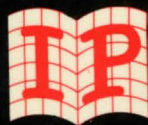


Creating Adventures on your Commodore 64



Clive Gifford
Robert Young

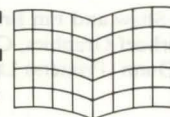
Contents

Creating Adventures on your Commodore 64

Interface 'Success in the Fast Lane' programming series

Foreword by Tim Hartnell

INTERFACE
PUBLICATIONS



London and Melbourne

First published in the UK by:
Interface Publications,
9-11 Kensington High Street,
London W8 5NP

Copyright © Young/Gifford 1984

ISBN 0 907563 49 X

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

ALL RIGHTS RESERVED

No use whatsoever may be made of the contents of this volume - programs and/or text - except for private study by the purchaser of this volume, without the prior written permission of the copyright holder.

Reproduction in any form or for any purpose is forbidden.

Books published by Interface Publications are distributed in the UK by WHS Distributors, St. John's House, East Street, Leicester LE1 6NE (0533 551196) and in Australia and New Zealand by PITMAN PUBLISHING. Any queries regarding the contents of this volume should be directed by mail to Interface Publications, 9-11 Kensington High Street, London W8 5NP.

First published in Australia by Pitmans Publishing Pty Ltd.
158 Bouverie Street, Carlton, Victoria 3053

1st Printing in the UK - June 1984

For Mum and Dad

Artwork by Peter Shaw and Fin Bristow
Typesetting by Verhoef Graphics, Clayton
Printed by J. W. Dunn (Printers) Ltd., Sutton, Surrey.

Contents

Foreword	iv
Introduction	1
Chapter One — Introduction to adventures and adventure programming	3
Chapter Two — A sample adventure, the dark forest	11
Chapter Three — A full-scale text adventure, The Nielson Papers	59
Chapter Four — Other forms of adventure ...	89
Chapter Five — The City of Atlantis	103
Chapter Six — Other games for your computer	123
Chapter Seven — Sound and graphics for your own adventures	141
Chapter Eight — Help! A guide to the adventures in this book	155
Chapter Nine — Some ideas for your own adventures	165

Foreword

Here, at last, is the key to creating and playing adventure programs on your Commodore 64. Written by two highly-experienced authors and adventurers, this book will take you further into the fabulous world of computer adventuring than you had dared hope.

You'll learn how computers organise the imaginary environments for adventures and how you can create your own, so they are logical and consistent — yet extremely difficult and challenging to solve. This book will show you how to write program lines which allow your Commodore 64 to understand words like "GO", "PICK UP", "UNLOCK", "GET" and all the rest which are needed so that the computer will follow player's commands.

Much more is given relating to the **mechanics** of writing adventure games. But, as well as this, you'll find a host of information to help you solve and enjoy commercial adventures.

And, best of all, there are four **complete** adventure programs, just begging to be typed into your Commodore 64 and solved.

The book also contains information on sophisticated graphics handling (including sprites) plus sound effects routines to incorporate into your own programs.

Time to buckle on your armour, and enter the fascinating world of Adventure gaming.

Tim Hartnell,
London, 1984.

Introduction

There can be no doubt that adventure games played on computers are 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 centre of the earth and transport you into a darkened maze of terrifying complexity, all at a touch of a key.

To many people, the real enjoyment is in trying to win the game. They certainly are a challenge to play. To find all the objects needed and to reach your goal is a task often requiring much thought. Adventures are much 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 one, you realise how complex the 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 four adventures in this book plus a whole host of adventure utilities and three non-adventure games.

The whole aim of this project is to help you get to the stage of being able to write your own adventure games as quickly as possible.

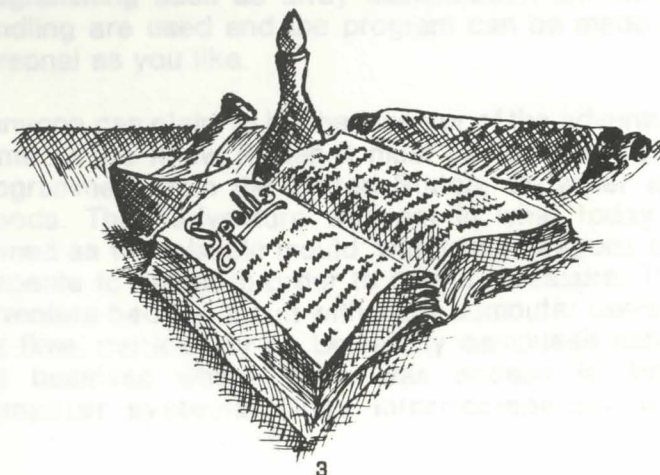
There is little more to say, save that by reading through the book in order, you will gain the most from it. Finally Good Luck and Happy Adventuring!

