVE and GO.

These commands are followed by the direction in which the player wishes to go, ie. north, south, east or west. The two words must be separated by a space. How this movement works is based on the logic used for linking locations in this adventure. Each location, if you have a look at the location data in the 'initialisation' section, has four pieces of data. Each refers to the new location the player would end up in if he or she went in a particular direction. The first number is for northern movement, the second for southern movement, and so on (in much the same way as 'The Dark Forest' system). When you decide to move in a certain direction, the computer gives the variable, K, a value, ie. '1' if it is north that you decided to move to, and so on. The program, if you are moving in an unblocked direction, makes the new location variable, L, equal to the figure in the room array which applies to your present position and the direction you wish to move in. For example, if you plan to move north and you are at location '1', then the new location value will be P(1, 1); that is the first location and movement up (K = 1). If you look up the data in line 167Ø, the first four numbers refer to location '1'. Moving north is handled by the first number which is a '2'. Therefore, the new location that you are moved to is location number '2'.

If after this process is gone through the new location is less than one, then the player cannot go that way and the program displays a suitable message and L is given its previous value back.

```
330 REM*****COMMANDS*****
340 REM
350 IF LEFT$(CT$,1)="N" THEN K=1
360 IF LEFT$(CT$,1)="S" THEN K=2
370 IF LEFT$(CT$,1)="E" THEN K=3
380 IF LEFT$(CT$,1)="W" THEN K=4
```

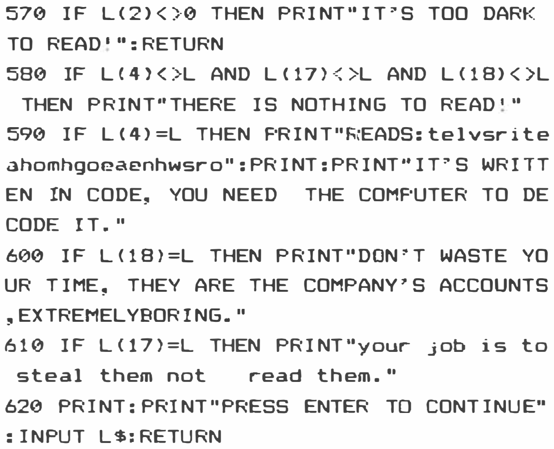
```
390 N=L:L=P(L,K)
400 IF L<1 THEN PRINT"YOU CAN'T GO THAT
WAY!":L=N
410 RETURN
```

Let's have a look at one of the 'red herring' commands, 'HELP'. This command, if entered, is responded to by a curt phrase, 'HELP YOURSELF, NO CLUES FROM ME'. The program then returns. This type of command soon arouses a humorous reply from the computer, takes up little memory and adds a little light-heartedness to the whole affair.

```
1060 PRINT"HELP YOURSELF, NO CLUES FROM
ME":RETURN
```

The READ command is an interesting command to look at and typifies many of the commands in this adventure. Many commands fall roughly into the following format: is the object required to do this as it is being carried by the player; is the player in the right location to do this; does the adventurer request the right object to use the command in conjunction with; and finally, are all the conditions, such as other essential objects, correct for the operation to take place. If there is anything wrong, some form of action (usually a message displayed) must be undertaken and the game goes back to the main loop. If everything is okay, then the program performs the operation, acknowledges that it has performed this operation and then returns to the main loop.

Let's transform this format into a practical example, READ. Firstly, if the torch is not carried by the player (ie. L (2), the part of the array denoting the torch, is not equal to zero) then the computer prints a 'too dark' message and returns. If none of the three readable items (the memo, the papers or the secret papers) are with the player in that location then the computer prints a 'nothing to read' message. If the memo is with the player (ie. L (4), the place in the object array for the memo, is equal to the adventurer's present location) then a coded message is printed and you are told that it is written in code and needs the computer to decipher it - yet another object for the adventurer to hunt for. If the player tries to read the papers, then he or she is told that they are just company accounts and nothing but a red herring. The player is then told off. After these actions have been followed, the computer waits for you to press a key to continue the game. The 'READ' routine is shown below.



```
570 IF L(2)<>0 THEN PRINT"IT'S TOO DARK
TO READ!":RETURN
580 IF L(4)<>L AND L(17)<>L AND L(18)<>L
 THEN PRINT"THERE IS NOTHING TO READ!"
590 IF L(4)=L THEN PRINT"READS:telvsrite
ahomhgoeaenhwsro":PRINT:PRINT"IT'S WRITT
EN IN CODE, YOU NEED  THE COMPUTER TO DE
CODE IT."
600 IF L(18)=L THEN PRINT"DON'T WASTE YO
UR TIME, THEY ARE THE COMPANY'S ACCOUNTS
,EXTREMELYBORING."
610 IF L(17)=L THEN PRINT"your job is to
 steal them not   read them."
620 PRINT:PRINT"PRESS ENTER TO CONTINUE"
:INPUT L$:RETURN
```

We will have a look at one part of another command, UNLOCK. This command, obviously enough, unlocks a locked door, providing you have the key, of course. The routine to use this command starts at line 700. The command checks that there is a locked door in the location, that you have the key and that it has not crumbled into dust due to over-use. It then checks that the last word in the sentence was 'door' so that the command phrases, 'UNLOCK THE SOUTH-WARDS FACING DOOR', 'UNLOCK LOCKED OFFICE DOOR' or simply 'UNLOCK DOOR' would all be accepted and yield the same response from the computer. This is true of all the commands, and some have an even wider choice of phrases that can be entered.

If all the previous conditions are met and the adventurer is in location 16 or location 28, then the door between these locations is opened and the program accordingly changes several numbers applying to the room logic. The southern direction in location number '28' is made equal to '16' and the northern direction in location '16' is made equal to '28'.

It can be seen that by manipulating the arrays, variables and strings set up in the early part of the game, it is a simple matter to move objects about, open and block passages, move around the locations, perform a wide number of actions involving certain conditions that must be met and subsequently, infuriate and amuse the player with simple one-line responses.

This sort of adventure is more versatile and generally more exciting to construct and play than most other kinds of adventure. The possibilities for commands are almost endless providing they are covered in the 'command' string and are set up logically with a subroutine which covers every possible condition.

What I have discussed with you about commands in this adventure could be said to be just the tip of the iceberg. Though the commands do act in a similar way to the ones shown above, what they actually do and what is needed to make them work has not been given away. When you start to play the game, you will see just how much can be crammed into well under 16K of memory. Because I have kept the game all text, converting it to other computers will not be that difficult. There is much memory space left that can be used to expand this game to almost double its present size. You can increase the memory of your computer by typing in POKE 25,6 and then typing NEW. The amount of memory you now have available is approximately 31Ø15 bytes (6½K more than previously available). What has, in fact, happened is that you have cleared the high resolution graphics pages. Using this technique stops you from using the graphics on the Dragon 32, but you do receive a terrific memory boost as a result. The decision is, of course, all up to you.

Before you get this adventure up and running on your computer, there are several points which deserve explanation. Firstly, when playing the game, after the computer responds to your command, there is a long pause. If you find the pause too long, then press a key and the program will continue. If, alternatively, you find the pause too short, then alter line 310 shown below.

```
310 FOR TT=1 TO 1000:IF INKEY$="" THEN N
EXT ELSE TT=1000
```

Secondly, you may have noticed that none of the adventures in this book feature a 'game save' or 'resurrection' feature. First, I will explain what both of these features are. A 'game save' feature simply lets you save your current status onto a cassette so that when you return to play the game some time later, you do not have to repeat the early stages of the game. A 'resurrection' feature allows a player killed during the adventure to continue playing the adventure albeit with a points or treasure loss. I have not included either feature because I do not feel that the adventures in the book need them. They are finely balanced between the computer and the player, and these extra features would make the game a lot easier to solve. 'The Dark Forest', with its random elements, would particularly suffer with the adventurer only going through the early stages once.

These two features are usually found on large commercial adventures but if you wish, you can add them to your program. The 'game save' feature can be added using the cassette file statements, and recording onto tape all the variables and arrays that have changed since the beginning of the game. A simple routine could then be developed to allow the data to be loaded back into the major program.

The 'resurrection' feature is far simpler. All you need to do is to add a small routine in front of the 'lose' routine at the end of the game. This short extra routine would take away a certain amount of treasure, points, etc, and let you carry on the game, perhaps moving you back to a location nearer the beginning of the adventure.

Here follows a complete listing of the program, The Nielson Papers. I hope you enjoy it. When you've typed it all in, why not check out the contents of this chapter if there are any routines you do not fully understand.

```
10 REM****THE NIELSON PAPERS****
20 REM*****A TEXT ADVENTURE*****
30 GOSUB 1730:GOSUB 1440
40 REM***MAIN LOOP***
50 ST=ST-1:IF ST=0 THEN 1300
60 CLS:PRINT:PRINT"YOU ARE IN ";L$(L)
70 IF L=7 AND L(17)=0 THEN 1220
80 IF L=7 AND L(17)<>0 THEN 1290
90 IF L=29 THEN PRINT"YOU ARE ON THE TOP
 FLOOR"
100 FOR M=1 TO 4:IF P(L,M)=-1 THEN PRINT
"LOCKED DOOR TO THE ";MID$(D$,(5*M)-4,5)
110 S=0:PRINT:PRINT"YOU HAVE:";
120 FOR D=1 TO 18:IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" " ELSE A$=" A "
130 IF S>0 AND L(D)=0 THEN PRINT TAB(9);
140 IF L(D)=0 THEN PRINT A$;O$(D):S=S+1
150 NEXT D
160 PRINT
170 Z=0:PRINT"YOU CAN SEE:";
180 FOR D=1 TO 18:IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" " ELSE A$=" A "
190 IF L(D)=L AND Z>0 THEN PRINT TAB(11)
;
200 IF L(D)=L THEN PRINT A$;O$(D):Z=Z+1
210 NEXT D
220 IF Z=0 THEN PRINT" NOTHING MUCH"
230 IF G1=L THEN PRINT:PRINT:PRINT"  gas
```

```
p! a security guard":DC=DC-1:IF DC=0 THE
N 1300
240 PRINT:INPUT"WHAT NOW";N$
250 GOSUB 1370
260 IF G1=L AND T<>13 AND T<>14 AND T<27
 THEN 300
270 IF T>96 OR T<1 THEN PRINT:PRINT TAB(
12);MID$(P$,((RND(4)-1)*8)+1,8):GOTO 290
280 ON T GOSUB 340,340,420,420,420,510,5
70,630,650,680,700,700,830,830,920,930,9
60,960,1020,1040,1040,1040,1060,1070,107
0,1100,1130,1130,1130
290 FOR TT=1 TO 500:NEXT:IF T<3 THEN 320
300 IF G1=L THEN PRINT"THE GUARD FIGHTS
BACK..."
310 FOR TT=1 TO 1000:IF INKEY$="" THEN N
EXT ELSE TT=1000
320 GOTO 40
330 REM*****COMMANDS*****
340 REM
350 IF LEFT$(CT$,1)="N" THEN K=1
360 IF LEFT$(CT$,1)="S" THEN K=2
370 IF LEFT$(CT$,1)="E" THEN K=3
380 IF LEFT$(CT$,1)="W" THEN K=4
390 N=L:L=P(L,K)
400 IF L<1 THEN PRINT"YOU CAN'T GO THAT
WAY!":L=N
410 RETURN
420 DV=0:FOR CT=1 TO 54 STEP 3:IF MID$(O
$,CT,3)=LEFT$(CT$,3) THEN DV=CT
430 NEXT:CT=(DV+2)/3:IF DV=0 THEN T=100:
```

```
RETURN
440 IF S>4 THEN PRINT"YOU MUST DROP SOME
THING FIRST":RETURN
450 IF (CT=8 OR CT=13) AND S>2 THEN PRIN
T"TO CARRY THE ";O$(CT);". YOU CAN    ON
LY CARRY 2 OTHER ITEMS.":RETURN
460 IF L(CT)<>L THEN PRINT"IT IS NOT HER
E":RETURN
470 IF CT=3 OR CT=11 THEN PRINT"DON'T BE
 SILLY,IT'S TOO HEAVY.":RETURN
480 IF L(17)=3 THEN ST=25:PLAY"L15O3CO1C
O3CO1CO3CO1CO3CO1C":PRINT"THE ALARMS HAV
E SOUNDED, YOU DO NOT HAVE MUCH TIME.":L
(17)=0:RETURN
490 IF L(CT)=L THEN L(CT)=0:PRINT"YOU AR
E CARRYING THE ";O$(CT):RETURN
500 IF L=3 AND CT=17 THEN PRINT"THEY ARE
 OUT OF REACH":RETURN
510 REM*****DROP*****
520 DV=0:FOR CT=1 TO 54 STEP 3:IF LEFT$(
CT$,3)=MID$(O$,CT,3) THEN DV=CT
530 NEXT:CT=(DV+2)/3:IF DV=0 THEN T=100:
RETURN
540 IF S=0 THEN PRINT"YOU HAVE NOTHING T
O DROP":RETURN
550 IF L(CT)<>0 THEN PRINT"YOU DON'T HAV
E IT TO DROP.":RETURN
560 L(CT)=L:PRINT"YOU HAVE DROPPED THE "
;O$(CT):RETURN
570 IF L(2)<>0 THEN PRINT"IT'S TOO DARK
TO READ!":RETURN
```

```
580 IF L(4)<>L AND L(17)<>L AND L(18)<>L
 THEN PRINT"THERE IS NOTHING TO READ!"
590 IF L(4)=L THEN PRINT"READS:telvsrite
ahomhgoeaenhwsro":PRINT:PRINT"IT'S WRITT
EN IN CODE. YOU NEED  THE COMPUTER TO DE
CODE IT."
600 IF L(18)=L THEN PRINT"DON'T WASTE YO
UR TIME, THEY ARE THE COMPANY'S ACCOUNTS
,EXTREMELYBORING."
610 IF L(17)=L THEN PRINT"your job is to
 steal them not   read them."
620 PRINT:PRINT"PRESS ENTER TO CONTINUE"
:INPUT L$:RETURN
630 IF (L(8)=0 OR L(13)=0) AND (CT$="LAD
" OR CT$="STO") AND L=3 THEN PRINT"YOU C
AN REACH THE SECRET-PAPERS NOW.":L(17)=3
:RETURN
640 IF L=6 THEN PRINT"YOU CANNOT CLIMB O
VER THE FENCE":RETURN
650 IF L=6 AND L(6)=35 AND L(9)=0 AND CT
$="FEN" THEN PRINT"THE FENCE HAS BEEN CU
T, YOU CAN GET OUT BY GOING NORTH.":P(6,
1)=7:RETURN
660 IF L=6 AND L(6)<>35 AND L(9)=0 AND C
T$="FEN" THEN PRINT"YOU CUT THE FENCE BU
T FORGOT IT WAS electrocuted!!!":SOUND 2
45,20:GOTO 1300
670 RETURN
680 IF L=20 AND CT$="MONEY" AND L(19)=0
THEN PRINT"THE MACHINE ACCEPTS THE NOTES
   AND CHANGES THEM INTO A SINGLE  COIN!
```

```
":O$(19)="COIN":MID$(O$,28,3)="COI":RETU
RN
690 PRINT"THIS MACHINE CHANGES THE NOTES
  THAT YOU ALREADY OWN INTO COINS it doe
s nothing else":RETURN
700 DDV=0:FOR M=1 TO 4:IF P(L,M)=-1 THEN
 DDV=1
710 NEXT M:IF DDV=1 AND L(14)=0 THEN KY=
KY+1:IF KY=3 THEN KY=0:L(14)=37:PRINT"TH
E KEY WAS OLD AND YOU USED IT TOO MUCH,
IT HAS CRUMBLED INTO  DUST":RETURN
720 IF DDV<>1 OR L(14)<>0 OR CT$<>"DOO"
THEN 790
730 PRINT"WITH A CREAK,THE DOOR OPENS...
":PLAY"V31L120CDC"
740 IF L=19 THEN P(19,2)=18:P(18,1)=19
750 IF L=2 OR L=27 THEN P(2,1)=27:P(27,2
)=2
760 IF L=16 OR L=28 THEN P(28,2)=16:P(16
,1)=28
770 IF L=21 OR 22 THEN P(22,3)=21:P(21,4
)=22
780 RETURN
790 IF L=19 THEN PRINT"YOU ARE TRAPPED I
NSIDE THE VAULT":IF L(14)<>0 THEN 1300
800 IF DDV<>1 THEN PRINT"THERE IS NO DOO
R TO UNLOCK FOOL!":RETURN
810 IF L(14)<>0 THEN PRINT"YOU DON'T HAV
E THE KEY":RETURN
820 T=100:RETURN
830 IF GS=1 AND G1=L AND L(1)=0 AND LEFT
```

```
$(CT$,2)="GU" THEN 890
840 IF G1<>L THEN PRINT"THERE IS NO GUAR
D HERE!"
850 IF L(1)<>0 THEN PRINT"YOU HAVE NOTHI
NG TO FIRE WITH"
860 IF LEFT$(CT$,2)<>"GU" THEN PRINT"SHO
OT WHAT?"
870 IF L(1)=0 AND G1=L AND LEFT$(CT$,2)=
"GU" AND GS<>1 THEN PRINT"YOU KILLED HIM
 BUT THE GUNSHOTS HAVE ACTIVATED THE SON
IC ALARM. YOU HAVE LITTLE TIME TO ESCAPE
":ST=15:PLAY"O3BCBCBCBCBC":G1=RND(28)+2:
IF G1=L OR G1=28 THEN G1=G1-1
880 RETURN
890 IF RND(3)<2 THEN PRINT"YOU MISSED HI
M":RETURN
900 PRINT"YOU KILLED HIM";G1=RND(28)+2:I
F G1=L OR G1=28 OR G1=7 THEN 900
910 RETURN
920 PRINT"TIME PASSES...":FOR TT=1 TO 20
00:NEXT:RETURN
930 IF LEFT$(CT$,3)<>"COF" OR L<>4 THEN
PRINT"DRINK WHAT?":RETURN
940 IF L=4 AND L(7)<>0 THEN PRINT"YOU HA
VE NOTHING TO DRINK FROM":RETURN
950 PRINT"YUK! IT TASTES AWFUL,BUT WHAT
DOYOU EXPECT FROM A MACHINE...":RETURN
960 IF L(2)<>0 THEN PRINT"IT'S TOO DARK
TO LOOK AROUND OR SEARCH":RETURN
970 IF L=3 THEN PRINT"THE SECRET-PAPERS
ARE ON THE TOPSHELF":RETURN
```

```
980 IF L=21 THEN PRINT"THERE IS A COIN L
OCK ON A TOILETDOOR":RETURN
990 IF L=2 THEN PRINT"THERE IS A LOCKED
DOOR WESTWARDSIT IS CONTROLLED VIA A CEN
TRAL  COMPUTER, THE ONLY WAY TO OPEN  IT
 IS TO CALCULATE IT'S CODE.":RETURN
1000 IF L=28 THEN PRINT"THERE IS A SIGN
SAYING TO GO UP ENTER 'GO N' ONCE UP, TO
 GO DOWN, ENTER 'GO S'.    (TYPICAL
BUREAUCRATIC LANGUAGE!)":RETURN
1010 PRINT"I'M LOOKING, BUT THERE'S LITT
LE TO SEE.":RETURN
1020 IF L(6)=0 THEN PRINT"THEY FIT YOU P
ERFECTLY":G$="YOU ARE WEARING GLOVES":L(
6)=35:RETURN
1030 PRINT"YOU ARE WEARING ALL THAT YOU
OWN":RETURN
1040 IF L(5)=0 AND RIGHT$(CT$,4)="MEMO"
THEN PRINT"THE MEMO READS...":FOR T=1 TO
 1500:NEXT T:PRINT"THE GLOVES ARE IN THE
 WASHROOM":RETURN
1050 IF L(5)=0 AND L=2 THEN PRINT"THE AN
SWER TO THE DOOR CODE HAS BEEN FOUND, JU
ST ENTER 'GO WEST'":P(2,4)=3:P(3,3)=2:RE
TURN
1060 PRINT"HELP YOURSELF, NO CLUES FROM
ME":RETURN
1070 PRINT:INPUT"HAVE YOU REALLY HAD ENO
UGH";A$
1080 IF LEFT$(A$,1)="Y" THEN PRINT"OK YO
U HAVE GIVEN UP...coward":SOUND 1,16:END
```

```
1090 RETURN
1100 IF LEFT$(CT$,3)="GUN" AND MID$(N$,5
,8)="SILENCER" AND L(12)=0 AND L(1)=0 TH
EN GS=1:PRINT"SILENCER ATTACHED, YOU CAN
 NOW  FIRE QUIETLY.":RETURN
1110 IF LEFT$(CT$,3)="LOC" AND MID$(N$,5
,4)="COIN" AND CC=1 AND L=21 THEN PRINT"
THE DOOR SWINGS OPEN, REVEALING THE GLOV
ES":L(6)=21:RETURN
1120 PRINT"YOU CAN'T PUT THAT THERE!":RE
TURN
1130 IF G1=L THEN 1150
1140 PRINT"THERE'S NO ONE NEAR YOU":RETU
RN
1150 INPUT"WITH WHAT";W$:IF LEFT$(W$,4)=
"WIRE" AND L(9)=0 AND RND(2)=1 THEN PRIN
T"THAT GOT HIM!":G1=0
1160 IF W$="ROPE" AND L(16)=0 AND RND(2)
=1 THEN PRINT"YOU FINISHED HIM OFF!":G1=
0
1170 IF LEFT$(W$,4)<>"WIRE" AND W$<>"ROP
E" THEN 1200
1180 IF G1=0 THEN G1=RND(28)+2:IF G1=28
OR G1=L OR G1=7 THEN 1180
1190 RETURN
1200 PRINT"YOU WILL HAVE TO FIGHT WITH Y
OURHANDS...":IF RND(5)=1 THEN PRINT"YOU'
VE KILLED HIM":G1=0:GOTO 1180
1210 RETURN
1220 PRINT:PRINT"YOU WERE 100% SUCCESSFU
L IN YOURMISSION AND THE ORGANISATION HA
```

```
SKINDLY LET YOU KEEP THE OBJECTS THAT YO
U COLLECTED..."
1230 FOR T=1 TO 18:READ A(T):NEXT T
1240 FOR T=1 TO 18:IF L(T)=0 THEN PRINT
O$(T);" WORTH":PRINT A(T);"POUNDS."
1250 NEXT T
1260 PRINT"except the secret papers"
1270 FOR T=1 TO 5000:NEXT T
1280 FOR T=1 TO 8:PRINT:PLAY"O3L8C":NEXT
 T:PRINT TAB(12);"WELL DONE":FOR T=1 TO
8:PLAY"O4L8C":PRINT:NEXT T:END
1290 PRINT:PRINT"YOU'RE SAFELY AWAY BUT
YOU DID  NOT STEAL THE SECRET PAPERS. TH
EORGANISATION ARE NOT PLEASED... YOU HAV
E BEEN SENT BACK TO TRY  AGAIN.":FOR T=1
 TO 4000:NEXT T:RUN
1300 FOR T=1 TO 4000:NEXT T:FOR T=1 TO 1
6:PRINT:NEXT
1310 PRINT"  00000   00000   00000   0
  ":PLAY"L16O2G"
1320 PRINT"  0       0   0     0     0":
PLAY"D"
1330 PRINT"  0000    00000     0     0":
PLAY"O1A"
1340 PRINT"  0       0   0     0     0":
PLAY"E"
1350 PRINT"  0       0   0   00000   000
00":PLAY"C"
1360 PRINT:PRINT:GOTO 1310
1370 REM****COMMAND HANDLING****
1380 T=1:IF LEN(N$)<3 THEN T=100:RETURN
```

```
1390 B$=MID$(C$,T,3):IF B$=LEFT$(N$,3) T
HEN T=(T+2)/3:GOTO 1420
1400 IF T>97 THEN 1430
1410 T=T+3:GOTO 1390
1420 FOR CT=LEN(N$) TO 1 STEP-1:IF MID$(
N$,CT,1)=" " THEN CT$=MID$(N$,CT+1,3) EL
SE NEXT CT
1430 RETURN
1440 REM***INITIALISATION***
1450 DIM O$(18),L$(30),P(30,4),L(18),A(1
8)
1460 FOR T=1 TO 18:READ O$(T):NEXT
1470 FOR T=1 TO 30:READ L$(T):NEXT
1480 FOR T=1 TO 18:READ L(T):NEXT
1490 DATA GUN,TORCH,COFFEE-MACHINE,SCRIB
BLED-MEMO,POCKET-COMPUTER,GLOVES,CUP,STO
OL,WIRECUTTERS
1500 DATA MONEY,COIN-MACHINE,SILENCER,LA
DDER,KEY,PAPER-CLIPS,ROPE,SECRET-PAPERS,
PAPERS
1510 DATA THE MAIN ENTRANCE,THE RECEPTIO
N AREA,THE DOCUMENT ROOM,THE STAFF CANTE
EN,THE DELIVERY ROOM
1520 DATA "A CORNER OF THE YARD NEXT TO 
AN ELECTROCUTED FENCE","THE ALLEYWAY BEH
IND THE COMPLEX, A CAR IS READY TO TAKE 
YOU AWAY"
1530 DATA THE COMPUTER ROOM,THE MAINTAIN
ANCE AREA,A DARK CORRIDOR,THE RECORDS DE
PT,AN UNTIDY OFFICE
1540 DATA THE TYPING POOL,THE RECREATION
```

```
 ROOM,A NARROW CORRIDOR,A PLUSH CONFEREN
CE    ROOM,AN UNLIGHTED OFFICE
1550 DATA THE STRONGROOM NEXT  TO THE VA
ULT,THE STEEL VAULT,THE ACCOUNTS DEPT,TH
E EXECUTIVE WASHROOMS
1560 DATA "A COLD, BARE ROOM",A DARK COR
RIDOR,"A CLEAN, NEW STOREROOM",A LARGE C
UPBOARD,A BRIGHT HALLWAY
1570 DATA THE TELEX ROOM,A LIFT,THE DIRE
CTORS         BOARDROOM,A SPACIOUS OFFICE
1580 DATA 0,2,4,27,8,33,25,13,9,19,20,30
,5,10,13,11,34,11
1590 D$="NORTHSOUTHEAST WEST "
1600 C$="GO MOVTAKGETSTEDROREACLICUTCHAO
PEUNLFIRSHOWAIDRISEALOOWEACOMCALDECHELCL
UQUIPUTHITFIGATTKIL"
1610 O$="GUNTORCOFMEMCOMGLOCUPSTOWIRMONC
OISILLADKEYCLIROPSECPAP"
1620 P$="PARDON? WHAT?    RUBBISH!REPHRAS
E"
1630 L=1:L(1)=0:K=0:ST=10000:KY=0:GS=0
1640 G1=RND(28)+2:IF G1=28 OR G=7 THEN 1
640
1650 DC=RND(6)+4
1660 FOR X=1 TO 30:FOR Y=1 TO 4:READ P(X
,Y):NEXT:NEXT
1670 DATA 2,0,0,0,-1,1,4,0,0,0,2,0,0,10,
9,2,0,15,6,0,0,0,5,0,0,0,0,0,15,24,0,21,
20,8,13,9
1680 DATA 4,0,9,1,0,27,14,12,17,0,11,16,
16,0,0,9,26,0,18,11,5,8,0,21,-1,23,12,26
```

```
,22,12,15,0
1690 DATA 19,20,18,14,0,-1,0,0,18,9,0,0,
2,17,8,-1,0,17,-1,28,16,23,0,17,8,0,26,1
2,0,0,0,27
1700 DATA 6,14,0,24,11,-1,25,13,29,16,0,
0,28,0,0,30,0,0,29,0
1710 DATA 80,10,0,1,200,8,3,10,22,0,0,10
,40,10,1,27,1000000,2
1720 RETURN
1730 CLS:NN$="*THE NIELSON PAPERS*"
1740 PRINT @ 130,"A FULL SCALE TEXT ADVE
NTURE":PRINT @ 364,"C.1983."
1750 FOR F=1 TO 3
1760 FOR T=1 TO 20:PRINT @ 228,MID$(NN$,
T,21-T):PRINT @ 249-T,LEFT$(NN$,T)
1770 PLAY"P10":NEXT:NEXT
1780 PRINT @ 228," ";:FOR T=1 TO 1000:NE
XT T:RETURN
```