Robert Young,
Clive Gifford,
1984.

CHAPTER ONE

Chapter One

An Introduction to Adventures and Adventure Programming



Robert Young
Give it a try and you will gain the most from it.
A Really Good Little and Happy Adventure!

Chapter One

An Introduction to Adventures and Adventure Programming



CHAPTER ONE

Adventures and Adventure programming is a huge area of computer entertainment that only recently has been exploited to its full capability. 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 has attracted thousands of computer owners.

Adventure games can offer a **real** challenge to those who want more from computer gaming than blasting spaceships out of existence. The range of adventures available for your computer is wide-ranging and contains something to interest and amuse everyone.

From a programming point of view, adventures are ideal projects for the new programmer. After fiddling around with a few very simple programs, you may find you want 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, many areas of simple BASIC programming such as array manipulation and string handling are used and the program can be made as personal as you like.

If anyone can claim to be the inventor of the adventure game as we know it then it must be two mainframe programmers from the United States, Crowther and Woods. Their adventure was set in what today is termed as the 'classic mould' with trolls, dragons and serpents to defeat in order to capture treasure. This adventure became a cult with many computer users of the time, particularly on University campuses and in big business where there was access to large computer systems. Once microcomputers with

sufficiently large memories came along, the adventure programs were scaled down to fit.

If you ask many people what is a computer adventure, they will answer that it is a game with monster, 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 home base. This is the 'classic' type of game following Woods and Crowther's model. There is now, of course, a much wider range of adventure games available with no mythical beasts, no treasure to collect and no objects to even contemplate picking up. Perhaps that is the beauty of adventure writing; almost any scenario can be used and any method of play can be adopted to suit the programmer's whim.

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

An adventure must have:

- a) Locations (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, e.g. escape or the accumulation of treasure. Without any objective, an adventure game would become pointless and boring.
- d) Some form of obstacle to hinder your progress, e.g. monsters, locked doors.
- e) Some element of discovery.

- f) Some form of command with which the adventurer can direct himself around the locations.

These different factors all seem essential in order to have a proper, playable and entertaining adventure.

How do you go about writing an adventure? The chapters following this attempt to answer this with practical examples, but let's consider a little theory first.

Structured Programming

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.

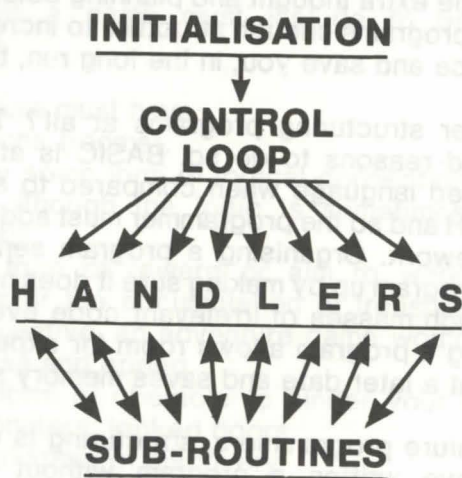
Why bother structuring programs at all? There are many good reasons to do so. BASIC is at heart an unstructured language when compared to something like FORTH and so the programmer must add his or her own framework. Organising a program sensibly can speed a program up by making sure it does not have 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 can be very difficult. Just by looking at the program listing, you may not know 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 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 variables, arrays and other parameters to their starting values. The control loop is the vital part. The program constantly cycles through this waiting for a player's input and then directing the program flow to one of the many handlers. The handlers cover all the actions and tasks that 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, send action back to the control loop and the process starts again.

It should be noted that this is just one method of structuring your adventure. There are lots of different ways which may include not having a bank of subroutines but incorporating them into the handlers.

Planning

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

Once you have written your parameters down, 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 program lines for the elusive bug. It is 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 Two

A Sample Adventure, The Dark Forest



Chapter Two

A Sample Adventure, The Dark Forest



CHAPTER TWO

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 him to search for and destroy The Creature, a strange beast that no-one has ever seen but who rules the county of Ramshire by terror. Once at the forest, your adventure has begun. You must find enough treasure and magic influence to survive, 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. Many fierce animals as well as the infamous Black Knight live in this area and will do their best to overcome you. Good Luck and happy fighting!