CHAPTER 4

OTHER FORMS OF ADVENTURES

So far, we have only considered the more standard type of adventures. If you remember from the first chapter, I emphasised that there were many extremely varied styles and formats of adventure. This chapter attempts to cover these other no less popular adventures and includes full implementations of two different types.

A good starting point, if you are new to adventure writing, is to start writing a grid-based adventure game. These types of adventure are very popular and some commercial software claiming to be 'full scale adventures in the classic mould' are, in fact, 'beefed-up' grid games. Grid games are generally less versatile and less challenging to play, though they can still provide great entertainment. A hybrid game using a grid basis, but including a list of commands similar to 'The Nielson Papers', would provide an exciting programming and adventure-playing challenge.

Let's look a little more closely at what a grid game actually entails. The most usual way for a grid to be formed is for it to be dimensioned in an array. In our grid game, 'The Golden Chalice', there are 1ØØ different locations; therefore, the array containing the locations is dimensioned with 1ØØ. A number of the elements in the array are then given certain values. These differing values are, in fact, the treasure, monster and pot-hole squares. All other squares not given a special value are left blank and are, in fact, empty squares that you can move easily between.

Each type of obstruction/object is given a certain value unique to that type of obstruction/object; for example, in 'The Golden Chalice', all the squares containing monsters are given the value '1Ø9'. In my game, the figure given for each item is the CHR$ of the symbol that represents that item on the map display.

Once you have dimensioned an array and given a value to each square, it is a simple matter of writing routines to handle each square. You will firstly have a number of IF . . . THEN statements telling the computer that if the square the player is standing on contains a monster (ie. in my game, if the value is equal to 1Ø9) then go to the special subroutine that deals with monsters. If you have a grid game with monsters, treasure, magic spells and quicksand, then you would have only four subroutines of this nature to write; obviously many games are larger and more complex than this thus demanding more subroutines to be worked out.

Each subroutine can be written separately allowing you to structure and break down your program into neat and tidy blocks of code (as mentioned in the first chapter). Once done, the next stage to aim at is the ability of the player to move from square to square (location to location).

The best way to allow movement in a grid game is to imagine the playing grid as a flat square, numbered along each row. In this example, I have used a ten by ten grid; this is not the only possible case - any size grid within your computer's memory requirements is possible. A diagram of the grid is shown on the next page.

If you press the key corresponding to rightwards movement on the grid then one is added to the position of the adventurer's character moving you rightwards. Looking at the diagram above, it is quite simple to see that to move left one is deducted, to move up/north a value of ten is deducted, while ten is added to the adventurer's position if you request to move downwards/southwards.

With these pieces written, it is now necessary to add all the 'frills' to the program. Lines will be needed to stop a player

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

moving off the boundaries of the grid, and 'win' and 'lose' routines need to be developed. All the other 'niceties' are really up to the person writing the adventure, but the more common ones are a map showing all the squares and what is contained on them, a title page, a timer feature, a list of possessions that can be acquired by going around the grid and probably the most important 'frill' of all, some sort of objective to the game, whether it is to accumulate as much treasure as possible or to find the secret code to the IBM mainframe, to give just two examples.

One can see, therefore, how this form of adventure satisfies all the conditions of an adventure I put forward in the first chapter.

Another point to note concerns how the different objects and obstructions are placed in at the beginning of the game. You can either place them randomly as I have done in 'The Golden Chalice' or by using DATA statements, keep them rigidly in one place all the time – it really is just a matter of your own preference. In this type of adventure, having a random element of some considerable influence on the game-play is perfectly permissible.

Now, moving on specifically to my grid game 'The Golden Chalice'. You are stuck in the Caves of Delirium somewhere on the planet Zarg, in the Methusian sector. Your aim is to find the fabled Golden Chalice and then climb out of the caves via one of the many openings in the cave roof. Sounds easy, huh? Well, there's a lot more to it than that. You have around 4Ø hours to find the chalice and must avoid the many tormented monsters that are also trapped in these unnatural caverns. There are many other features, just two of which I will tell you about.

Firstly, you need four pieces of rope to climb out of the openings in the cave roof. Secondly, if you are carrying the map, then press the 'M' key and you will catch a glimpse of the cavern system but lose three hours as a penalty. To move, use the compass points N, S, E and W, remembering that you are placed near the centre of the cave system.

Oh, just one final point. If you find a torch then you will be given your cave situation for the next few moves before the torch burns out. You may find that you have some trouble decoding the map as all the various objects and obstructions are represented by single characters. Tough! I will leave you to sort out the map as yet another problem to confront while trying to find 'The Golden Chalice'.

```
1Ø REM*******THE GOLDEN CHALICE*******
2Ø DIM A(1ØØ):P=Ø:C=P:MK=P:SC=P:M=P
3Ø FOR T=1 TO 1ØØ:A(T)=RND(1ØØ):IF A(T)<
56 THEN A(T)=46
```

```
40 IF A(T)>55 AND A(T)<66 THEN A(T)=63
50 IF A(T)>65 AND A(T)<76 THEN A(T)=109
60 IF A(T)>75 AND A(T)<86 THEN A(T)=42
70 IF A(T)>85 AND A(T)<96 THEN A(T)=32
80 IF A(T)>95 AND A(T)<101 THEN A(T)=48
90 NEXT T
100 FOR T=1 TO 5:A(RND(100))=41:NEXT T
110 FOR T=1 TO 100:IF A(T)=41 THEN C=C+1
120 NEXT T:IF C<4 THEN C=0:GOTO 100
130 MP=RND(100):A(MP)=35:TM=43:C=0
140 GC=RND(100):A(GC)=36:X=30+(RND(3)*10
)+(RND(2)+4)
160 GOSUB 400
170 GOTO 240
180 INPUT"WHAT NOW";N$
190 IF N$="M" AND A(MP)=46 THEN GOSUB 40
0
200 IF N$="N" AND X>10 THEN X=X-10
210 IF N$="S" AND X<90 THEN X=X+10
220 IF N$="E" AND X/10<>INT(X/10) THEN X
=X+1
230 IF N$="W" AND (X-1)/10<>INT((X-1)/10
) THEN X=X-1
240 CLS:PRINT">>>>>>>THE GOLDEN CHALICE<
<<<<<<"
250 IF M>9 AND C<6 THEN PRINT"CAVE NUMBE
R ";X
260 PRINT"YOU HAVE";TM;"HOURS LEFT.":PRI
```

```
NT
270 PRINT"YOU ARE"
280 SOUND 180,1:IF A(X)=63 THEN 440
290 IF A(X)=41 THEN 470
300 IF A(X)=35 THEN 510
310 IF A(X)=36 THEN 550
320 IF A(X)=48 THEN 600
330 IF A(X)=109 THEN 630
340 IF A(X)=42 THEN 720
350 IF A(X)=32 THEN 760
360 IF RND(2)=1 THEN PRINT TAB(7)"IN AN
EMPTY CAVE." ELSE PRINT TAB(7)"ON A CLEA
R LEDGE."
370 C=C+1:TM=TM-1:IF TM<1 THEN TM=0:GOTO
 810
380 PRINT
390 GOTO 180
400 REM*****MAP DISPLAY*****
410 CLS:PLAY"L12DEC":A(X)=72:PRINT:PRINT
420 FOR T=1 TO 100:PRINT CHR$(A(T));:IF
T/10=INT(T/10) THEN PRINT
430 NEXT T:FOR T=1 TO 2000:NEXT T:TM=TM-
3:A(X)=46:RETURN
440 REM******TORCH******
450 PRINT TAB(7)"NEXT TO A BURNING TORCH
."
460 PRINT"THIS WILL HELP YOUR EXPLORATIO
N":C=0:A(X)=46:M=M+10:GOTO 370
```

```
470 REM*******ROPE*******
480 PRINT TAB(7)"IN A CAVE WITH SOME":PR
INT"ROPE ON THE FLOOR."
490 A(X)=46:P=P+1:IF P>3 THEN PRINT"YOU
HAVE ALL THE ROPE!"
500 GOTO 370
510 REM******MAP******
520 PRINT TAB(7)"STANDING NEXT TO A MAP
OFTHE CAVES OF DELIRIUM."
530 PRINT:PRINT"IF YOU EVER WISH TO SEE
THE MAP,ENTER 'M' AND YOU WILL RECIEVE A
QUICK GLANCE AT IT. HOWEVER THISCOSTS YO
U 3 HOURS IN TIME."
540 A(MP)=46:PLAY"O3L4CDEFGFG":GOTO 370
550 REM****GOLDEN CHALICE****
560 PRINT TAB(7)"IN THE TREASURE ROOM":P
LAY"O2L8CEDFGABB"
570 PRINT"BESIDE YOU, THE CRYSTAL STATUE
":PLAY"O3L8CEDFGABB"
580 PRINT"IN IT'S HAND,THE GOLDEN CHALIC
E!":PLAY"O4L8CEDFGABB"
590 A(GC)=46:GOTO 370
600 REM*****FLOOD*****
610 PRINT TAB(7)"UP TO YOUR NECK IN WATE
R!":PRINT"THE FLOOD HAS COME.":PLAY"O1L1
0BAGFEDCC"
620 PRINT"YOU MANAGE TO SWIM TO SAFETY B
UTYOU WASTE 10 HOURS.":TM=TM-10:GOTO 370
```

```
630 REM******MONSTER******
640 RESTORE:FOR A=1 TO RND(5):READ M$:NE
XT
650 DATA SERPENT,BACKBREAKER,GRIZZLY,GIA
NT RAT,WOLFMAN
660 PRINT TAB(7)"IN deep trouble. YOU HA
VEAWOKEN THE ";M$
670 PRINT"YOU MUST EITHER FIGHT, OR FLEE
  ENTER YOUR CHOICE (1=FIGHT, 2=FLEE)."
680 INPUT F:IF F=2 THEN X=X+RND(5)-RND(5
):TM=TM-6:PRINT"YOU'RE SAFE FOR THE MOME
NT":GOTO 370
690 R=RND(3):IF R=1 THEN PRINT"YOU KILLE
D HIM WITH NO EFFORT   WHATSOEVER! WELL
DONE":PLAY"L8O4CDEFGCDEFGCDEFGABB":TM=TM
+10:MK=MK+1:GOTO 370
700 IF R=2 THEN PRINT"AFTER A FIERCE BAT
TLE, HE STARTSTO RETREAT. YOU HAVE WON!"
:TM=TM-3:PLAY"O3L4CED":MK=MK+1:GOTO 370
710 PRINT"THE BATTLE WAS HARD. YOU FOUGH
T BRAVELY, BUT THE ";M$;" WON":PLAY"O1L4
EDC":GOTO 890
720 REM*******QUICKSAND*******
730 PRINT TAB(7)"IN QUICKSAND AND SINKIN
G":PRINT"FAST!!!"
740 FOR T=50 TO 1 STEP -1:SOUND T,1:NEXT
750 SOUND 1,8:GOTO 890
760 REM******OPENING******
```

```
770 PRINT TAB(7)"STANDING BELOW A SHAFT
OFLIGHT, THERE LIES AN OPENING!!!"
780 PLAY"O4L100CDEFGGFEDC"
790 IF P=4 THEN GOTO 970
800 GOTO 370
810 REM******OUT OF TIME******
820 PMODE 4,1:PCLS:SCREEN 1,1:CIRCLE(128
,96),80
830 CIRCLE(128,96),20:LINE(208,96)-(193,
96),PSET:LINE(48,96)-(63,96),PSET
840 LINE(128,176)-(128,161),PSET:LINE(12
8,16)-(128,31),PSET
850 LINE(128,96)-(118,20),PSET:PLAY"L40O
3CP1":LINE(128,96)-(118,20),PRESET
860 LINE(128,96)-(128,16),PSET:PLAY"L40O
1CP1"
870 FOR T=1 TO 1000:NEXT T
880 CLS 0:PRINT @ 232,"TIMES UP!";:FOR T
=1 TO 1000:NEXT T
890 REM******DEFEAT*******
900 CLS:PRINT:PRINT"YOU ARE DEFEATED, TH
E CAVES MAKEAN IDEAL BURIAL CHAMBER FOR
YOU AND THE OTHERS THAT HAVE FAILED."
910 IF A(GC)=46 THEN SC=SC+60
920 SC=SC+(15*P)+(10*MK)+TM
930 PRINT:PRINT"YOUR EXPLORATION LEVEL I
S ";SC
940 PLAY"L200O1CDEFGABO2CDEFGABO3CDEFGAB
```

```
O4CDEFGAB"
950 PLAY"BAGFEDCO3BAGFEDCO2BAGFEDCO1BAGF
EDCCCC"
960 PLAY"P4":GOTO 940
970 REM******VICTORY******
980 PLAY"O3L4GP2":CLS:PRINT:PRINT"++++++
+++++WELL DONE+++++++++++"
990 PRINT:PRINT"YOU MANAGED  TO ESCAPE F
ROM THE CAVES OF DELIRIUM."
1000 IF A(GC)<>46 THEN 1020
1010 PRINT"AND WITH THE PRICELESS GOLDEN
   CHALICE.":PLAY"L20O4CDEFGGFGABBBB"
1020 SC=SC+300:PLAY"L20O4CEDDP2":PRINT:P
RINT:GOTO 910
```

An area of adventure gaming that we have not yet discussed so far is the graphic adventure. On a machine such as the Dragon 32, the possibilities for a high quality adventure played entirely in one of the graphics modes are very good and indeed, there are many commercial adventures of this type. Generally, graphic adventures are real-time games using INKEY$ rather than INPUT for the adventurer to enter directions and actions.

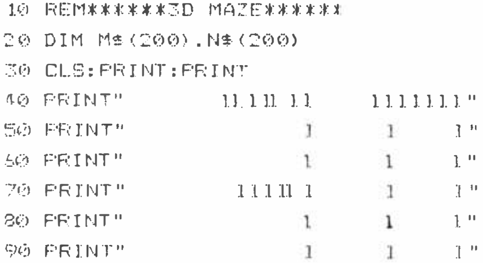
This chapter contains one area of graphics adventuring, the 3D maze game, with a program to illustrate this form of programming simply called '3D MAZE'. The object of the game is very simply to find the treasure, collect it and then make your way out through the exit. The interest in the game is generated by the graphics and movement around the maze.

To move around the maze, you must use the '1' and '2' keys for left and right respectively, the 'O' key to go forward and the 'L' key to turn around 180 degrees. The maze walls and exits will be displayed on-screen in perspective. At the beginning of the game, if you turn around you will see the exit and that is where you must aim for once you have acquired the riches hidden somewhere in the depths of the maze. You will need to play for a short while to get the idea of the movement and perspective. The number of recognised moves is stored and displayed at the end of the game should you succeed. The treasure is in one of seven random positions and its precise location will be decided at the start of the game.

From a programming angle, this program is very interesting to look at. Over half of the program is data. There are two basic chunks of data: the first major block, starting at line 680, contains the information about exits for the DRAW commands to interpret; and the second block of data, gives all the location links, ie. if you turn around 180 degrees at location '6' you will be at location '3'. There are 100 locations, but for each one there are two directions to look at them from (hence the 200 items of data in each block). The key presses are interpreted by lines 260 and 280.

I'm sure you will enjoy this excellent program, which was originally written for the ZX Spectrum by Scott Vincent of Ashford, Middlesex. Scott is a talented programmer and a very patient character - he had to be as we spent a long and often frustrating time converting this program. It was worth it and I'm sure you will agree once you've spent some time delving in the depths of the maze.

This concludes our discussion on other forms of adventures.



```
10 REM*******3D MAZE******
20 DIM M$(200).N$(200)
30 CLS:PRINT:PRINT
40 PRINT"       11.1 11 11   1111111"
50 PRINT"              1     1    1"
60 PRINT"              1     1    1"
70 PRINT"        11111 1     1    1"
80 PRINT"              1     1    1"
90 PRINT"              1     1    1"
```

```
100 PRINT"      1111111    1111111":PRI
NT:PRINT:PRINT TAB(14);"maze"
110 FOR T=1 TO 500:NEXT T
120 FOR X=1 TO 200:READ M$(X):NEXT
130 FOR X=1 TO 200:READ N$(X):NEXT
140 K=0:L=1:TR=0:X=RND(7)
150 LA=-201*(X=1 OR X=4 OR X=6 OR X=7)-8
1*(X=2)-127*(X=3)-169*(X=5)
160 LB=-201*(X=4 OR X=6)-100*(X=1)-84*(X
=2)-122*(X=3)-172*(X=5)-196*(X=7)
170 LC=-94*(X=1)-98*(X=2)-124*(X=3)-131*
(X=4)-117*(X=5)-191*(X=6)-199*(X=7)
180 PMODE 4,1:COLOR 0,1:PCLS:SCREEN 1,1
190 PCLS:FOR X=1 TO 9:IF MID$(M$(L),X,1)
<>"0" THEN 210
200 ON X GOSUB 430,450,400,470,490,400,5
10,530,400
210 NEXT X
220 FOR X=1 TO 9:IF MID$(M$(L),X,1)<>"1"
 THEN 240
230 ON X GOSUB 440,460,550,480,500,590,5
20,540,630
240 NEXT X
250 IF L=2 OR L=LA OR L=LB OR L=LC THEN
GOSUB 330:IF L=2 AND TR=1 THEN 140
260 A$=INKEY$:IF A$="" THEN 260
270 PLAY"V31L240O1CD"
280 P=-VAL(LEFT$(N$(L),3))*(A$="1")-VAL(
MID$(N$(L),4,3))*(A$="0")-VAL(MID$(N$(L)
,7,3))*(A$="2")-VAL(RIGHT$(N$(L),3))*(A$
="L")
```

```
290 IF A$="R" THEN 410
300 IF P=0 THEN 260
310 L=P
320 K=K+1:GOTO 190
330 IF L=2 AND TR=0 THEN DRAW"BM100,100:
U8R8L8D4R6L6D4R8BR8E8BL8F8BR8R4U8L4R8BD8
L4BR16U8L4R8BM0,140;R85U100R80D100R90":P
LAY"O3L8CGEFBDD":RETURN
340 IF L=2 THEN CLS RND(8):PRINT @ 230,"
WELL DONE";K;" MOVES";:PLAY"O4L100CDEFGA
BP40":GOTO 340
350 IF L=LA THEN DRAW"BM116,76;C16R22BM1
16,99;R22C0":RETURN
360 IF L=LB THEN DRAW"BM100,60:C1R54BM10
0,115;R54C0":RETURN
370 IF L=LC THEN DRAW"BM72,32;C1R110BM72
,143;R110;C0"
380 IF TR=1 THEN CLS:PRINT @ 225,"YOU AL
READY HAVE THE TREASURE!":PLAY"P1":PMODE
 4,1:COLOR 0,1:SCREEN 1,1:RETURN
390 TR=1:CLS:PRINT @ 227,"YOU HAVE FOUND
 THE TREASURE!":PLAY"O4L8CDEFGABBB":PMOD
E 4,1:SCREEN 1,1:COLOR 0,1:PCLS:RETURN
400 RETURN
410 Y=1
420 REM*******DRAW BLOCK*******
430 DRAW"BM40,0;F32BM40,175:E32":RETURN
440 DRAW"BM24,0;D175U32R48U111L48":RETUR
N
450 DRAW"BM183,32:E32BM183,143;F32":RETU
RN
```

```
460 DRAW"BM231,0;D175U32L48U111R48":RETU
RN
470 DRAW"BM73,33;F27BM73,142;E27":RETURN
480 DRAW"BM73,33;F11D87G11BM85,115;R15U5
5L15":RETURN
490 DRAW"BM155,60;E27BM182,142;H27":RETU
RN
500 DRAW"BM182,142;H11U87E11BM170,115;L1
5U55R15":RETURN
510 DRAW"BM101,61;F19BM101,114;E19":RETU
RN
520 DRAW"BM101,61;F8D37G8BM110,76;R6F4H4
D23L6R6E4":RETURN
530 DRAW"BM154,61;G19BM154,114;H19":RETU
RN
540 DRAW"BM154,61;G8D37F8BM145,76;L6G4E4
D23R6L6H4":RETURN
550 DRAW"BM73,32;R110BM73,143;R110"
560 IF LEFT$(M$(L),1)="0" THEN DRAW"BM72
,33;D109" ELSE DRAW"BM72,33;C1D109C0"
570 IF MID$(M$(L),2,1)="0" THEN DRAW"BM1
83,33;D109" ELSE DRAW"BM183,33;C1D109C0"
580 RETURN
590 DRAW"BM101,60;R54BM101,115;R54"
600 IF MID$(M$(L),4,1)="0"THEN DRAW"BM10
0,61;D53" ELSE DRAW"BM100,61;C1D53C0"
610 IF MID$(M$(L),5,1)="0" THEN DRAW"BM1
55,61;D53" ELSE DRAW"BM155,61;C1D53C0"
620 RETURN
630 DRAW"BM116,99;R23BM116,76;R23C1BM117
,98;E4BM117,77;F4BM138,98;H4BM138,77;G4C
```

```
0"
640 IF MID$(M$(L),7,1)="0" THEN DRAW"BM1
16,77;D21" ELSE DRAW"BM116,77;C1D21C0"
650 IF MID$(M$(L),8,1)="0" THEN DRAW"BM1
39,77;D21" ELSE DRAW"BM139,77;C1D21C0"
660 RETURN
670 REM******DATA BLOCK******
680 DATA"111","","000000010","101","0000
10000","010101","010000011","000010101",
"000010111","000011","000000010","011","
100000000","111","000100000","101","101"
,"001","010101","101","100001","011","00
1","011","101"
690 DATA"100000000","011","010011","0001
10011","011","011","101","100011","101",
"000111","111","011","111","000111","000
011","100000111","010000101","101","111"
,"000101","010101","101","100010101","00
1","000100010"
700 DATA"011","010111","111","011","1000
10000","011","001","101","110011","111",
"011","000111","001","001","110001","110
000111","001","101","110001","111","0000
10100","101","010100101","010101","111"
710 DATA"100101","000010101","010000000"
,"101","100000010","100000001","01010000
0","011","000001","111","111","000000111
","111","100111","000111","000100111","1
11","000000100","001","001","100001","01
0011","001","000010011","010001"
720 DATA"010110000","100000010","1100000
```

```
00","011","000000101","111","000101","00
0111","101","110101","000000111","101","
011","000000110","111","000101","001","1
10000000","100000101","111","011","00000
1"
730 DATA"100011","001","000100011"
740 DATA"100011","010000001","011","101"
,"101","001","111","011","010101","011",
"101","101","101","000000110","011","000
110000","000011","110000000","000000111"
,"100011","000000011","000000110","11000
0000","000110111","000110000"
750 DATA"110111","110011","000011","011"
,"011","011","000011","101","101","111",
"101","011","000100000","011","101","100
000100","101","011","000010001","011","1
01","010001","000111","110100011","111"
760 DATA"000110100","000000110","0000001
00","000010011","100000000","010011","00
0100000","010000010","100000101","000010
000","101","000101","000000010","111","0
11","001","011","101","101","010000000",
"000001","000010000","011","001","000011
"

770 DATA 003000004002","000000000001","0
00005000006","025000000026","00000700000
8","000004002003","000010009011","000006
000005","000052000053","000012000013","0
000800007","000000014015"
780 DATA"009011000010","018000017016","0
00013000012","015000000014","02000000002
```

```
1","000000000019","000017016018","023000
000022","016018000017","000000021020","0
00000000024","000000022023"
790 DATA"028000000027","002003000004","0
00000026025","000030029031","00005900006
0","000000032033","000000027028","035000
000034","029031000030","033000000032","0
00036000037","039000038040"
800 DATA"000000034035","042000043041","0
00053000052","000037000036","04003900003
8","000045044046","054000000055","071000
072070","000047000048","000043041042","0
49000000050","044046000045"
810 DATA"000000000051","000048000047","0
00000050049","000038040039","01100001000
9","000000057056","041042000043","000000
055054","000000000058","056000000057","0
63061064062","031000030029"
820 DATA"000000067066","000060000059","0
00000000069","000000000065","06206306106
4","064062063061","000000000068","066000
000067","061064062063","046000045044","0
00073000074","101000000102"
830 DATA"000076075077","000072070071","1
21000122120","078079000080","00007400007
3","000087088086","081000000082","075077
000076","085084000083","000080078079","0
00000082081","000098000097"
840 DATA"086000087088","079000080078","0
00090000089","084000083085","08808600008
7","000092000091","000089000090","095000
```

```
094093","000091000092","000000000096","0
0000000100","093095000094"
850 DATA"000083085084","000000000099","0
00097000098","000094093095","00010310510
4","070071000072","146144147145","000000
102101","000107000106","104000103105","0
00109000108","000106000107"
860 DATA"110000000111","114112115113","0
00108000109","153000000152","00000011111
0","000150000151","116000117118","000171
000172","000000000119","113114112115","1
18116000117","077000076075"
870 DATA"000000126127","000124000123","1
20121000122","000000000125","00012300012
4","130128000129","000122120121","000000
135134","127000000126","131000000132","0
00000000133","128000129130"
880 DATA"000000132131","000129130128","0
00000136137","139000000138","13400000013
5","137000000136","000141000140","000000
138139","000143000142","000140000141","1
44147145146","000173000174"
890 DATA"105104000103","000142000143","0
00149000148","145146144147","00015100015
0","000148000149","112115113114","115113
114112","000154000155","000000156157","0
00000152153","000000159158"
900 DATA"000155000154","157000000156","1
60000000161","163000164162","15800000015
9","000000161160","000180000181","000000
165166","168000000167","162163000164","1
```

```
6600000000165","00000000169170"
910 DATA"000172000171","000000167168","1
70000000169","000117118116","00017500017
6","147145146144","178000179177","000174
000173","000176000175","000182000183","0
00181000180","177178000179"
920 DATA"000164162163","000184000185","0
00179177178","186187000188","00018300018
2","190000000189","000194000195","000185
000184","187000188186","000000191192","0
00000000193","000000189190"
930 DATA"192000000191","196000000197","0
00188186187","000199000198","00019500019
4","000000197196","000000000200","000198
000199"
940 GOTO 940
```

CHAPTER 5
THE CITY OF ATLANTIS

This adventure game was originally written by Ian Watt. Ian is an experienced adventure programmer who has contributed many of his programs to magazines and is the author of **'Creating Adventures On Your BBC'.** This game was originally written on a BBC Micro and was subsequently converted to the Dragon 32 by myself.

At the start of this adventure; you find yourself imprisoned in the detention centre of the fabled city of Atlantis. Your diving expedition took you too close to the city and the Atlantans think you are a security risk. On your own and guarded by an Atlantan guard, you must find your way back to the diving ship you left some time ago.

Many hazards lie in your path and you find it hard to cope with the thin atmosphere that prevails in this city under the sea. Trust to your luck and skill, and make haste for the power reactor that provides the city with its energy resources is becoming unstable.

After being asked whether you require instructions or not, the computer displays the name of the location, the objects there, the items that you are carrying (the inventory) and then awaits your command. Some commands are only one word, such as 'NORTH', while others involve a verb and a noun such as 'GET FLOWER'. Following on from this piece is a list of all the commands and objects together with their codes. These codes are abbreviated forms of the full words and enable you to enter each command and object quickly. If you have a two word command, first enter the command like 'FIGHT' or 'EXAMINE' and the computer will then prompt you with the word 'OBJECT'. You may then enter the object to be manipulated.

There is a long pause between command operations allowing you to read and note any response that the computer displays. If you wish to speed up the program, just press a key when a pause starts and the program will continue.

This program, while being an adventure of the standard kind, is written in a different style to my own and, therefore, should provide an interesting contrast. There is no right or wrong way to put together an adventure program and both styles achieve the same objective, a playable adventure game.

Here then are the lists of locations, commands and objects that will aid you in your quest.

LOCATIONS:

(1) Detention Area.
(2) Indoctrination Area.
(3) Sports Arena.
(4) Barren Ground.
(5) Blocked Exit From The Arena.
(6) Advanced Part Of The City.
(7) Primitive Part Of The City.
(8) Old Part Of The City.
(9) Garden Walkway.
(10) Food Growing Area.
(11) Central Dome.
(12) Atlantan Lecture Theatre.
(13) Nuclear Fusion Power Reactor.
(14) Water Hydrolysis Area.
(15) Atlantan Radio Station.
(16) Border Of The City.
(17) Passage Out Of The Kingdom.
(18) Slave Labour Area.
(19) Animal Specimen Centre.
(20) Underwater Observation Point.
(21) Forbidden Zone.
(22) Equipment Storage Area.

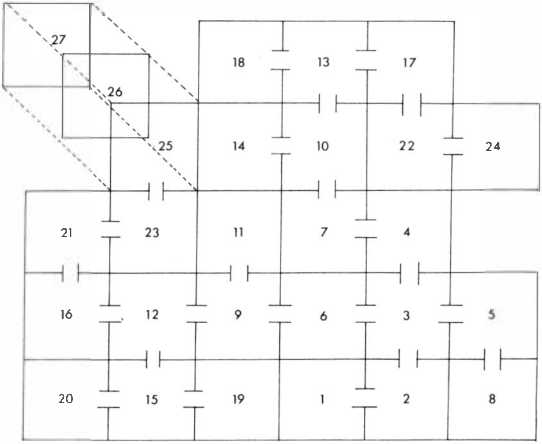
(23) Murky Lagoon.
(24) Air Lock.
(25) Steep Incline.
(26) Cliff Face.
(27) Top Of The Cliff.