I do not want to give many of the secrets away as without them, the adventure loses a lot of 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 'N' but 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 you to relax and increase your strength. However, each hour's peace costs a certain amount of cash, so you must 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 option 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 — 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 the Nielson Papers. The locations and the logic linking them up is always the same. This adventure could, perhaps be a little more structured than it is but as it stands, the game runs smoothly. I know where all the routines lie and you will easily be able to glean all that you need to know from the listing.

At the end of the 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 organised programming it would have been impossible to split the program up into neat blocks at all. The program sections have been given the following titles: Initialisation; Control Loop; Monsters and Fight; Attributes and Possessions; Locations; The Final Confrontation; and Win or Lose. This is the order that we will look at the program sections.

Initialisation

Initialisation in this program is in two blocks, the initialisation subroutine and the start game sequence. We will look at the short game sequence first. It is

shown below:

```

10 REM*****
20 REM*** THE DARK FOREST ***
30 REM*****
50 GOSUB 2630
55 POKE 53280,0:POKE 53281,11
60 PRINT "s":PRINT "QQQQQQQQQ1111111111
1pTHE DARK FOREST"
65 REM CLEAR SCREEN DOWN CURSOR RIGHT CU
RSOR BLACK
70 PRINT "Q111111111pPRESS ANY KEY TO S
TART"
75 REM CURSOR DOWN RIGHT CURSOR BLACK
80 GOSUB 3320
90 GET A$:IF A$="" THEN 90
100 PRINT "Q11111111EYOU HAVE DARED TO E
NTER...":FOR DD=1 TO 1000:NEXT DD
110 PRINT "Q111111111111GOOD LUCK"
120 FOR DD=1 TO 1000:NEXT DD

```

The first genuine program line, line 50 goes to the initialisation subroutine from 2630 onwards. The next few lines set screen and border colours, clear the screen, set up the title page and wait for a key to be pressed. Once a key is pressed (the A\$ 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:

```
2620 REM *** INITIALISE ***
2630 DIM M(3):DIM Z(21,5):DIM Q(5,5)
2640 M(1)=INT(30*RND(1))+21
2650 M(2)=INT(70*RND(1))+11
2660 M(3)=INT(6*RND(1))+1
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=INT(18*RND(1))+2
2880 IF Z(RO,5)=0 THEN Z(RO,5)=A:GOTO 2890
2885 GOTO 2870
2890 NEXT A
2892 VO=54296:AD=54277:SR=54278:WL=54276
:HF=54273:LF=54272
2900 RETURN
```

Line 2630 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 given values in lines 2640, 2650 and 2660. The section of code between 2670 and 2800 reads the values of the room data into the room array Z. There are 21 rooms each having five numbers assigned to them. 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 1 and the variable used for the special idol, P, is set to 0.

Finally, in this section, the starting positions of the five monsters are determined by the lines 2860 to 2890. The fifth number in the room array data is set to a value other than 0 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 most vital component of this game. The functions that it performs are wide-ranging and are detailed below.

The first two lines perform the presentation 'niceties', that is clearing the screen, pausing the program, playing a quick note (GOSUB 3370) and printing up the attribute values (the GOSUB 730 handles that).


```

130 FOR TD=1 TO 1500:NEXT TD:PRINT "␣":R
EM CLR SCRΝ
140 GOSUB 3370:GOSUB 730:PRINT

```

The player's response is handled by line 280 which then sends the computer 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 150 to 180 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 which has a rough relationship to the number of hours 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 "110.K. YOU
CAN SLEEP HERE BUT IT WILL"
201 REM RIGHT CRSR
205 PRINT "COST YOU":CS;"COIN/S FOR EVER
Y HOURS UNINTERRUPTED REST":PRINT
210 INPUT "HOW MANY HOURS":H
215 IF H<1 OR M(2)-(H*CS)<0 THEN PRINT "
TRY AGAIN SLEEPY!!":GOTO 210
220 M(2)=M(2)-(H*CS):M(1)=M(1)+H:PRINT "
sQQQQQQ1111111111ZZZZZZZZZZ"
221 REM CLR SCRΝ DOWN CRSR RIGHT CRSR
222 FOR DD=1 TO 500*H:NEXT DD:PRINT "␣":
GOSUB 730:GOTO 250:REM CLR SCRΝ

```

The final lines in the control loop are particularly concerned with locations. Lines 250 to 254 direct the program to the correct location subroutine. Line 260 checks to see if there is a monster in the same room as you and 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 the 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 he asked 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
THAT DIRECTION":GOSUB 3150:GOTO 260