COMMANDS:	NORTH	'N'
	SOUTH	'SO'
	EAST	'EAS'
	WEST	'WES'
	GET	'G'
	TAKE	'TA'
	PICK UP	'PI'
	DROP	'D'
	LEAVE	'LE'
	THROW	'TH'
	KILL	'K'
	FIGHT	'FI'
	EXAMINE	'EX'
	FRISK	'FR'
	SEARCH	'SE'
	OPEN	'O'
	PUSH	'PU'
	UNLOCK	'U'
	WEAR	'WEA'
	EAT	'EAT'
	FEED	'FEED'
	INSERT	'I'
	CLIMB	'C'
	QUIT	'Q'
	RAISE	'R'
	PLANT	'PL'
	VAULT	'V'

OBJECTS: The number after the object denotes the starting position of that object. If the number is '-2' then that means that the object is not in a room.

HYDROGEN	'HY'	14
DIVING GEAR	'D'	-2
FUEL SLOT	'FU'	13
LOCKER	'LO'	22
ATLANTAN	'AT'	1
SCIENTIST	'SC'	2
GLADIATOR	'GL'	3
OLD DOOR	'OL'	5
MASTER KEY	'MA'	8
FRUIT	'FR'	10
GORILLA	'GO'	19
SEA MONSTER	'SEA'	21
THICK GLASS	'T'	20
ROPE LADDER	'R'	25
HANDLE	'HA'	27
AIR LOCK	'AI'	24
CHAINS	'C'	18
MICROPHONE	'MI'	15
WARRIOR	'WAR'	16
ORG	'OR'	17
SLEEPY ORG	'SL'	-2
GROUND	'GR'	4
JACKET	'J'	6
FLOWER	'FL'	9
LECTURER	'LE'	12
BOOKLET	'B'	11
SEEDS	'SEE'	-2
POLE VAULT	'P'	-2
WALL	'WAL'	7
METAL ROD	'ME'	23

Here is a plan of the adventure scenario. This will help you greatly in trying to find your way around the adventure. The numbers on the plan correspond to the numbers on the location titles.



Here follows the complete listing of 'The City Of Atlantis' for you to type in.

```
10 REM****************************
20 REM******CITY OF ATLANTIS******
30 REM****************************
40 X=0
50 CLEAR:GOSUB 1520
60 GOSUB 1630
70 PRINT:PRINT:PRINT"DO YOU WANT INSTRUC
TIONS? (Y/N)";:INPUT Z$:IF Z$="Y" THEN 8
0 ELSE 160
80 CLS:PRINT"WHILE ON A DIVING EXPEDITIO
N YOUWERE CAPTURED BY THE PEOPLE FROMTHE
 CITY OF ATLANTIS UNDER THE  SEA AND YOU
 ARE NOW A PRISONER  OF THESE PEOPLE."
90 FOR Z=1 TO 5000:IF INKEY$="" THEN NEX
T Z ELSE Z=5000:NEXT Z
100 CLS:PRINT" YOU HAVE TO DEPEND ON GIV
ING   THE COMPUTER INSTRUCTIONS SO    TH
AT YOU MAY ESCAPE."
110 FOR Z=1 TO 5000:IF INKEY$="" THEN NE
XT Z ELSE Z=5000:NEXT Z
120 CLS:PRINT"ENTER A COMMAND OR AN OBJE
CT AS THE COMPUTER PROMPTS YOU - YOU  WI
LL FIND THAT, FOR EACH WORD   YOU WILL E
NTER THE LEAST NUMBER OF LETTERS THAT WI
LL DETERMINE  ONE WORD FROM ANOTHER."
130 FOR Z=1 TO 5000:IF INKEY$="" THEN NE
XT Z ELSE Z=5000:NEXT Z
140 CLS:PRINT"IF ONE COMBINATION OF COMM
AND & OBJECT DOES NOT WORK THEN TRY   AN
OTHER."
150 FOR Z=1 TO 5000:IF INKEY$="" THEN NE
XT Z ELSE Z=5000:NEXT Z
160 CLS:A=1:W=0:V=0:U=0
170 W=W+1:RESTORE:FOR T=1 TO 1000:IF INK
EY$<>"" THEN T=1000:CLS:NEXT T ELSE NEXT
 T:CLS
180 IF E(6)=-2 AND U<6 THEN U=U+1
190 IF (A=6 OR A=11) AND E(11)=0 THEN PR
INT"THE gorilla IS HUNGRY AGAIN...":GOTO
 720
200 IF W>34 AND E(1)<>-2 THEN PRINT"THE
POWER REACTOR HAS BECOME    UNSTABLE AND
 HAS BLOWN UP!":GOTO 720
```

```
210 IF A=21 AND E(12)=A AND E(11)=0 THEN
 E(11)=-2:E(12)=-2:PRINT"THE gorilla HAS
 RUN AFTER THE   sea monster."
220 IF E(20)=A AND E(24)=-1 THEN PRINT"T
HE FLOWER MAKES THE org SLEEPY":E(20)=-2
:E(21)=A
230 IF E(25)=A AND E(26)=0 THEN E(26)=-2
:E(27)=0:PRINT"THE CRAZY lecturer TAKES
THE    BOOKLET AND GIVES YOU SEEDS."
240 FOR B=1 TO A:READ A$:NEXT B:PRINT @
2,A$
250 PRINT"EXITS:- :";:IF A(A)<>0 THEN PR
INT"NORTH:";
260 IF B(A)<>0 THEN PRINT"SOUTH:";
270 IF C(A)<>0 THEN PRINT"EAST:";
280 IF D(A)<>0 THEN PRINT"WEST:";
290 PRINT:PRINT"OBJECTS:-";
300 H=0:FOR C=1 TO 30:IF E(C)<>A THEN NE
XT C ELSE PRINT":";H$(C):NEXT C
310 PRINT"[-----------------------------
-]"
320 PRINT:PRINT"INVENTORY:-";
330 F=0:FOR C=1 TO 30:IF E(C)<>0 AND E(C
)<>-1 OR F=6 THEN NEXT C ELSE PRINT":";H
$(C):NEXT C
340 PRINT"[-----------------------------
-]"
350 REM
360 PRINT"COMMAND";:D$="":M=0:N=0
370 IF F>1 AND A=26 THEN PRINT"THE ROPE
LADDER HAS BROKEN...":GOTO 720
```

```
380 IF A=16 AND E(17)<>0 THEN PRINT"THE
WARRIOR CATCHES YOU AS YOU  TRY AND ESCA
PE OUT OF THE CITY.":GOTO 720
390 IF A=16 THEN PRINT"THE WARRIOR SEES
YOUR CHAINS,    AND THINKS YOU WILL NOT G
O FAR."
400 IF E(18)=0 THEN PRINT"SOME ATLANTANS
 HEAR YOU OVER    THEIR RADIOS AND INVES
TIGATE...":GOTO 720
410 IF A=21 AND E(12)=A THEN PRINT"THE s
ea monster HAS CAUGHT YOU.":GOTO 720
420 INPUT D$:DD$=LEFT$(D$,3):FOR C=1 TO
29:IF LEFT$(DD$,1)=B$(C) OR LEFT$(DD$,2)
=B$(C) OR DD$=B$(C) THEN M=C:C=29:NEXT C
:GOTO 440
430 NEXT C:GOTO 450
440 IF M>0 AND M<5 OR M=24 OR M=28 OR M=
29 THEN GOTO 490 ELSE GOTO 470
450 PRINT"I'M SORRY, BUT I DO NOT KNOW H
OWTO ";D$;" SOMETHING.":GOTO 170
460 PRINT"I'M SORRY, BUT I DO NOT KNOW
  WHAT A ";D$;" IS.":GOTO 170
470 PRINT"OBJECT?    ";:D$="":N=0
480 INPUT E$:EE$=LEFT$(E$,3):FOR C=1 TO
30:IF EE$=G$(C) OR LEFT$(EE$,1)=G$(C) OR
 LEFT$(EE$,2)=G$(C) THEN N=C:C=30:NEXT C
 ELSE NEXT C490 PRINT C$(M);:IF M>0 AND
M<5 OR M=24 OR M=28 OR M=29 THEN 500 ELS
E PRINT" ";H$(N)
500 PRINT
510 ON M GOTO 590,590,590,590,740,740,74
```

```
0,790,790,790,840,920,850,850,850,960,96
0,960,1050,1100,1130,1170,1210,1240,1250
,1310,1350
520 PRINT"I CANNOT DO THAT":GOTO 170
530 PRINT"I AM ALREADY CARRYING IT":GOTO
 170
540 PRINT"I AM CARRYING TOO MUCH":GOTO 1
70
550 PRINT"O.K.":GOTO 170
560 PRINT"I DO NOT SEE IT HERE":GOTO 170
570 PRINT"I AM NOT CARRYING IT":GOTO 170
580 PRINT"I DO NOT SEE A PLACE TO PUT IT
":GOTO 170
590 IF (M<>3 AND A=1) THEN V=1:PRINT"THE
 atlantan LAUGHS AT YOU       WALKING INT
O THE WALL.":GOTO 170
600 IF M=3 AND A=1 AND V=0 THEN PRINT"TH
E atlantan WILL NOT LET YOU   PASS.":GOT
O 170
610 IF V=1 THEN V=0:PRINT"THE atlantan I
S DISTRACTED AND  DOES NOT SEE YOU LEAVE
."
620 IF E(7)=A THEN PRINT"THE gladiator W
ILL NOT LET YOU  PASS.":GOTO 170
630 IF E(8)=A AND M=2 THEN PRINT"YOU CAN
NOT WALK THROUGH THE OLD DOOR.":GOTO 170
640 IF E(20)=A AND M=2 THEN PRINT"THE cr
g IS IN THE WAY.":GOTO 170
650 IF E(29)=A AND M=1 THEN PRINT"THE WA
LL IS TOO SOLID FOR YOU TOGO THROUGH IT.
":GOTO 170
```

```
660 DV=A
670 IF M=1 THEN A=A+A(A)
680 IF M=2 THEN A=A+B(A)
690 IF M=3 THEN A=A+C(A)
700 IF M=4 THEN A=A+D(A)
710 IF A=DV THEN PRINT"NO EXIT!":GOTO 17
0 ELSE GOTO 170
720 PRINT:PRINT:PRINT"PRESS SPACE TO STA
RT AGAIN"
730 IF INKEY$<>" " THEN 730 ELSE GOTO 50
740 IF N<>1 AND N<>2 AND N<>9 AND N<>10
AND N<>15 AND N<>17 AND N<>18 AND N<>21
AND N<>23 AND N<>24 AND (N<26 OR N>2) AN
D N<>30 THEN 520
750 IF E(N)=0 OR E(N)=-1 THEN 530
760 IF F=6 THEN 540
770 IF E(N)<>A THEN 560
780 E(N)=0:GOTO 550
790 IF E(N)<>0 AND E(N)<>-1 OR N=23 AND
E(24)=-1 AND H>4 THEN 570
800 IF N=11 THEN 520
810 IF H=6 THEN 580
820 IF N=23 AND E(24)=-1 THEN E(24)=A
830 E(N)=A:GOTO 550
840 PRINT"YOU DO NOT HAVE ENOUGH ENERGY
TODO THAT IN THE THIN ATMOSPHERE.":GOTO
170
850 IF E(N)<>A AND E(N)<>0 AND E(N)<>-1
THEN 560 ELSE IF N<>6 AND N<>13 AND N<>1
6 AND N<>22 AND N<>23 THEN PRINT"I SEE N
OTHING WORTH TAKING NOTE OF.":GOTO 170
```

```
860 IF N=22 THEN PRINT"THE GROUND IS PRE
PARED FOR THE  PLANTING OF SOME SEEDS.":
GOTO 170
870 IF N=23 THEN PRINT"THERE IS A BUTTON
 HOLE IN THE   JACKET.":GOTO 170
880 IF N=16 THEN PRINT"TO OPEN THE AIR L
OCK, IT MUST BERAISED.":GOTO 170
890 IF N=13 AND E(N)<>A THEN 560 ELSE IF
 N=13 THEN PRINT"SOME FISH ARE SWIMMING
ABOUT ON THE OTHER SIDE OF IT.":GOTO 170
900 IF E(N)<>A THEN PRINT"I DO NOT SEE H
IM HERE.":GOTO 170
910 E(6)=-2:PRINT"HE GIVES YOU SOMETHING
 TO HELP  YOU COPE WITH THE ATMOSPHERE."
:GOTO 170
920 IF N<>7 THEN 520
930 IF E(N)<>A THEN 900
940 IF U=0 OR U=6 THEN PRINT"YOU WERE NO
 MATCH FOR THE        gladiator.":GOTO 72
0
950 E(7)=-2:PRINT"YOU HAD JUST ENOUGH EN
ERGY TO   DEFEAT THE gladiator.":GOTO 17
0
960 IF N<>4 AND N<>8 OR H=6 THEN 520
970 IF E(N)<>A THEN 560
980 IF N=4 AND M<>17 THEN 1010
990 IF U=6 THEN PRINT"YOU DO NOT HAVE EN
OUGH ENERGY.":GOTO 170
1000 E(N)=-2:PRINT"YOU HAVE OPENED THE O
LD DOOR.":GOTO 170
1010 IF E(9)<>0 THEN PRINT"I CANNOT DO T
```

```
HAT...YET.":GOTO 170
1020 PRINT"THE LOCKER OPENS...AND CLOSES
."
1030 IF E(2)=-2 THEN E(2)=A:PRINT"SOME D
IVING GEAR FELL OUT."
1040 GOTO 170
1050 IF N<>2 AND N<>23 AND N<>24 THEN 52
0
1060 IF E(N)=-1 THEN PRINT"I AM ALREADY
WEARING IT.":GOTO 170
1070 IF E(N)<>0 THEN 570 ELSE IF N=24 AN
D E(23)<>-1 THEN 580
1080 IF N=24 THEN PRINT"THE FLOWER GIVES
 OFF A PUNGENT  ODOUR, BUT DOES NOT HARM
 YOU.":E(N)=-1:GOTO 170
1090 E(N)=-1:GOTO 550
1100 IF N<>10 THEN 520
1110 IF E(N)<>0 THEN 570
1120 E(N)=-2:PRINT"O.K...BUT IT WAS MEAN
T TO BE FORSOME ANIMALS.":GOTO 170
1130 IF N<>11 THEN 520
1140 IF E(N)<>A THEN 560
1150 IF E(10)<>0 THEN PRINT"I HAVE NOTHI
NG TO FEED IT WITH.":GOTO 170
1160 E(10)=-2:E(11)=0:PRINT"THE gorilla
FOLLOWS YOU IN HOPE OF GETTING MORE FOOD
.":GOTO 170
1170 IF N<>1 THEN 520
1180 IF E(N)<>0 THEN 570
1190 IF A<>13 THEN PRINT"I SEE NO PLACE
TO INSERT IT.":GOTO 170
```

```
1200 E(1)=-2:PRINT"THE ADDITION OF MORE
FUEL TO THEREACTOR HAS STABILISED IT.":G
OTO 170
1210 IF N<>14 THEN 520
1220 IF E(N)<>A THEN 560
1230 IF A=26 AND E(15)=0 THEN A=25:E(N)=
A:GOTO 550 ELSE IF A=26 THEN A=27:E(N)=A
:GOTO 550 ELSE IF A=25 THEN A=26:E(N)=A:
GOTO 550 ELSE IF A=27 THEN A=26:E(N)=A:G
OTO 550
1240 PRINT @ 450,"ARE YOU SURE? (Y/N)";:
IF INKEY$="Y" THEN 720 ELSE IF Q$="N" TH
EN 170 ELSE 1240
1250 IF N<>16 THEN 520
1260 IF A<>24 THEN 560
1270 IF E(15)<>0 THEN PRINT"I HAVE NOTHI
NG WITH WHICH I CAN OPERATE THE RAISING
MECHANISM.":GOTO 170
1280 IF E(2)<>-1 THEN PRINT"YOU HAVE DRO
WNED IN THE WATER   THAT HAS FLOODED THE
 AIR LOCK.":GOTO 720
1290 Y=150-W:IF Y>X THEN X=Y
1300 PRINT"YOU HAVE MANAGED TO REACH THE
   SURFACE AND YOUR DIVING SHIP.":PRINT
@ 450,"SCORE=";Y;"    BEST SCORE=";X:GOTO
 720
1310 IF N<>27 THEN 520
1320 IF E(N)<>0 THEN PRINT"I AM NOT CARR
YING THEM.":GOTO 170
1330 IF A<>4 OR H=6 THEN PRINT"I SEE NO
PLACE TO PLANT THEM.":GOTO 170
```

```
1340 E(N)=-2:E(28)=A:PRINT"THE SEEDS GRO
W QUICKLY AND FORM TALL POLE VAULT SHAPE
D PLANTS.":GOTO 170
1350 IF N<>29 THEN 520
1360 IF A<>7 THEN 560
1370 IF E(28)<>0 THEN PRINT"I HAVE NOTHI
NG TO VAULT WITH.":GOTO 170
1380 A=10:PRINT"YOU MANAGE TO VAULT THE
WALL.":GOTO 170
1390 DATA DETENTION AREA,INDOCTRINATION
AREA,SPORTS AREA,BARREN GROUND,BLOCKED E
XIT FROM ARENA,ADVANCED PART OF CITY,PRI
MITIVE PART OF CITY,OLD PART OF CITY,GAR
DEN WALKWAY,FOOD GROWING AREA,CENTRAL DO
ME,ATLANTAN LECTURE THEATRE
1400 DATA NUCLEAR FUSION POWER REACTOR,W
ATER HYDROLYSIS AREA,ATLANTAN RADIO STAT
ION,BORDER OF CITY,PASSAGE OUT OF KINGDO
M,SLAVE LABOUR AREA,ANIMAL SPECIMEN CENT
RE,UNDERWATER OBSERVATION POINT,FORBIDDE
N ZONE,EQUIPMENT STORAGE AREA
1410 DATA MURKY LAGOON,AIR LOCK,STEEP IN
CLINE,CLIFF FACE,TOP OF CLIFF
1420 DATA 0,1,1,0,0,0,3,-3,2,3,0,0,0,0,-
3,5,0,0,0,0,0,-5,2,0,0,0,0
1430 DATA 0,0,-1,-1,3,0,0,0,0,-3,-2,3,-3
,0,0,0,5,0,0,0,-5,0,0,0,-2,0,0
1440 DATA 1,0,2,0,0,-3,-3,0,-3,0,0,-3,4,
-4,4,-4,0,-5,0,-5,2,2,0,0,0,0,0
1450 DATA 0,-1,3,3,-2,3,0,0,3,4,0,4,5,0,
5,0,-4,0,-4,0,0,0,-2,-2,0,0,0
```

```
1460 DATA N,SO,EAS,WES,G,TA,PI,D,LE,TH,K
,FI,EX,FR,SE,O,PU,U,WEA,EAT,FE,I,C,Q,R,P
L,V
1470 DATA NORTH,SOUTH,EAST,WEST,GET,TAKE
,PICK UP,DROP,LEAVE,THROW,KILL,FIGHT,EXA
MINE,FRISK,SEARCH,OPEN,PUSH,UNLOCK,WEAR,
EAT,FEED,INSERT,CLIMB,QUIT,RAISE,PLANT,V
AULT
1480 DATAHY,D,FU,LO,AT,SC,GL,OL,MA,FR,GO
,SEA,T,R,HA,AI,C,MI,WAR,OR,SL,GR,J,FL,LE
,B,SEE,P,WAL,ME
1490 DATA HYDROGEN,DIVING GEAR,FUEL SLOT
,LOCKER,ATLANTAN,SCIENTIST,GLADIATOR,OLD
 DOOR,MASTER KEY,FRUIT,GORILLA,SEA MONST
ER,THICK GLASS,ROPE LADDER,HANDLE,AIR LO
CK,CHAINS,MICROPHONE,WARRIOR,ORG,SLEEPY
ORG,GROUND,JACKET,FLOWER,LECTURER
1500 DATA BOOKLET,SEEDS,POLE VAULT,WALL,
METAL ROD
1510 DATA 14,-2,13,22,1,2,3,5,8,10,19,21
,20,25,27,24,18,15,16,17,-2,4,6,9,12,11,
-2,-2,7,23
1520 REM****ARRAYS****
1530 DIM L$(27):FOR T=1 TO 27:READ L$(T)
:NEXT T
1540 DIM A(27),B(27),C(27),D(27)
1550 FOR T=1 TO 27:READ A(T):NEXT:FOR T=
1 TO 27:READ B(T):NEXT
1560 FOR T=1 TO 27:READ C(T):NEXT:FOR T=
1 TO 27:READ D(T):NEXT
1570 DIM B$(27):FOR T=1 TO 27:READ B$(T)
```

```
:NEXT T
1580 DIM C$(27):FOR T=1 TO 27:READ C$(T)
:NEXT
1590 DIM G$(30):FOR T=1 TO 30:READ G$(T)
:NEXT
1600 DIM H$(30):FOR T=1 TO 30:READ H$(T)
:NEXT
1610 DIM E(30):FOR B=1 TO 30:READ E(B):N
EXT B
1620 RETURN
1630 CLS 3:PRINT @ 108,"CITY  OF";
1640 PRINT @ 224,"";
1650 PRINT"111 111 1   111 1 1 111 111 1
11"
1660 PRINT"1 1  1  1   1 1 111  1   1  1
"
1670 PRINT"111  1  1   111 111  1   1  1
11"
1680 PRINT"1 1  1  1   1 1 111  1   1   
 1"
1690 PRINT"1 1  1  111 1 1 1 1  1  111 1
11"
1700 PLAY"O3L8CEDFGABBACF1"
1710 RETURN
```

CHAPTER 6

OTHER GAMES FOR YOUR COMPUTER

You may ask why I have included other games in an adventure book? Well, there are several reasons. Firstly, an introduction into games writing can show how the computer makes decisions and how to 'weight' the game to make it more entertaining. A few of the techniques used in writing these games can be transferred and used in writing good adventures. Secondly, and perhaps the main reason for their inclusion is that they are really good fun and entertaining to play.

I have kept the comments on each program reasonably brief, with the instructions explained and some of the techniques highlighted.

ALIEN SWARM

This is the one moving graphics game in the book. The game is set in high resolution graphics with the PUT command moving the alien ships around the screen. The instructions are displayed first and when these are fully understood, the game gets under way. I really do not have to tell you any more than that as the game format should be obvious to you once you have read the instructions and have had a practice game.

The program is divided into neat blocks, almost modules, of BASIC code. These have all been labelled with REM statements for extra clarity. Each block is explained below:

(1) The initial title and the question asking you if you want to see the instructions.

(2) Initialisation. All the program's variables are given their starting values, and the graphics needed in the game are defined and stored in GET arrays.

(3) Action. This is the real core of the program. The various ships and bases are moved, the keyboard is checked for a key press, and if it is the fire button, then the program jumps to a separate 'fire' routine. The keyboard is scanned using PEEKs which give a faster response and also allow for key repeats.

(4) Laser Fire. This is the routine that launches the missile, allows it to move while moving all the other pieces, and checks for a 'hit ship'. If a ship is hit then the score is increased, certain musical notes are played and the game goes back to the 'action' section.

(5) The end game. As its title suggests, this routine finishes the game off by printing the score and prompting the player for another game.

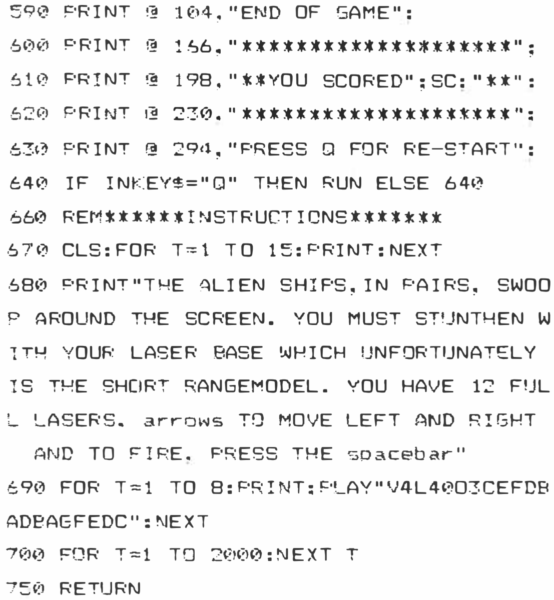
(6) Instructions. This routine provides the uninitiated player with some brief rules, etc, on the game itself.

This modular approach to games writing is a good way of constructing a well-structured, easy-to-understand program. It is not possible in all games to use this approach effectively, but in a game such as the one under discussion you can see that the program is improved as a result. You will also find it easier to modify in this format.

```
10 REM*****ALIEN SWARM*****
20 CLS:PRINT"()()()()()ALIEN SWARM()()()
()()"
25 PRINT:PRINT:PRINT:INPUT"INSTRUCTIONS"
;A$:IF A$<>"N" THEN GOSUB 660
40 REM******INITIALISE******
50 A=RND(180)+25:B=RND(100)+40
60 X=120:Y=170
70 M=1:SC=0
80 DIM B(30,12),L(30,12)
```

```
90 DIM A(35,20),K(35,20)
100 PMODE 2,1:PCLS:SCREEN 1,1
130 LINE(120,180)-(148,180),PSET
140 CIRCLE(50,50),15,1,0.3
150 CIRCLE(50,50),7
160 PAINT(43,50),1,1:PAINT(59,50),1,1
170 GET(33,40)-(67,60),A,G
190 DRAW"BM120,180;U2R4U1R1U1R1U1R8U4R1D
4R8D1R1D1R1D1R4D2"
190 PAINT(127,177),1,1
200 GET(120,170)-(150,181),B,G
205 LINE(0,185)-(255,185),PSET
210 REM******ACTION******
220 PUT(33,40)-(67,60),K,PSET
230 PUT(X,Y)-(X+30,Y+11),B,PSET
240 PUT(A,B)-(A+34,B+20),A,PSET
250 PUT(E,D)-(E+34,D+20),A,PSET
260 PLAY"L100O1DBCG"
270 PUT(X,Y)-(X+30,Y+11),L,PSET
280 PUT(A,B)-(A+34,B+20),K,PSET
290 PUT(E,D)-(E+34,D+20),K,PSET
300 IF PEEK(343)=223 AND X>30 THEN X=X-3
0
310 IF PEEK(344)=223 AND X<225 THEN X=X+
30
320 A=A+RND(50)-RND(50):IF A>230 OR A<25
 THEN A=120
322 B=B+RND(40)-RND(30):IF B>150 OR B<40
```

```
 THEN B=70
330 E=RND(180)+25:D=RND(100)+40
340 IF PEEK(345)=223 THEN GOSUB 360
350 GOTO 210
360 REM*****LASER FIRE*****
370 PUT(X,Y)-(X+30,Y+11),B,PSET
380 PUT(A,B)-(A+34,B+20),A,PSET
390 PUT(E,D)-(E+34,D+20),A,PSET
400 FOR C=170 TO 110 STEP-2
410 IF C>B AND C<B+20 AND X+15>A AND X+1
5<A+34 THEN GOSUB 520
420 GOTO 430
430 LINE(X+15,Y)-(X+15,Y-60),PSET
440 NEXT C
450 LINE(X+15,Y)-(X+15,Y-60),PRESET
460 PUT(X,Y)-(X+30,Y+11),B,PSET
470 PUT(A,B)-(A+34,B+20),K,PSET
490 PUT(E,D)-(E+34,D+20),K,PSET
490 PLAY"L255O2CBCBCBCBCB"
500 M=M+1:IF M=12 THEN 580
510 RETURN
520 PUT(E,D)-(E+33,D+20),A,PSET
530 PUT(A,B)-(A+33,B+20),A,PSET
540 PLAY"L255O1GFEDCC"
550 SC=SC+100
560 RETURN
570 REM******END OF GAME*******
580 CLS 0
```



```
590 PRINT @ 104,"END OF GAME":
600 PRINT @ 166,"********************";
610 PRINT @ 198,"**YOU SCORED";SC:"**":
620 PRINT @ 230,"********************";
630 PRINT @ 294,"PRESS Q FOR RE-START":
640 IF INKEY$="Q" THEN RUN ELSE 640
660 REM******INSTRUCTIONS*******
670 CLS:FOR T=1 TO 15:PRINT:NEXT
680 PRINT"THE ALIEN SHIPS,IN PAIRS, SWOO
P AROUND THE SCREEN. YOU MUST STUNTHEN W
ITH YOUR LASER BASE WHICH UNFORTUNATELY
IS THE SHORT RANGEMODEL. YOU HAVE 12 FUL
L LASERS. arrows TO MOVE LEFT AND RIGHT
  AND TO FIRE. PRESS THE spacebar"
690 FOR T=1 TO 8:PRINT:PLAY"V4L40O3CEFDB
ADBAGFEDC":NEXT
700 FOR T=1 TO 2000:NEXT T
750 RETURN
```

SOLITAIRE

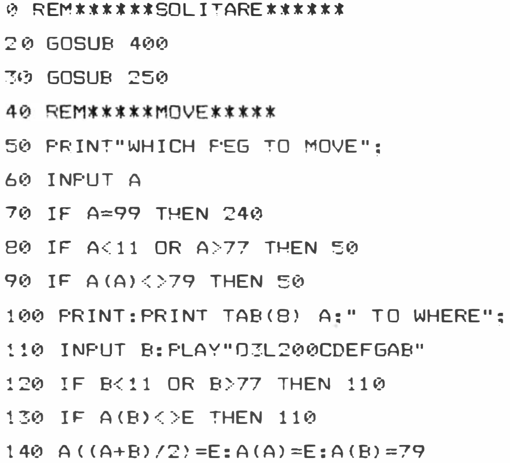
This classic game, at one time played with small pebbles in ancient Greece, is given an up-to-date touch in this computerised version. You must jump over one peg with another, either horizontally or vertically, and the peg that has been jumped over is removed from the board. The object of the game using this movement is to clear the board of pegs except for one left in the middle. In this game you move by entering the co-ordinates of the peg you wish to move followed by the co-ordinates of the position you wish to move the peg to. The computer will reject any illegal moves. You enter the side number and then the top number, eg. if you wish to move peg '6' (side co-ordinate), '4' (top co-ordinate), to position '4' (side co-ordinate), '4' (top co-ordinate), you would first enter '64' followed by '44'.

If you succeed in completing the puzzle, the computer will tell you how many moves you took. If you are not so fortunate, the computer will display the number of pegs still left.

If, after a number of attempts you still cannot succeed, then try the solution below. Many thanks to George Furlonger of Fareham for his solution.

Solution:

46 - 44, 65 - 45, 57 - 55, 37 - 57 45 - 65,
75 - 55, 73 - 75, 63 - 65, 54 - 56 57 - 55,
25 - 45, 55 - 35, 75 - 55, 34 - 54 36 - 34,
43 - 63, 51 - 53, 31 - 51, 32 - 52 63 - 43,
51 - 53, 43 - 63, 55 - 53, 63 - 43 34 - 32,
13 - 33, 15 - 13, 43 - 23, 13 - 33 32 - 34,
24 - 44.



```
0 REM******SOLITARE******
20 GOSUB 400
30 GOSUB 250
40 REM*****MOVE*****
50 PRINT"WHICH PEG TO MOVE";
60 INPUT A
70 IF A=99 THEN 240
80 IF A<11 OR A>77 THEN 50
90 IF A(A)<>79 THEN 50
100 PRINT:PRINT TAB(8) A;" TO WHERE";
110 INPUT B:PLAY"O3L200CDEFGAB"
120 IF B<11 OR B>77 THEN 110
130 IF A(B)<>E THEN 110
140 A((A+B)/2)=E:A(A)=E:A(B)=79
```

```
150 MV=MV+1:CO=0
170 FOR F=11 TO 75
180 IF A(F)=79 THEN CO=CO+1
190 NEXT F
200 GOSUB 250
205 PRINT:PRINT"THERE ARE ";CO;" PEGS LE
FT"
220 IF CO<>1 THEN 40
230 IF A(44)=79 THEN PRINT:PRINT"YOU COM
PLETED THE PUZZLE IN ":PRINT"JUST ";MV;"
 MOVES!":PLAY"L8O4CDEFGABO5CC":END
240 PRINT:PRINT"THE GAME IS OVER, AND YO
U FAILED":PLAY"L4O1EDCC":END
250 REM******DISPLAY*******
260 CLS
270 PRINT"ENTER SIDE CO-ORDINATE FIRST."
280 PRINT TAB(8)"ENTER 99 TO CONCEDE"
290 PRINT:PRINT"  1 2 3 4 5 6 7"
300 PRINT TAB(5);
310 FOR D=11 TO 75
320 T=10*(INT(D/10))
330 IF D-T=8 THEN D=D+2:PRINT T/10:PRINT
 TAB(5):;GOTO 350
340 PRINT CHR$(A(D));" ";
350 NEXT D:PRINT"  7"
360 PRINT:PRINT"MOVES SO FAR: ";MV
370 RETURN
400 REM******INITIALISE*******
```

```
410 CLS:DIM A(97):E=42
420 FOR D=11 TO 75
430 T=10*(INT(D/10))
440 IF D-T=8 THEN D=D+3
470 READ A(D)
480 NEXT D
490 MV=0
500 RETURN
510 REM*******DATA BLOCK********
520 DATA 32,32,79,79,79,32,32
530 DATA 32,32,79,79,79,32,32
540 DATA 79,79,79,79,79,79,79
550 DATA 79,79,79,42,79,79,79
560 DATA 79,79,79,79,79,79,79
570 DATA 32,32,79,79,79,32,32
580 DATA 32,32,79,79,79
```

FOUR BY FOUR

A family favourite this, where you must attempt to get four of your counters in a row either horizontally, diagonally or vertically. Your pieces are represented by the 'H' characters and the computer's pieces by the 'C' characters. The computer always allows you to go first and you move simply by entering the column you wish to slide a counter down.

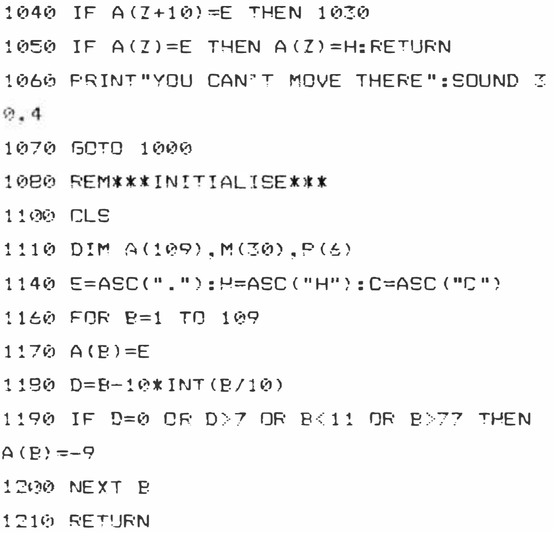
The computer plays very fast (replies are almost instant) and surprisingly well! The computer is weaker near the beginning of the game but the longer the game progresses, the stronger an opponent it gets.

```
10 REM******FOUR BY FOUR*******
20 GOSUB 1080
```

```
25 GOSUB 860:GOSUB 680
30 GOSUB 970:GOSUB 860:GOSUB 680
40 GOSUB 110:GOTO 25
110 REM***COMPUTER MOVE***
120 PRINT:PRINT"STAND BY FOR MY MOVE..."
130 SOUND 180,1:B=10
140 B=B+1
150 IF A(B)=-9 THEN 180
160 IF A(B)=C THEN X=C:GOTO 210
170 IF A(B)=H THEN X=H:GOTO 210
180 IF B<77 THEN 140
190 GOTO 490
210 REM***CHANCE OF 4 IN A ROW***
230 IF A(B+1)=X AND A(B+2)=X AND A(B+3)=
E AND A(B+13)<>E THEN MOVE=B+3:GOTO 650
240 IF A(B-1)=X AND A(B-2)=X AND A(B-3)=
E AND A(B+7)<>E THEN MOVE=B-3:GOTO 650
250 IF A(B+1)=X AND A(B+2)=X AND A(B-1)=
E AND A(B+9)<>E THEN MOVE=B-1:GOTO 650
260 IF A(B-1)=X AND A(B+2)=X AND A(B+1)=
E AND A(B+11)<>E THEN MOVE=B+1:GOTO 650
270 IF A(B+1)=X AND A(B-1)=X AND A(B+2)=
E AND A(B+12)<>E THEN MOVE=B+2:GOTO 650
280 IF A(B+1)=X AND A(B-1)=X AND A(B-2)=
E AND A(B+8)<>E THEN MOVE=B-2:GOTO 650
290 IF A(B-1)=X AND A(B-2)=X AND A(B+1)=
E AND A(B+11)<>E THEN MOVE=B+1:GOTO 650
310 IF B>20 THEN IF A(B-10)=X AND A(B-20
)=X AND A(B+10)=E AND A(B+20)<>E THEN MO
VE=B+10:GOTO 650
330 IF A(B+11)=X AND A(B+22)=X AND A(B-1
1)=E AND A(B-1)<>E THEN MOVE=B-11:GOTO 6
50
340 IF A(B+9)=X AND A(B+18)=X AND A(B-9)
=E AND A(B+1)<>E THEN MOVE=B-9:GOTO 650
360 REM***CHANCE OF 3 IN A ROW***
380 IF A(B+1)=X AND A(B+2)=E AND A(B+12)
<>E THEN MOVE=B+2:GOTO 650
390 IF A(B+1)=X AND A(B-1)=E AND A(B+9)<
>E THEN MOVE=B-1:GOTO 650
400 IF A(B-1)=X AND A(B-2)=E AND A(B+8)<
>E THEN MOVE=B-2:GOTO 650
420 IF A(B+10)=X AND A(B-10)=E AND A(B)<
>E THEN MOVE=B-10:GOTO 650
440 IF A(B+9)=X AND A(B-9)=E AND A(B+1)<
>E THEN MOVE=B-9:GOTO 650
450 IF B>11 THEN IF A(B+11)=X AND A(B-11
)=E AND A(B-1)<>E THEN MOVE=B-11:GOTO 65
0
460 GOTO 180
490 FOR N=1 TO 3
500 M(N)=0
510 NEXT N
520 COUNT=0
530 FOR B=11 TO 77
540 IF A(B)<>C AND A(B)<>H THEN 600
```

```
550 IF A(B+1)=E AND A(B+11)<>E THEN COUN
T=COUNT+1:M(COUNT)=B+1
560 IF A(B-1)=E AND A(B+9)<>E THEN COUNT
=COUNT+1:M(COUNT)=B-1
570 IF A(B-10)=E AND A(B)<>E THEN COUNT=
COUNT+1:M(COUNT)=B-10
580 IF A(B-11)=E AND A(B-1)<>E THEN COUN
T=COUNT+1:M(COUNT)=B-11
590 IF A(B-9)=E AND A(B+1)<>E THEN COUNT
=COUNT+1:M(COUNT)=B-9
600 NEXT B
610 IF COUNT<>0 THEN 640
620 PRINT:PRINT"I THINK IT'S A DRAW.":SO
UND 120,6:END
640 MOVE=M(RND(COUNT))
650 A(MOVE)=C
660 RETURN
670 REM****WIN CHECK****
680 X=H
700 B=10
710 B=B+1:IF A(B)<>X THEN 770
730 IF A(B+1)=X AND A(B+2)=X AND A(B+3)=
X THEN 800
740 IF B>30 THEN IF A(B-10)=X AND A(B-20
)=X AND A(B-30)=X THEN 800
750 IF B>33 THEN IF A(B-11)=X AND A(B-22
)=X AND A(B-33)=X THEN 800
760 IF B>27 THEN IF A(B-9)=X AND A(B-18)
```

```
=X AND A(B-27)=X THEN 800
770 IF B<77 THEN 710
780 IF X=H THEN X=C:GOTO 700
790 RETURN
800 REM***WIN FOUND***
810 PRINT:PRINT
820 IF X=H THEN PRINT"YOU'VE BEATEN ME,
HUMAN!"
830 IF X=C THEN PRINT"I'VE DEFEATED YOU
HUMAN!"
835 SOUND 200,6:END
850 REM***PRINT BOARD***
860 CLS:PRINT"*********FOUR BY FOUR*****
*****":PRINT
870 FOR K=10 TO 70 STEP 10
890 PRINT:PRINT TAB(5);
900 FOR J=1 TO 7
910 PRINT CHR$(A(K+J));" ";
920 NEXT J:NEXT K
935 PRINT:PRINT TAB(5)"1 2 3 4 5 6 7":PR
INT
960 RETURN
970 REM***HUMAN MOVE***
990 PRINT"  YOUR MOVE..."
1000 PRINT"WHICH COLUMN DO YOU WISH TO"
1010 INPUT"MOVE INTO ";J
1020 Z=J
1030 Z=Z+10
```



```
1040 IF A(Z+10)=E THEN 1030
1050 IF A(Z)=E THEN A(Z)=H:RETURN
1060 PRINT"YOU CAN'T MOVE THERE":SOUND 3
0,4
1070 GOTO 1000
1080 REM***INITIALISE***
1100 CLS
1110 DIM A(109),M(30),P(6)
1140 E=ASC("."):H=ASC("H"):C=ASC("C")
1160 FOR B=1 TO 109
1170 A(B)=E
1180 D=B-10*INT(B/10)
1190 IF D=0 OR D>7 OR B<11 OR B>77 THEN
A(B)=-9
1200 NEXT B
1210 RETURN
```