```

```

240 CHK=B
250 IF B>10 THEN GOTO 254
252 ON B GOSUB 980,1000,1020,1170,1200,1
250,1270,1290,1310,1420:GOTO 270
254 B=B-10:ON B GOSUB 1610,1630,1660,168
0,1700,1850,1880,1910,1930,1950:B=B+10
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 this routine. The lines up to 360 check to see which monster is in the location and print 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 ROB
BERS SEIZE YOU":E=40
330 IF MN=2 THEN GOSUB 3410
340 IF MN=3 THEN PRINT "SIR RUFUS OF GAS
CONY ATTACKS...":E=20
350 IF MN=4 THEN PRINT "A FEROCIOUS BEAR
POUNCES":E=20
360 IF MN=5 THEN PRINT "YOU HEAR THE WAR
CRY OF THE 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 to decide on a direction. The answer is then handled in the same way as any normal direction command. Four points are taken from the player's strength level as well.

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$:F$=
LEFT$(F$,1)
380 PRINT:IF F$="F" THEN 410
390 IF RND(1)>0.4 THEN INPUT "WHICH DIRE
CTION, COWARD":N$:M(1)=M(1)-4:GOTO 130
400 PRINT "NO! YOU MUST STAND AND FIGH
T"
405 PRINT "YOU LOSE 10 STRENGTH":M(1)=M(
1)-10

```

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 then constructed. A simple bar graph, for want of a better name, is displayed with a bar for the player's attribute and a bar for the monster's attribute. These bars are sized in proportion to the actual level of the respective attribute. Line 510 handles any attribute that is larger than sixty.

```

410 REM *** FIGHT ***
420 Z(B,5)=0:RO=INT(19*RND(1))+1

```



```

430 IF Z(R0,5)=0 THEN Z(R0,5)=MN:GOTO 440
435 IF Z(R0,5)<>0 THEN GOTO 420
440 PRINT "WHICH ATTRIBUTE DO YOU WISH TO FIGHT WITH (ENTER WORD)"
445 INPUT 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 2910
490 IF A=2 AND M(2)<1 THEN PRINT "CANNOT USE MONEY":GOTO 440
500 IF A=3 AND M(3)<1 THEN PRINT "CANNOT USE MAGIC"
510 PRINT "s":IF M(A)>60 THEN DD=M(A)-60:M(A)=60
515 REM CLEAR SCREEN
530 PRINT "sQ1111111111RzTHE BATTLE COMMENCES.."
535 REM CLR SCR DOWN CRSR RIGHT CRSR RE U ON LT BLUE
540 PRINT "QQ YOU: ";
541 REM DOWN CRSR PURPLE YELLOW
545 FOR TX=1 TO M(A):PRINT "q":NEXT:REM SHIFT KEY + Q KEY
546 PRINT
548 PRINT "QQrENEMY: ";
550 FOR XT=1 TO E:PRINT "q":NEXT:REM SH IFT KEY + Q KEY
560 GOSUB 3390

```

The fight continues with one being docked from each of the opponents' scores until one of the attributes reaches zero. If the player has won, then he will be granted extra points for one of his attributes. If he loses, the game is over. The bar chart is updated every cycle of the loop, with the bars getting smaller.

One additional point of interest in this part of the program is the way the fight is weighted in favour of the player. You can see in line 610 that 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 will not be resigned to defeat but will be waiting to see if he can overcome the monster's advantage.

```

570 REM *** MAIN FIGHT LOOP ***
580 IF INT(7*RND(1))=4 THEN GOTO 600
590 E=E-1
600 GOSUB 3390
610 IF INT(3*RND(1))=1 THEN GOTO 630
620 M(A)=M(A)-1
630 GOSUB 3390
640 IF E>0 AND M(A)>0 THEN GOTO 530
650 IF M(A)<1 THEN GOTO 2910
660 PRINT "QQYOU WIN!!QE":REM DOWN CRSR WHITE
670 GOSUB 3390
680 ER=INT(40*RND(1))+1:BU=INT(7*RND(1))+1
690 IF BU<4 THEN PRINT "YOU GAIN":ER;"STRENGTH":M(1)=M(1)+ER+DD
700 IF BU<7 AND BU>3 THEN PRINT "YOU GAIN":ER;"MONEY UNITS":M(2)=M(2)+ER+DD
710 IF BU=7 THEN ER=INT(ER/10)+1:PRINT "YOU GAIN":ER;"MAGIC SPELLS"
715 IF BU=7 THEN M(3)=M(3)+ER+DD
720 FOR T=1 TO 1500:NEXT T:PRINT "s":GOTO 250:REM CLEAR SCREEN