MANAGING DIRECTOR

In this game, you are (surprise, surprise) the Managing Director of a large company. You must attempt to keep the factory running until you manage to make 10000 dollars (this counts capital and stocks together). You have a lot of problems though. First, you have to deal with the far from friendly unions who won't always let you get rid of the people you want to and insist on pay rises which you have to follow. On top of that, there are workers who hardly ever meet the production targets you set, raw material suppliers who enjoy putting their prices up and the consumer market who resist price rises quite strongly.

After all that, you may feel that the whole world is against you. Do not despair, a few attempts of this simulation and you will get better and better at looking after your factory. While this can hardly be called a serious simulation, it does show simple business ideas in operation. You will learn, after a few frustrating attempts, how to manipulate the various resources available to you.

The program is quite a long one but you can see from playing it a couple of times, how much work it has to perform. The random number generator is used frequently to give more variation to the game; however, where necessary, I have 'weighted' the outcome to a particular problem to make the program more enjoyable.

```
10 REM*****MANAGING DIRECTOR******
20 GOSUB 1670
30 WE=WE+1
40 GOSUB 930
50 GOSUB 1300
60 GOSUB 930
70 GOSUB 1130
80 GOSUB 930
90 GOSUB 730
100 GOSUB 140
110 CA=CA-WA*WO-RC
120 GOTO 30
140 REM*****PROBLEMS*****
150 CLS:IF RND(0)<.45 THEN 260
170 A=RND(7):PRINT:PRINT
190 PRINT"THE UNIONS ARE DEMANDING A"
200 PRINT"PAY RISE OF";A;"%"
210 WA=INT(100*(WA+WA*A/100))/100
220 GOSUB 1840
```

```
230 PRINT:PRINT"PAY PER EMPLOYEE IS NOW
$";WA
240 GOSUB 1840:CLS
260 IF RND(0)<.81 THEN 410
270 PRINT:PRINT:PRINT"A FIRE IN YOUR WAR
EHOUSE HAS"
280 PRINT"DESTROYED SOME STOCK. PLEASE"
290 PRINT"STAND BY FOR A REPORT ON"
300 PRINT"THE DAMAGE CAUSED....":GOSUB 1
840
330 A=INT(RND(ST/2):ST=ST-A
350 PRINT:PRINT"THERE WERE";A;A$
360 PRINT"DESTROYED. THEY WERE"
370 PRINT"WORTH $";A*SP;"RETAIL"
380 GOSUB 1840:PRINT"STOCK IN HAND IS NO
W";ST;A$
410 IF RND(0)>.3 THEN 560
420 CLS:PRINT;PRINT
440 PRINT"YOUR MAIN SUPPLIER HAS"
450 PRINT"ANNOUNCED A PRICE RISE...":GOS
UB 1940
470 A=INT(RND(100*CO/7))/100
480 IF A<.01 THEN 470
490 PRINT:PRINT"THE COST OF MAKING ";A$
500 PRINT"HAS RISEN BY $";A;"EACH"
510 GOSUB 1840:CO=CO+A
530 PRINT:PRINT"IT NOW COSTS $";CO
540 PRINT"TO MAKE EACH ONE...":GOSUB 184
```

```
0
560 IF RND(0)<.65 AND MA<SF THEN RETURN
570 CLS:PRINT:PRINT
590 PRINT"YOU HAVE A CHANCE TO RAISE"
600 PRINT"YOUR PRICE. YOUR ";A$
610 PRINT"NOW SELL FOR $";SF:GOSUB 1840
630 PRINT:PRINT"WHAT PERCENTAGE INCREASE
 WOULD"
640 INPUT"LIKE TO IMPOSE";A
650 RE=RE+A:SF=INT(100*(SF+A*SF/100))/10
0
680 GOSUB 1840
690 PRINT:PRINT"THE ";A$;" NOW SELL":PRI
NT"FOR $";SF:GOSUB 1840
700 RETURN
730 REM*********SALES**********
740 PRINT:PRINT"YOUR TOTAL STOCK OF"
750 PRINT A$;" IS";ST:GOSUB 1840
770 PRINT:PRINT"PLEASE STAND BY FOR A"
780 PRINT"SALES REPORT..."
790 R=INT(ST/(RE/100)):A=RND(R)
900 IF A>ST THEN 790
930 PRINT:PRINT"THE TOTAL NUMBER OF ";A$
940 PRINT"SOLD WAS";A
950 ST=ST-A:ZA=A*SF
970 PRINT:PRINT"THE INCOME FROM THAT"
980 PRINT"SALE WAS $";ZA
990 CA=INT(A*SF*100)/100+CA
```

```
900 GOSUB 1840:GOSUB 1940:RETURN
930 REM******CHAIRMAN'S REPORT*******
940 CLS:IF CA+ST<1 THEN 1510
960 IF CA+ST>9999 THEN PRINT"YOU'VE MADE
 $10,000 AND CAN NOW RETIRE...":GOTO 159
0
970 PRINT"SHOP FLOOR REPORT, SIR."
980 PRINT"      FOR WEEK":WE
990 PRINT:PRINT"CAPITAL IN HAND IS $";IN
T(CA*100)/100
1000 PRINT "RUNNING COSTS ARE $";RC:"A W
EEK"
1010 PRINT"YOUR STORES HOLD";ST:A$
1020 PRINT"WORTH $";INT(ST*SP*100)/100
1030 PRINT"THEY SELL FOR $";SP;"EACH"
1040 PRINT"AND COST $";CO:"EACH TO MAKE"
1050 PRINT"YOUR WORKFORCE IS NOW"
1060 PRINT WO:"STRONG, AND YOU ARE"
1070 PRINT"PAYING THEM $":WA:"EACH, SO T
HE"
1080 PRINT"WAGES BILL THIS WEEK IS $";WA
*WO
1085 PRINT:INPUT"PRESS ENTER TO CONTINUE
";D$:CLS
1090 PRINT"EACH PERSON CAN MAKE";PR
1110 PRINT A$;" A WEEK, A TOTAL"
1120 PRINT"OUTPUT OF";PR*WO:PRINT:SOUND
150,1:RETURN
```

```
1130 PRINT:INPUT"HOW MANY TO PRODUCE";MA
1140 IF MA<1 THEN RETURN
1160 IF MA*CO>CA THEN PRINT"YOU DO NOT H
AVE ENOUGH MONEY":GOTO 1130
1170 IF MA>PR*WO THEN PRINT"YOU DO NOT H
AVE ENOUGH WORKERS!":GOTO 1130
1180 PRINT"YES SIR, THE TARGET FOR WEEK"
;WE
1190 PRINT"IS";MA;A$:MA=INT(MA-RND(MA/5)
)
1200 GOSUB 1840:PRINT:PRINT"THE NUMBER O
F ";A$
1230 PRINT"ACTUALLY PRODUCED IN WEEK";WE
1240 PRINT"WAS";MA;"..."
1250 ST=ST+MA:CA=CA-CO*MA
1270 GOSUB 1840:RETURN
1300 REM******STAFF******
1310 PRINT"HOW MANY PEOPLE DO YOU"
1320 INPUT"WANT TO HIRE";A:WO=WO+A
1340 PRINT:PRINT"THE TOTAL WORKFORCE"
1350 PRINT"IS NOW";WO;"STRONG":GOSUB 184
0
1370 IF A>0 THEN RETURN
1380 GOSUB 930
1390 PRINT"HOW MANY PEOPLE DO YOU"
1400 INPUT"WISH TO FIRE";A
1410 IF A=0 THEN GOSUB 1840:RETURN
1420 IF A>WO THEN 1390
```

```
1430 A=RND(A):GOSUB 1840
1450 PRINT:PRINT"THE UNIONS WILL ALLOW"
1460 PRINT"YOU TO GET RID OF":A
1470 WO=WO-A:GOSUB 1840:RETURN
1510 REM******BANKRUPTCY******
1520 CLS:PRINT TAB(14)"YOUR":PRINT
1530 PRINT"  OOOO  OOOO  OOOO  O  O  OOO
O"
1540 PRINT"  O  O  O  O  O  O  O O   O"
1550 PRINT"  OOOO  OOOO  O  O  OO    OOO
O"
1560 PRINT"  O  O  O O   O  O  O O   O"
1570 PRINT"  OOOO  O  O  OOOO  O  O  OOO
O"
1590 PRINT:PRINT"ENTER 'Y' FOR ANOTHER S
TINT"
1600 PRINT"AS MANAGING DIRECTOR."
1640 IF INKEY$="Y" THEN RUN ELSE 1640
1670 REM******INITIALISE******
1690 FOR Z=1 TO RND(6):READ A$:NEXT Z
1720 CA=500+RND(500):ST=100+RND(500)
1740 SP=10+RND(5)
1750 CO=7+RND(5)
1760 IF CO>SP THEN 1750
1770 WO=7+RND(10):WA=12+RND(SP)
1790 PR=5+RND(6):RC=100+RND(20)
1800 WE=0:RE=1:RETURN
1840 FOR Z=1 TO 3000:NEXT Z:RETURN
```

```
1850 DATA KEYBOARDS,WIDGETS,RADIOS,DISC
DRIVES, BEER MATS, ENGINES, UMBRELLAS
```

CHAPTER 7

SOUND AND GRAPHICS FOR YOUR OWN ADVENTURES

As the title suggests, this chapter includes some ideas that you can use when writing your own adventures. These graphic and sound routines can liven up a drab adventure and give you ideas to create others.

While many purists believe that an adventure should either be purely text or purely graphic, there are many people (I amongst them) who believe that an occasional graphic display can only heighten the enjoyment that a user receives from playing the game, providing two factors are noted. Firstly, the graphic displays should be used in locations that are not often visited - a player does not want to repeatedly come across the same display too often - and also there should not be too many of them about if the adventure is to be a primarily text one. Secondly, they should not take too long to generate; a high resolution picture of great complexity may look wonderful, but if it takes six or seven minutes to be drawn then the magic is lost.

We will look at some graphic displays first. By simply using pattern displays occasionally, the game can be enlivened. The Dragon 32, though limited in some graphic aspects, does have a good range of easy-to-use graphic commands. With the use of simple routines, effective patterns can be generated such as the one on the next page.

```
1Ø REM*****TIME WARP*****
2Ø PMODE 3,1:PCLS:SCREEN 1,Ø
3Ø C=RND(4):X=RND(254):Y=RND(192)
35 COLOR C,C
4Ø LINE(128,96)-(X,Y),PSET
5Ø GOTO 3Ø
```

Line 2Ø sets up one of the high resolution screens; in fact, the four colour (red, green, blue and yellow), 128 by 192 screen. Line 3Ø chooses three random numbers; the first, 'C', acts as the colour of the line; the second is the horizontal position of the line; and the third is the vertical position of the line. Line 35 changes the colour of the next line to be drawn. Line 4Ø draws the line from the centre of the screen to the co-ordinates generated in line 3Ø. This line creates the effect with all the lines starting from the centre and bursting out all over the screen. Line 5Ø takes the program back to line 3Ø to draw another line.

Your computer also has a powerful CIRCLE command which is the basis of this next simple pattern. Why not try and adjust the parameters in this program and see if you can achieve some other surprising effects.

```
1Ø REM******CIRCULAR MOTIONS*******
2Ø N=1:PMODE 4,1:PCLS:SCREEN 1,1
3Ø Y=1Ø-(1Ø*N)
4Ø FOR T=1 TO 25Ø STEP 11
5Ø CIRCLE(T,Y),2Ø,N
6Ø NEXT T
7Ø Y=Y+2Ø:IF Y>18Ø THEN 8Ø ELSE GOTO 4Ø
8Ø N=N+1:IF N>1 THEN N=Ø
9Ø GOTO 3Ø
```

This next pattern is an ever-changing one that can hold one's attention for some considerable time. The pattern constantly evolves and it is fair enough to say that after a couple of minutes, the display becomes quite spectacular. When you consider the length of the program compared to the results you get on the screen, you begin to realise what an impressive little program this is. It is ideal for an adventure as it uses little memory and could be situated at the end of the game where, say, the adventurer is trapped forever. Anyway try it out.

```
 10 REM****ETERNAL TRANSFORMATION****
 20 PMODE 3,1:PCLS:SCREEN 1,0
 30 FOR A=3 TO 7:FOR B=2 TO 6
 40 FOR C=0 TO 3:FOR D=0 TO 3
 50 COLOR D,C
 60 W=0:X=255:Y=0:Z=191
 70 LINE(W,Y)-(X,Z),PSET,B
 80 W=W+B:X=X-B:Y=Y+A:Z=Z-A
 90 IF W<255 AND Y<191 THEN 70
 100 NEXT D:NEXT C:NEXT B:NEXT A
 110 GOTO 40
```

This next program generates my favourite pattern of the lot, and serves as a good illustration of the computer's LINE command. Watch the triangles as they go bouncing around the screen. Why not add several lines to erase the previous triangle before it draws the next, to give a more animated display.

```
 10 REM*****TRAVELLING TRIANGLES*****
 20 PMODE 3,1:PCLS:SCREEN 1,0
 30 C=RND(4):X=RND(254):Y=RND(192)
 40 A=RND(254):B=RND(192)
```

```
50 COLOR C,C
60 LINE(128,96)-(X,Y),PSET
70 LINE(X,Y)-(A,B),PSET
80 LINE(A,B)-(128,96),PSET
90 FOR T=1 TO 50:NEXT T
100 GOTO 30
```

Let's get onto something a little less abstract now and start to design some graphic shapes on-screen which represent something a little more recognisable than our patterns have. In 'The Dark Forest' there are a number of locations depicted in high resolution. I have indicated where these occur in the program in chapter two, so have a look at them to supplement the ones given below.

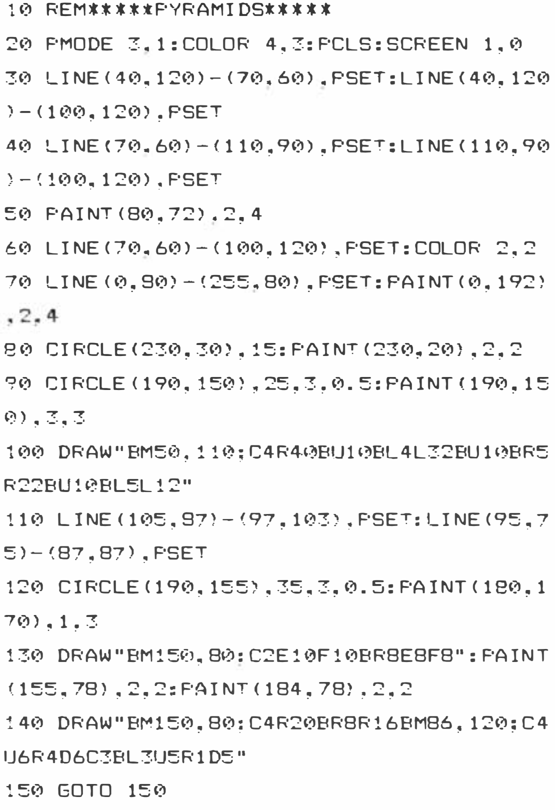
With the computer's highest resolution being in either black/white or black/green, the capability for creating good space scenes is excellent. Below are two such scenes using DRAW, PAINT, LINE, CIRCLE and PSET to complete the scene. Note how in both programs, PSET is used to randomly position 'stars' on the background.

```
10 REM******LUNAR SCENE******
20 PMODE 4,1:PCLS:SCREEN 1,1
30 DRAW"BM0,60;R6D4R3D7R3D2R4D3R3D6R2D1R
4D7R6D3R4D2R3D1R4D3R5D2R4"
40 DRAW"BM213,137;U6R2U6R4U3R5U3R6U11R5U
4R3U12R3U4R1U6R3U5U3R3U12R3U10R2U1R1U2R3
"
50 CIRCLE(150,100),100,1,.5,0.15,0.5
60 DRAW"BM145,114:R4F8D12G4L12H4U12E8BD2
4BR2F6L12E6"
70 DRAW"BM158,134;F4D10L1R2BL32R2L1U10E4
"
```

```
80 LINE(140,124)-(155,130),PSET,BF
90 FOR T=1 TO 150:PSET(RND(255),RND(100)
,1):NEXT T
100 GOTO 100
```

```
0 REM******CRATERED PLANET******
20 PMODE 4,1:PCLS:SCREEN 1,1
30 CIRCLE(128,190),190,1,0.5
40 CIRCLE(158,140),50,1,0.3
50 CIRCLE(158,150),30,1,0.3,0.5,0
60 CIRCLE(30,170),50,1,0.3
70 CIRCLE(30,180),20,1,0.3
80 CIRCLE(40,40),10:CIRCLE(40,40),15,1,0
.5
90 PAINT(125,145),1,1:PAINT(1,175),1,1
100 FOR S=1 TO 50:A=RND(250):B=RND(100)
110 PSET(A,B,1):PAINT(40,40),1,1
120 NEXT
130 GOTO 130
```

Let's get more down to earth now and try our hand at recapturing the flavour of Egypt on the video display. This program, slightly longer than the rest, produces a believable picture of a typical desert scene together with pyramids. If you don't like my choice of colours or shape of pyramid, then go right ahead and change them as much as you like. The whole idea of these graphics is to give you some starting points to create your own displays (though you can use my own, by no means, definitive versions as theyarein your own adventures).



```
10 REM*****PYRAMIDS*****
20 PMODE 3,1:COLOR 4,3:PCLS:SCREEN 1,0
30 LINE(40,120)-(70,60),PSET:LINE(40,120
)-(100,120),PSET
40 LINE(70,60)-(110,90),PSET:LINE(110,90
)-(100,120),PSET
50 PAINT(80,72),2,4
60 LINE(70,60)-(100,120),PSET:COLOR 2,2
70 LINE(0,90)-(255,80),PSET:PAINT(0,192)
,2,4
80 CIRCLE(230,30),15:PAINT(230,20),2,2
90 CIRCLE(190,150),25,3,0.5:PAINT(190,15
0),3,3
100 DRAW"BM50,110;C4R40BU10BL4L32BU10BR5
R22BU10BL5L12"
110 LINE(105,97)-(97,103),PSET:LINE(95,7
5)-(87,87),PSET
120 CIRCLE(190,155),35,3,0.5:PAINT(180,1
70),1,3
130 DRAW"BM150,80:C2E10F10BR8E8F8":PAINT
(155,78),2,2:PAINT(184,78),2,2
140 DRAW"BM150,80;C4R20BR8R16BM86,120;C4
U6R4D6C3BL3U5R1D5"
150 GOTO 150
```

You should now be able to isolate which lines draw which object on the screen and modify the program accordingly.

Objects and characters can be as easily displayed on-screen. Just to show you what can be done, there follows two programs: one drawing a gilded knife or dagger and the other depicting a cheeky little Gremlin.

```
1Ø REM*****A KNIFE OR DAGGER*****
2Ø PMODE 4,1:PCLS:SCREEN 1,1
3Ø LINE(3Ø,9Ø)-(17Ø,7Ø),PSET
4Ø LINE(3Ø,9Ø)-(17Ø,11Ø),PSET
5Ø LINE(3Ø,9Ø)-(17Ø,9Ø),PSET
6Ø DRAW"BM17Ø,7Ø;U2ØR8D25R45F5D2ØG5L45D2
5L8U6Ø"
7Ø PAINT(18Ø,9Ø),1,1:DRAW"BM185,7Ø;CØD4Ø
BR1ØU4ØBR1ØD4ØBR1ØU4ØBR1ØD4Ø"
8Ø GOTO 8Ø
```

```
1Ø REM******A LAUGHING GREMLIN*******
2Ø PMODE 4,1:PCLS:SCREEN 1,1
3Ø X=5Ø:FOR T=5 TO 3Ø STEP 2
4Ø CIRCLE(X,4Ø),T,1,1.2:X=X+2
5Ø NEXT T:X=12Ø
6Ø FOR T=5 TO 3Ø STEP 2
7Ø CIRCLE(X,4Ø),T,1,1.2:X=X+2:NEXT T
8Ø FOR T=1 TO 18
9Ø CIRCLE(1Ø5,1ØØ),T,1,1.8:NEXT T
1ØØ CIRCLE(1Ø5,1ØØ),6Ø,1,Ø.7,Ø.1,Ø.4
11Ø CIRCLE(1Ø5,9Ø),6Ø,1,1.4,Ø.1,Ø.41
12Ø GOTO 12Ø
```

Note how the circles drawn (lines 3Ø and 7Ø) are offset by two each time to give the depth of the Gremlin's eyes.

Finally in the graphics part of this chapter, here is another little pattern for your amusement and possible inclusion in your very own adventure game. Little has to be said about the program as the display produced is explained by the title, 'Lace Webb'.

```
10 REM*****LACE WEB*****
20 PMODE4,1:PCLS0:SCREEN1,1
30 FOR Z=4 TO 20 STEP 4
40 IF Z=12 OR Z=20 THEN 60 ELSE 50
50 FOR X=0 TO 255 STEP Z:GOTO70
60 FOR X=255 TO 0 STEP -Z
70 Y=INT(X*191/255)
80 LINE(0,Y)-(X,191),PSET
90 LINE(X,191)-(255,191-Y),PSET
100 LINE(255,191-Y)-(255-X,0),PSET
110 LINE(255-X,0)-(0,Y),PSET
120 NEXT X:C=1:M=85
130 FOR T=1 TO M
140 CIRCLE(128,96),T,C,0.6
150 NEXT T
160 C=C+1:IF C>1 THEN C=0
170 M=M-9:IF M<9 THEN 170
190 GOTO 130
```

If you are considering using quite a few graphic displays in your games, you may come across a large problem and that is what to do about on-screen writing. Even if your picture is very good, it may require a few words at the bottom of the screen to give the player the hint that he or she needs. As this computer cannot mix text and graphics on the same screen, you are left with two choices: either display the picture for a short time and then revert back to the text screen with the message printed on it; or define graphically the letters you need to make up your message and draw them at the required place on the graphic screen.

Both techniques have good and bad points. It is much simpler and less memory-consuming to revert back to the text screen, but this method is far less presentable than the graphics screen message. Let's consider the graphics screen option.

The letters themselves can be created on the graphics screen in one of several ways. The LINE command could be used to create each letter or the letter could be defined by the DRAW command; the latter option is really the most efficient way. If you intend to use quite a few words, then the best thing to do would be to define a whole character set using DRAW and have the data required to do this positioned as a subroutine in the program. You could then use an ON . . . GOSUB to take you to the particular line of data to draw each character. The letters could be numbered '1' for the A character, '2' for the B character, and so on. To write a word, a DATA statement could hold the numbers needed in the ON . . . GOSUB statement, ie. to write the word 'HELLO', the data would thus be 8,5,12,12,15.

In my 'Graphic Alphabet' program, I have defined an alphabet including numbers and certain punctuation marks from line 9ØØØ onwards. Line 13Ø goes to the subroutine that defines the specified character. Line 2Ø is also important in that it sets up the graphic screen. The rest of the program provides the demonstration aspect of the program by converting your computer into a graphic typewriter. Line 3Ø provides your starting position and initialises the variables required. Line 4Ø prints a tiny cursor and 'strobes' the keyboard waiting for a key to be pressed. Line 5Ø converts the key pressed into a numeric value for X, 'X' being the variable in the ON . . . GOSUB statement in line 13Ø.

Lines 6Ø to 12Ø convert any values of V$ that line 5Ø was unable to do. As with line 5Ø, the figure is stored in X. Finally, line 14Ø checks to see if you have reached the bottom of the screen. Line 15Ø then draws the character and adds one to the variable, L (the 'characters across' value). If L reaches 25 then a new line is started.

By typing the whole program in, you will see how the graphics are displayed. You can delete the unnecessary lines and keep the 'characters' routine together with an ON... GOSUB and a DRAW command somewhere. You would then have your text as part of the graphics which can be now accessed by a DATA statement. If you would prefer to have the characters smaller or larger, or perhaps in a different mode, then a little work has to be done. The scale parameter on your computer's DRAW command would need to be accessed by adding an extra string containing 'Sn', where n is the scale of the lettering. The character is drawn to the scale of n/4 so that if n=2 then the character would be half the previous size, and if n=8 then the character would be double the previous size. If the normal size is required, then there is no need for the extra string. That is fairly straightforward - what will take longer to do is to change some of the characters defined. Some of my characters use odd numbers which, if halved in size, will be rounded and may cause some DRAW directions to be one pixel out either way. The best way to combat this is to print every character on the screen and change the 'odd' ones accordingly.

```
10 REM*****GRAPHIC ALPHABET*****

20 PMODE 4,1:PCLS:SCREEN 1,0

30 DRAW"BM10,17":L=1:M=17

40 DRAW"C1L2":V$=INKEY$:DRAW"C0R2":IF V$
=""  THEN 40

50 X=ASC(V$)-61:DRAW"C1":IF X<1 THEN X=X
+45

60 IF V$=" " THEN X=1

70 IF V$="." THEN X=2

80 IF V$="," THEN X=3

90 IF V$="/" THEN X=30

100 IF V$=CHR$(9) AND L>1 THEN X=31
```

```
110 IF V$=CHR$(13) THEN X=42
120 IF L<2 AND V$=CHR$(8) THEN PLAY"O1L1
00DE":GOTO 40
130 ON X GOSUB 9010,9020,9030,9040,9050,
9060,9070,9080,9090,9100,9110,9120,9130,
9140,9150,9160,9170,9180,9190,9200,9210,
9220,9230,9240,9250,9260,9270,9280,9290,
9300,9310,9330,9340,9350,9360,9370,9380,
9390,9400,9410,9420,9430
140 IF L>23 AND M>180 THEN FOR T=1 TO 10
00:NEXT T:PLAY"O4L20CED":PCLS:GOTO 30
150 DRAW A$:L=L+1:IF L>24 THEN L=1:M=M+1
2:LINE(10,M)-(10,M),PRESET
160 GOTO 40
9000 REM****DATA FOR DRAW*****
9010 A$="BR10":RETURN
9020 A$="BR4R2U1L2BD1BR6":RETURN
9030 A$="BR5BU1L1D1L1BR7":RETURN
9040 A$="U8R6D4L6BR6D4BR4":RETURN
9050 A$="U8R5F1D3L6BR6D3G1L5BR10":RETURN
9060 A$="U8R6BD8L6BR10":RETURN
9070 A$="R1U8L1R6D8L6BR10":RETURN
9080 A$="U8R6BD4BL1L5BD4R6BR4":RETURN
9090 A$="U8R6BD4BL2L4BR10BD4":RETURN
9100 A$="U8R6BD8L6BR6U3L3BR7BD3":RETURN
9110 A$="U8BR6D8BU4L6BR10BD4":RETURN
9120 A$="BU8R6BL3D8L3R6BR4":RETURN
9130 A$="U3BD3R4U8R2L6BD8BR10":RETURN
```

```
9140 A$="U8D5E5G4F4BR5":RETURN
9150 A$="U8D8R6BR4":RETURN
9160 A$="U8F3E3D8BR4":RETURN
9170 A$="U8D1F6D1U8BD8BR4":RETURN
9180 A$="U8R6D8L6BR10":RETURN
9190 A$="U8R6D4L6BD4BR10":RETURN
9200 A$="BU1U7R6D7L6BR3BU2F3BR4":RETURN
9210 A$="U8R6D4L6R2F4BR4":RETURN
9220 A$="R6U4L6U4R6BD8BR4":RETURN
9230 A$="BR3U8L3R6BR4BD8":RETURN
9240 A$="U8BR6D8L6BR10":RETURN
9250 A$="BU8D5F3E3U5BR4BD8":RETURN
9260 A$="U8D8E3F3U8BR4BD8":RETURN
9270 A$="BU8D1F6D1BL6U1E6U1BD8BR4":RETUR
N
9280 A$="BU8F3E3BG3D5BR7":RETURN
9290 A$="BU8R6D1G6D1R6BR4":RETURN
9300 A$="BU6U2R5F1D1G1L2D2BD2D1BR7":RETU
RN
9310 A$="C0L10U1R10U1L10U1R10U1L10U1R10U
1L10U1R10U1L10BD8"
9320 PLAY"L100O5CED":L=L-2:RETURN
9330 A$="U8R6D8L6U1E6U1BD8BR4":RETURN
9340 A$="BR4U8G3BR9BD5":RETURN
9350 A$="BU8R6D4L6D4R6BR4":RETURN
9360 A$="BU8R6D4L5BR5D4L6BR10":RETURN
9370 A$="BU2U6D6R6L3U2D4BR7":RETURN
9380 A$="R6U4L6U4R6BR4BD8":RETURN
```

```
9390 A$="U8R6BD4L4BR4D4L6BR10":RETURN
9400 A$="BU8R6D2G6BR10":RETURN
9410 A$="U8R6D8L6U4R6BR4BD4":RETURN
9420 A$="BU4U4R6D4L6R6D4BR4":RETURN
9430 A$="":L=25:PLAY"O3L100DEC":RETURN
```

We now turn our attention to sound aspect of programming. Your computer has been criticised by many for only having one sound channel. However, single channel sound is, although less versatile, much easier to use than multi-channel sound. You can still get good simple effects with your single channel as the routines below suggest. The names in the REM statements are generally self-explanatory.

```
0 REM*********POLICE SIREN*********
20 PLAY"L8O3BCBCBCBCBCBCBCBCBC"
```

```
10 REM********WARP ENGINES********
20 PLAY"L255O2CDEFGABO3CDEFGABO4CDEFGABA
GFEDCO3BAGFEDCO2BAGFED"
30 GOTO 20
```

```
0 REM*********LASER CYCLE*********
20 PLAY"L255O1CDCDCDEFEFEFEFGAGAGAGAGABA
BABABABAGFGFGFGFGFEDEDEDEDEDL60C"
```

```
0 REM*********LASER CYCLE 2*********
20 PLAY"L100O2CEGBO3DFAO4CEGBAFDO3BGECO2
AFD"
30 GOTO 20
```

```
0 REM*******'MUNCH'*******
20 PLAY"L255O1CEDP6"
30 GOTO 20
```

```
0 REM*******'MUNCH' 2*******
20 PLAY"L255O1CDO4CO1DCP6"
30 GOTO 20
```

```
10 REM*******LASER CYCLE 3********
20 PLAY"L100O2CEGBO3DFAO4CEGBAFDO3BGECO2
AFD"
30 GOTO 20
```

```
0 REM*******LASER CYCLE 4********
20 PLAY"L255O2CEGBO3DFAO4CEGBAFDO3BGECO2
AFD"
30 GOTO 20
```

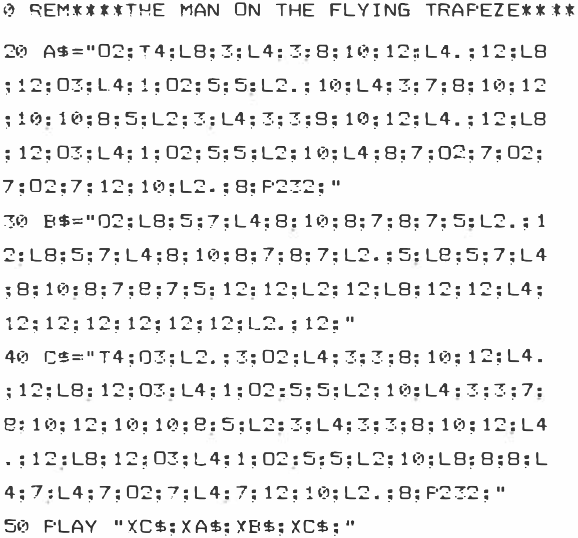
```
0 REM*******SOUND*******
20 PLAY"T200O4L40CL60EL80GL100BL140O5L18
0DL255G"
```

```
0 REM*******SOUND LOOP*******
20 PLAY"T200O4L40CL60EL80GL100BL140O5L18
0DL255G"
30 PLAY"P4":GOTO 20
```

Despite the rather 'arcade-type' titles given to the sounds, I'm sure that at least a couple will be of interest or use in your adventure programs. Admittedly, the computer is better at playing tunes than creating weird and wonderful effects.

For creating tunes, I have used the most effective method known to me, and this is using substrings. That is, letting a string equal a series of musical notes and then playing the string as a substring within the whole PLAY command string. This helps to remove endless repeats of a frequently-played bar and also allows you to experiment more.

Here are a few tunes below, just to whet your appetite. Be careful when you type in the strings as the octave is represented by 'O' (the letter) which should be distinguished from a zero, 'Ø'.



```
Ø REM****THE MAN ON THE FLYING TRAPEZE****
2Ø A$="O2;T4;L8;3;L4;3;8;1Ø;12;L4.;12;L8
;12;O3;L4;1;O2;5;5;L2.;1Ø;L4;3;7;8;1Ø;12
;1Ø;1Ø;8;5;L2;3;L4;3;3;9;1Ø;12;L4.;12;L8
;12;O3;L4;1;O2;5;5;L2;1Ø;L4;8;7;O2;7;O2;
7;O2;7;12;1Ø;L2.;8;P232;"
3Ø B$="O2;L8;5;7;L4;8;1Ø;8;7;8;7;5;L2.;1
2;L8;5;7;L4;8;1Ø;8;7;8;7;L2.;5;L8;5;7;L4
;8;1Ø;8;7;8;7;5;12;12;L2;12;L8;12;12;L4;
12;12;12;12;12;12;L2.;12;"
4Ø C$="T4;O3;L2.;3;O2;L4;3;3;8;1Ø;12;L4.
;12;L8;12;O3;L4;1;O2;5;5;L2;1Ø;L4;3;3;7;
8;1Ø;12;1Ø;1Ø;8;5;L2;3;L4;3;3;8;1Ø;12;L4
.;12;L8;12;O3;L4;1;O2;5;5;L2;1Ø;L8;8;8;L
4;7;L4;7;O2;7;L4;7;12;1Ø;L2.;8;P232;"
5Ø PLAY "XC$;XA$;XB$;XC$;"
```

```
0 REM********YANKEE DOODLE*******
20 A$="T2;V15;02;L8;3;8;8;10;12;8;L4;12;
L8;9;8;10;12;L4;8;L8;7"
30 B$="3;8;9;10;12;03;1;02;12;10;8;7;3;5
;7;L4;8;L8;8;P16"
40 PLAY "XA$;XB$;"
```

```
0 REM******76 TROMBONES******
20 A$="T4;02;L8;9;10;12;03;L4.1;02;8;5;L
4;3;L9;1;L4;01;12;02;L8;10;L4.;8;L8;8;10
;12;03;1;2;L4.;3;02;12;9;L4;6;L8;3;L2.;1
;L8;9;10;12;03;1;3;L4.;5;1;02;8;L4;5;L8;
1;"
30 B$="L4;6;L8;8;L4;10;L8;12;03;L4.;1;02
;L4;12;L8;10;L2;9;L8;12;03;L4;3;L8;3;L4;
1;02;L9;10;L2.;03;3;P8;"
40 C$="L4.;6;10;03;1;L4;02;12;L8;10;L2;9
;L8;12;03;L4;3;L8;3;L4;1;L8;02;12;03;L2.
;1;P9;"
50 PLAY"XA$;XB$;XA$;XC$;"
```

```
0 REM******THE ENTERTAINER*******
20 A$="L1503GG#A04FP1503A04FP1503A04FP5"
30 B$="FGG#AFGAP15EGP15FP5"
40 C$="04DC03B04DFAP12GEDGP5"
50 D$="04FGG#AFGAP15FGP15AFGAP15FGP15"
60 E$="AFGAP15EGP15F"
70 PLAY "XA$;XB$;XA$;XC$;XA$;XB$;XD$;XE$;"
```

When using the PLAY command, the notes can either be numeric (between one and 12) or just like musical notation (A, B, C, etc). The first three tunes use the numeric technique while the last tune uses musical notation.