```

Attributes and Possessions

Lines 730 to 780 make up the attributes routine. They have been defined in the initialisation section and are modified in the rest of the adventure. These attributes take the place of objects which can be handled in other adventures. A quick glance at the attributes can give you an idea of how well you are doing and in which areas, if any, you are weak.

The routine simply takes one away from the strength attribute and then displays the attributes. If the strength is less than one, then the program jumps to the 'lose routine' from 2910 onwards.

```
730 REM *** ATTRIBUTES ***
740 M(1)=M(1)-1: IF M(1)<1 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.

```
790 IF RND(1)>.8 AND B<>1 AND B<>15 AND
    B<>20 THEN PS=INT(5*RND(1))+1: GOTO 810
800 RETURN
810 IF PS=1 THEN PRINT "AN OLD GLASS BO
    TLE IS WITHIN REACH. IT IS COVERED";
```

```
815 IF PS=1 THEN PRINT " IN COBWEBS"
820 IF PS=2 THEN PRINT " HERE IS A LARGE
    OAK CHEST. A WEIRD"; PRINT "SIGN IS";
825 IF PS=2 THEN PRINT " CARVED ON THE L
    ID"
830 IF PS=3 THEN PRINT " A VELLUM BOOK L
    IES BY YOUR FEET"
840 IF PS=4 THEN PRINT " A GOLD LOCKET O
    N A CHAIN IS CLOSE TO"; PRINT "YOU"
850 IF PS=5 THEN PRINT " BEFORE YOU IS A
    BOX OF EXOTIC FRUIT"
```

The next two program lines accept the player's decision to open the particular object or not. If the player is brave and decides to open the object, then there is a three in ten chance of an Orc jumping out and attacking. This chance is handled by lines 890 to 896.

The rest of the routine is concerned with printing, and altering, the attributes of the player according to which object he or she safely opened. The descriptions are quite detailed and add a little extra realism to the game. As opposed to having a large pause loop, we choose to let the player press a key to continue. This is handled by lines 950 and 960.

```
870 INPUT "DO YOU WANT TO OPEN IT"; OP$
880 PRINT: IF LEFT$(OP$,1)<>"Y" THEN RETU
    RN
885 IF RND(1)<.7 THEN GOTO 900
890 PRINT "AN EVIL ORC LEAPS OUT AND ATT
    ACKS YOU"
894 FOR TT=1 TO 1000: NEXT TT: PRINT "AFTE
    R A BRIEF STRUGGLE IT RUNS AWAY"
```



```

896 PRINT "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 THE":PRINT "ESSENCE":
905 IF PS=1 THEN PRINT " GIVES YOU NEW S
TRENGTH AND LIFE":M(1)=M(1)+8
910 IF PS=2 THEN PRINT "MUCH TREASURE IS
CONTAINED IN THE CHEST"
915 IF PS=2 THEN PRINT "YOU ARE A RICH M
AN":M(2)=M(2)+35+INT(15*RNDC(1))
920 IF PS=3 THEN PRINT "IT CONTAINS 6 PO
WERFUL SPELLS":M(3)=M(3)+6
930 IF PS=4 THEN PRINT "IT HAS A SMALL P
OTRAIT OF THE BEAUTIFUL"
935 IF PS=4 THEN PRINT "PRINCESS CAROLIN
E IT IS QUITE VALUABLE":M(2)=M(2)+10
940 IF PS=5 THEN PRINT "YOU ARE SO TEMPT
ED THAT YOU EAT SEVERAL"
945 IF PS=5 THEN PRINT "THEY ARE DELICIO
US AND ADD TO YOUR STRENGTH"
947 IF PS=5 THEN M(1)=M(1)+12
950 PRINT:PRINT "***** PRESS ANY KEY TO
CONTINUE *****"
960 GET A$: IF A$="" THEN 960
965 PRINT "␣":RETURN:REM CLEAR SCREEN

```

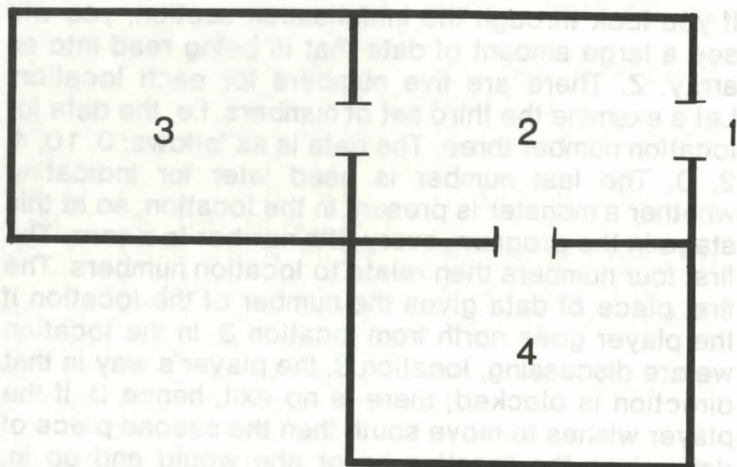
Locations

This is the most important part of any adventure. Without locations to move between, there would be no exploration and little discovery, and these two elements are vital to make an entertaining adventure.

You need to formulate a system of logical connection between each location and its exits. The adventures in this book show you several versatile ways of constructing 'room logic'. 'The Dark Forest' uses one of these systems.

If you look through the initialisation section, you will see 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, i.e. the data for location number three. The data is as follows: 0, 10, 4, 2, 0. The last number is used later for indicating whether a monster is present in the location, so at this stage in the program, every fifth number is a zero. The first four numbers then 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 discussing, location 3, the player's way in that direction is blocked; there is no exit, hence 0. If the player wishes to move south then the second piece of data gives the location he or she would end up in, location 10. 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 150 to 180 in the control loop, you will see how this information is interpreted when the player wishes to move. Line 150 simply means, 'If the player requests to go north and the first item of data in the present location is not equal to 0 (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 0, 0, 0, 2

Location 2 0, 4, 1, 3

Location 3 0, 0, 2, 0

Location 4 2, 0, 0, 0

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. It is a good idea to remember the locations as simple rooms with open doors in certain directions. When designing your adventure, map out your locations first on paper. You will then find it easier to code in a data list and will also have at your disposal a complete map of your adventure.

This will be very useful when testing all the parts of the adventure, or when a friend is playing the game and is in trouble.

Many extra options can be attached with this system. 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 in regard to the other piece of data.

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 of this system are very wide indeed.

Once the player has moved to a new location, the computer ON... GOSUB's to the line number dealing with that location.

The locations follow. Most are self-explanatory, so it is only necessary for us to give details if the location is particularly complex.


```

970 REM *** LOCATIONS ***
980 PRINT "YOU ARE IN A SMALL CLEARING I
N THE DARK FOREST YOU CAN PROCEED IN";
985 PRINT " ANY DIRECTION THAT YOU WISH"
990 RETURN
1000 PRINT "YOUR VIEW FROM OAKRIDGE HILL
IS";PRINT "MAGNIFICENT, YOU CAN SEE";
1005 PRINT " GRENDALE THE HERMIT'S LOD
GE IN THE DISTANCE"
1010 RETURN

```

Note that the momento at the bottom of the well can come in extremely useful. (Part of the adventure involving the code on the momento is the hardest piece for the adventurer to solve.)

```

1020 PRINT "YOU HAVE FOUND YOURSELF IN A
PEACEFUL FIELD";PRINT
1030 PRINT "THERE IS A SMALL EMPTY WELL
NEXT TO YOU DO YOU WANT TO GO DOWN IT";
1032 PRINT " (Y/N)"
1035 INPUT Q$
1040 IF Q$="N" THEN PRINT "ALL RIGHT STA
Y IN THE FIELD THEN";RETURN
1050 IF INT(3*RND(1))=1 THEN 1120
1060 POKE 53280,0:POKE 53281,0:PRINT "2"
:REM CLEAR SCREEN
1070 FOR XX=1 TO 39
1080 PRINT TAB(XX) "E*";FOR TD=1 TO 100;
NEXT TD:REM WHITE
1090 NEXT XX
1095 PRINT "QQQQQ11111111111111111111SPLASH!!
";FOR TD=1 TO 1000:NEXT TD
1096 REM DOWN CRSR RIGHT CRSR

```