Using the latter method, it is quite easy to transfer sheet music to your computer. Try to use single line melodies and preferably well-known tunes. One of the many music books for beginners contain many tunes perfect for your requirements. A final technical point, the semi-colons in between each note are mere spacers to make the lines more readable and do not have to be included.

You may think that the tunes are very nice, but what use are they to me in writing an adventure. A tune can really be of two uses. Firstly, as a theme tune at the beginning or the end of the game, or as a companion to a special display. Secondly, a tune can be a clue or puzzle itself. '76 trombones' is an ideal example. The tune is known but often people cannot put the title to it. You could, therefore, have a location that you were trapped in until you could guess the two-digit code that held the door shut. The correct code would be '76' and your only aid would be to hear a snatch from the tune. The number of guesses the player could have, before being trapped forever, would be severely limited to say three or four meaning that the player really had to know the tune before he or she could proceed any further with the adventure.

This is just one of the many possible things that you can do with sound in an adventure. Generally, sound is a good feature to add as it is relatively cheap on memory and if you don't like the final effect, then you can always turn your television down!

CHAPTER 8

HELP! A GUIDE TO THE ADVENTURES IN THIS BOOK

You must only read this chapter if you have failed despairingly to get into the adventures in this book. I deliberately withheld as much of the information about the adventures that was possible without spoiling the programming knowledge that you were given. Naturally, some of the surprises had to be given away to demonstrate the particular programming point being made at the time; however, as you will probably have found there were still many little traps for you to fall into. Here, I will give you a few tips to get you further into the adventures in this book. There are no extra clues given to the maze game, the only advice would be to construct your own map of the maze by constantly travelling around it. Then you will be able to travel to any point that you so require by referring to your map.

The first adventure, that I will give help on is the grid game, 'The Golden Chalice'.

With this game, the one thing that you must do is concentrate on the map at the beginning of the game. The symbols may seem odd, but are detailed below.

Inverse M	Monster.
'H'	Human (You).
')'	Rope.
Space	Roof opening.

'?'	Torch.
'O'	Flood.
'X'	Quicksand.
'#'	Map.
'$'	Chalice.

In the brief time that the map appears on the screen, try to memorise where the map is. If you collect the map, you can then move around the adventure and when lost, for a fee of three hours, see the mess that you have got yourself into. Try to write everything down and remember that you need four of the five pieces of rope.

As this adventure is largely random, that is all the help I can give other than to say watch the screen at all times.

'The Dark Forest' is a more structured and difficult adventure to solve. Again, there are some parts that are random in design but the locations and the links between them are always the same. Below are some tips.

(1) Do not go north from the beginning or you will land up in prison.
(2) You have a better than average chance of benefiting from the opening of a possession in all bar extreme circumstances.
(3) Your starting figures of money, strength and magic will almost certainly be not enough to vanquish the Creature at the end of the game. Therefore, you must do your best to obtain as many extra points (mainly through possessions) as possible.
(4) When fighting, choose your strongest attribute, for if you lose, the game is over. Note that whatever the battle level of the Creature, the magic level is divided by ten. Magic is often the best attribute to use.
(5) The battle strengths of the monsters are: Sir Rufus Of Gascony, 20 (weak); A Ferocious Bear, 20 (weak); A Band Of Robbers, 40 (strong); Grendal, The Hermit, 40 plus a random element of up to 15 (strong and unpredictable); and the Black Knight, 60 (keep well away from him, this enemy will give anyone a terrible mauling).
(6) The bridge guarded by the troll marks the only way to the Creature's lair. You must offer at least 40 (preferably more)

pieces of gold to get yourself through. If the troll rejects your offering then you are moved to a distant and tricky part of the adventure.

(7) The cave's entrance is to the south, the clue you are given is just to slow you down considering the answer.

(8) The well location is interesting. You are asked if you wish to go down the well. There is a large chance that you will be drowned as the computer's reckoning that the well is empty is not to be trusted here. If you do go down the well and are safely at the bottom, you will see an ornament with the magic word 'GRANDOS' inscribed on it. The ornament is the Ram of Anag, think about it . . . The Anag-Ram! Therefore, the magic word is not 'GRANDOS' but an anagram of that word, DRAGONS. This is the code word that should be used when confronting the woodcutter's hut.

(9) If you have been round the adventure, collected the Ram of Anag and have landed in the Black Knight's prison, do not worry. The computer will ask you for a final command. If you enter the code word, then you can escape. It's no use just entering the code word if you do not own the ornament – the computer recognises cheats!

(1Ø) Try to boost up your money and spells in particular. If at any time your strength appears to be getting low, you can have a rest which will revitalise you. Spells are good for fighting your adversaries and are particularly effective against the Creature.

(11) Try to codify your movements. Wandering aimlessly around the adventure is not a very good way of attempting to solve it. Write down on paper your movements, actions and try to construct a scenario map. These will all aid you in trying to solve the adventure.

A much tougher adventure than the previous one, 'The Nielson Papers' will keep you adventuring for some time. Here are some hints to aid you in your progress. Be warned, I have not told you everything!

(1) From the opening location, you can only move north.

(2) Do not fire your gun without the silencer, unless it is a real emergency. The sonic alarm in the building will pick up the

sound and will alert the security forces. You will then have little time to escape (another 15 moves or so).
(3) There are 30 or so commands for you to find, understand and use. A list of them is below:

GO
MOVE – Allows you to travel around game, use N, S, E and W.

TAKE
GET
STEAL – Collect a movable object.

DROP – Self-explanatory – this command allows you to discard an object.

READ – Read certain messages.

CLIMB – Climb up to the shelf with papers on, but you must have the stool or ladder to do so.

CUT – Cut fence with wirecutters. Care should be taken here, without gloves you will be burnt to a crisp.

CHANGE – Allows you to change money to coin for the executive washroom.

OPEN
UNLOCK – If carrying the key these commands allow you to unlock doors but must be used discriminatingly. Some of the doors are very old.

FIRE
SHOOT – If carrying a gun, these commands allow you to fire at a security guard.

WAIT – Simply, a pause.

DRINK – Allows you a drink, if you have a cup.

SEARCH
LOOK – Allows you greater detail of a location than the normal description. Must have torch to do this.

WEAR – Allows you to wear the gloves.

COMPUTE
CALCULATE
DECODE – Allows you to work out the coded memo or the computer-controlled lock in the document room.

HELP – A simple plea for assistance.

CLUE – Another plea for assistance!

QUIT – Type this when you're fed up with the game.

PUT – With this command, you can put the money into the coin-operated lock and attach the silencer to the gun.

HIT
FIGHT
ATTACK
KILL – These commands all allow you to vanquish a security guard. If you do not have the rope or wirecutters, then you will fight with your hands.

(4) There are 18 objects, only a couple like 'paper-clips' and 'papers' are totally useless and have been put in as red herrings to confuse you.
(5) The key which is in the dark corridor opens all the locked doors.
(6) If you get as far as escaping, you need to get to the fence at the back of the delivery room. You will need the wirecutters to snip through the fence and also the special gloves to stop you being electrocuted.
(7) Once in the document room, you will need a stool or ladder to climb to the top shelf.
(8) Try to decode the memo, its advice is pretty important.
(9) Once in the maintenance area, going west will just take you further into it.
(10) Be careful when around the vault because if you get locked in there without the key, your game is over.
(11) Plan your escape carefully. Once you have taken the papers, the alarms will sound and your time left to move becomes very short.
(12) Finally, as with 'The Dark Forest', always map out the adventure. By creating a plan or map of the adventure, you can travel to any required location easily. It is the only way to solve the adventure.

Much aid has already been given on 'The City Of Atlantis' adventure in the form of a map, a list of commands and a list of the objects. Here though, are a few extra points of note.

(1) If you bump into a wall in room 1, the Atlantan is distracted with laughter and you can then go east to room 2.
(2) In the Radio Station, do not stay for too long – some Atlantans may hear you over the radio and come and investigate.
(3) Search the scientist in room 2 and you will be given something to alleviate your problems with the thin atmosphere. You will then have enough strength to fight the gladiator in room 3.

(4) Try tosteer clear of any monsters until you think you know how to dispose of them. Remember, that in most cases, the 'baddies' are blocking or hiding a course of action open to you. Many of these courses of action are essential to your escape from Atlantis.
(5) A cryptic clue to finish with: remember your horticultural knowledge, it will come in useful at some point . . . .

CHAPTER 9

SOME IDEAS FOR YOUR OWN ADVENTURES

After reading through all of this book, I hope you will be inspired to rush off and write your own adventure program. Sometimes, thinking up a plot with all the characters, objects and locations can be difficult. Below, I have illustrated briefly a few ideas that you could possibly use.

(a) **The Plot**
Obvious ones are often the best, such as stealing the treasure, battling against some grotesque creature to save planet Earth and rescuing your imprisoned friends. Most adventures involve you finding something – the actual item can be almost anything, such as a lost city or the last dilithium crystal left that will enable your stranded spacecraft to blast off back to your home planet.

More difficult to think of is the actual setting that your adventure is to be in. Try at first to narrow it down to an area, eg. space, medieval, underground, everyday, desert, ocean, recent history, and the like. Everyday times can often be a fascinating area to place your adventure in. Most people expect the medieval- or fantasy-type game when they settle down to play an adventure, so you can imagine their surprise when they find the first location to be on a number 73 bus going towards the town centre! My game, 'The Nielson Papers', uses a fairly everyday setting but with a 'secret agent' approach. For extra interest, you could possibly merge a couple of these ideas and form a rather surreal adventure with three or four totally different settings in the one adventure.

Another possibility concerns a more personal adventure written around your own home. The characters could be people that you know and you could add a few locations if you do not have enough rooms to fill up your adventure. Another area that you could base your adventure on is from a book or story that you have read. Of course, by doing this you will have to keep it for your own personal use as copyrights on all written material are very strict and usually enforced heavily. If you are writing an adventure just for yourself, then using a well-known story gives you many of the locations and characters needed.

(b) **Locations**

Thinking up 20 or 30 location names is not always an easy task, especially when they all have to tie up with the adventure. In all adventures, there are a few locations that perform little or no task and are just there for you to travel through. Below are a list of descriptions that can be used for this purpose.

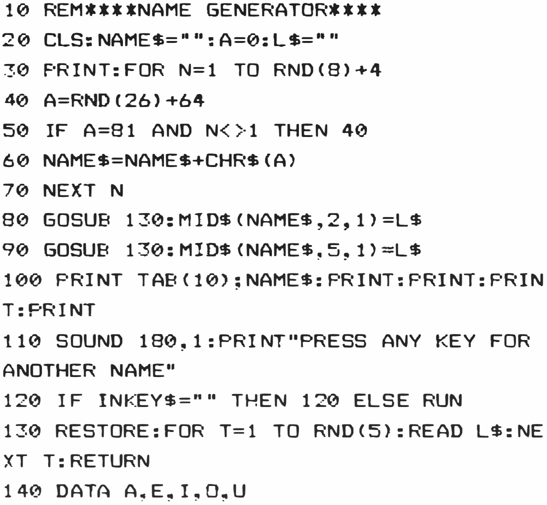
A dark, dank storeroom.
A spacious hallway, richly decorated with paintings.
A musty corridor.
A windswept garden, covered in weeds.
A clean, white room sparsely furnished.
An area of strange, tangled undergrowth.
A vast storage room, cold and uninhabited.
A small meadow of no significance.
A gently rising hill.
A crumbling building built some time ago.
A collection of ruins.

These are just a few of the possible descriptions that you could use. They are fairly non-specific and are short, so that if detail needs to be added then, you will find room in the memory to do so.

You will need some very specific locations to tie in with the whole aim of your adventure and these can be obtained from your plot quite easily.

(c) The Characters

Names of creatures, both good and bad, are not that hard to create, but why work yourself at all when you have a powerful computer nearby. The program below generates random creature names. The combinations of letters are not totally random as there are several lines that guarantee a vowel every second and fifth letter and a line to stop a 'Q' coming anywhere except at the front of the word. It's only a simple program but you may get a few useful names from it.



```
10 REM****NAME GENERATOR****
20 CLS:NAME$="":A=0:L$=""
30 PRINT:FOR N=1 TO RND(8)+4
40 A=RND(26)+64
50 IF A=81 AND N<>1 THEN 40
60 NAME$=NAME$+CHR$(A)
70 NEXT N
80 GOSUB 130:MID$(NAME$,2,1)=L$
90 GOSUB 130:MID$(NAME$,5,1)=L$
100 PRINT TAB(10);NAME$:PRINT:PRINT:PRIN
T:PRINT
110 SOUND 180,1:PRINT"PRESS ANY KEY FOR
ANOTHER NAME"
120 IF INKEY$="" THEN 120 ELSE RUN
130 RESTORE:FOR T=1 TO RND(5):READ L$:NE
XT T:RETURN
140 DATA A,E,I,O,U
```

For those of you with a printer that are even more lazy, why not type in the following program which dumps a whole catalogue of names onto paper. If you have a printer with a buffer, it is advisable to press the BREAK key after some time, as the printer will have stored quite a few names in its memory that have not yet been printed out. Whenever you press the BREAK key, you should expect quite a few more names to be printed, the exact number depending on the size of the printer buffer.

```
10 REM****NAME ON PRINTER GENERATOR****
15 PRINT#-2,""
20 PRINT#-2,"***********CREATURE NAME LI
ST***********":PRINT#-2,""
30 NAME$="":A=0:L$=""
40 FOR N=1 TO RND(8)+4
50 A=RND(26)+64
60 IF A=81 AND N<>1 THEN 50
70 NAME$=NAME$+CHR$(A)
80 NEXT N
90 GOSUB 130:MID$(NAME$,2,1)=L$
100 GOSUB 130:MID$(NAME$,5,1)=L$
110 PRINT#-2,NAME$;"      ";
120 GOTO 30
130 RESTORE:FOR T=1 TO RND(5):READ L$:NE
XT T:RETURN
140 DATA A,E,I,O,U
```

A very large percentage of the names formulated will be unusable, but you only need to gain a few names to have made it all worthwhile. On the next page is a sample printout of some words generated; I have ringed those which may be suitable, but you may have other ideas and prefer some I have ignored – just have a look ....

```
***********CREATURE NAME LIST***********
BAOOUI      AECUERUWG      JECYUUK      XOF
FUODDLMFK      OOLLIFDXDJIC      ZUXNEYFDN
    CIMJOB      JIMZUDWNDPX      XILYEMGC
D     ZOZJIERCAN      QIPSOOWZJBF      TUR
TISE      RIADIOYDVJXF      COEFOYXCZR
 DEFUUEXUT      ZUYTES      GEYIAPUPJWR
  PAXKEFGUN      LOUNADYWPF      EIMLIWY
```

UICMAABIXSZ   RUZIOWTR   XUHWIZPZ
WUGD   KIMSECTJOEK   XORYIUNIA   G
OTBUWOW   VURGU   KICVAUJNJ   HOLL
EGI   GEPVITK   IAACUWC   PAYWIETM
WML   WAFHIEWZ   BEXXATHVJRE   ZAF
WEIV   RAOUOJFRZT   CUFMI   XOPPID
VWNKM   QOWLOL   QAPEUTHF   KIBLA
REMCOD   IOMZAIP   XECRARZKB
WUNZIU   FOOJE   SODVISOEEV   HOU
XAHGU   GAKZIFT   KAHTO   WIPBALTM
YSLN   FUKLETLZ   XOTKEOZ   DURSUD
SFJTPB   OEGWILA   HEKCU   KAECU
IODZIYG   IITCERKIONH   MAJRAEZYL
Y   RAXOABBBWZ   CUJJUNDS   AOBKOR
PTFKH   IOUFEARCAJ   AOWNO   EOFNU
B   QIMDIBTUWZKK   KEVDACX   AAGJI
BFRU   VAYEAFSWYUC   OIMAEJFUJT
XERVO   IUJJAZLR   VITTINNDHZ   WA
NFIU   VOAUAF   CUSRAFPRO   NATII
UUBRO   GENLUH   DUTMUY   GIKB
IG   DAEDEV   ROPJULPST   PUELAKDU
O   GOYMILTBTLR   GUSEI   GAZAOWU

SOME USEFUL BOOKS

Despite its popularity, there are few books of note on this subject. I have searched the bookshelves for any good volumes and below are, in my opinion, the most informative and useful.

WRITING BASIC ADVENTURE PROGRAMS FOR THE TRS-80

Frank Dacosta. Tab books. ISBN 0 8306 1422 2.
Despite the book being set around a different computer, this is a most useful guide. Lots of clever programming tricks described in a light informal format make this a 'must' for adventure writers everywhere on any machine.

THE ZX81 POCKET BOOK

Trevor Toms. Phipps Associates. ISBN 0 950 7302 2.
30 pages of adventure programming make this worth considering, particularly if you are shrewd enough to pick it up at a discount price. Many shops are selling ZX81 material cheaper and you could well benefit from this. Trevor Toms uses a more unorthodox framework for adventures and if you have previous knowledge of a ZX machine, then you will find little difficulty in converting it to your new computer.

ADVENTURE WRITING

Aardvark-80,2352 S. Commerce, Walled Lake, MI 48088, USA.
This 16-page booklet selling in the United States of America (for the exorbitant figure of five dollars) is a terrific help to all adventure writers. The adventure program included, 'Death Ship', is broken down in detail and illustrates the programming techniques used very well. Recommended.

CREATING ADVENTURE GAMES ON YOUR BBC MICRO

Ian Watt. Interface/Addison Wesley. ISBN 0 201 14678 9.
Ian really does know his stuff when it comes to writing adventure programs. In this book, there are three full scale adventures all written with a skill that makes them very difficult to beat while still being fairly simple to understand in terms of programming used.

Halfway through a complex maze or in the middle of a fight with a fearsome dragon, haven't you ever wondered what it would be like to create your own adventure using your ideas for locations, creatures and obstacles? Well, if you own a Dragon computer then here is your chance.

This book takes you stage by stage through the world of computer adventure gaming, from the very basics to the more advanced concepts. Along the way you will be shown how to construct game maps, insert problems for the player to solve and learn how to use all the techniques that are the difference between an average adventure and an excellent one. In addition, there are no less than five full adventures for you to type in and enjoy, ranging from the visually stunning 3D Maze to the hideously complex 'Nielson Papers'.

Written by a dedicated adventurer who has a regular monthly column in a Dragon publication just on this subject, the book also features a chapter on utilising sound and graphics in your adventures and a number of other games including 'Swarm' and 'Four by Four' making this the best value Dragon book around.

Another great book from

INTERFACE
PUBLICATIONS

CREATING ADVENTURE GAMES ON YOUR DRAGON 32 CLIVE GIFFORD

IP