```

1097 POKE 53280,0:POKE 53281,11
1100 PRINT "2QQQQQHOW SILLY OF ME, I THO
UGHT IT WAS EMPTY"
1101 REM CLR SCR DOWN CRSR
1105 M(1)=10:M(2)=10:M(3)=1:B=1
1110 PRINT:PRINT "YOU DO MANAGE TO GET O
UT, BUT YOU LOSE MUCH MONEY AND STRENGT
H"
1115 GOTO 1150
1120 PRINT:PRINT "A SMALL NICHE IN THE W
ALL HIDES A GOLD"
1125 PRINT "FIGURE, THE RAM OF ANAG"
1130 PRINT "CARVED ON THE BASE IS THE WO
RD UGRANDOSE";REM LT RED WHITE
1135 PRINT "THE FIGURE IS VALUABLE LOOK
AFTER IT"
1140 P=1:M(3)=M(3)+1:M(2)=M(2)+5
1150 PRINT:PRINT "***** PRESS ANY KEY T
O CONTINUE *****"
1160 GET A$:IF A$="" THEN GOTO 1160
1165 GOSUB 730:RETURN
1170 PRINT "THE BARON'S HOUSE IS NEAR BY
YOU CAN HEAR MUSIC....."
1190 RETURN
1200 PRINT "YOU ARE IN A DAMP HUT WITH A
LEAKY ROOF A MUSTY SMELL LINGERS";
1202 PRINT " AND THERE IS ANOTHER R
OOM TO THE RIGHT"
1210 FOR TD=1 TO 1500:NEXT TD
1220 GOSUB 3150
1230 PRINT "WHAT'S THIS? A SMALL DOOR IN
FRONT OF YOU (N)"
1240 RETURN
1250 PRINT "AROUND YOU IS A SPARSELY FIL
LED STORE ROOM THIS IS A DEAD END"
1260 RETURN

```

```

1270 PRINT "YOU STAND IN THE MIDDLE OF B
ERRY HAMLET IT IS VERY QUIET THE ";
1275 PRINT "PLACE SEEMS          DESERTED"
1280 RETURN
1290 PRINT "YOU ARE ON A GRASSY HILL SLO
PING          DOWNWARDS INTO A VALLEY";
1295 PRINT "    A SMALL 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, then you will be allowed to enter. A much more sure way to enter the hut is to find the special treasure which will give you the code word that works every time. Every 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.

```

1310 PRINT " YOU ARE STANDING OUTSIDE A
WOODCUTTER'S HUT ENTER AT YOUR OWN ";
1315 PRINT "PERIL":GOSUB 3150
1320 PRINT:PRINT "TYPE IN RIGHT CODE TO
ENTER HUT":INPUT E$:PRINT
1330 F$="IVORYMAGICSILVEGRENDRUFUSDIAMOB
ERRYBARON"
1340 FOR T=1 TO INT(8*RND(1))+1:FF$=MID$
(F$,T*5-4,5):NEXT T
1345 IF LEFT$(E$,5)=FF$ OR (E$="DRAGON"
AND P=1) THEN 1360

```

```

1350 PRINT "YOU HAVE USED THE WRONG CODE
YOU HAVE BEEN CHASED AWAY BY";
1355 PRINT " THE WOODSMAN INTO  ANOTHER
PART OF THE FOREST"
1357 B=INT(3*RND(1))+2:M(1)=M(1)-8:RETUR
N
1360 PRINT "CORRECT CODE!!! ENTER HUT BY
HEADING 'N'":RETURN

```

The next location, location 10, is the prison that the Black Knight throws you into. Line 1420 sets the screen to black and then the rest of the routine gives you some textual details of your predicament and offers you one way of escaping. We will not tell you any more about how to escape; that is up to you to work out.

```

1370 PRINT N$="N":RETURN
1420 PRINT "␣":POKE 53280,0:POKE 53281,0
:GOSUB 3320:REM CLR SCRIN
1430 FOR T=1 TO 1000:NEXT T:POKE 53280,0
:POKE 53281,11
1440 PRINT "QQYOU FELL INTO A BEAR TRAP
AND WERE          CAUGHT BY THE BLACK";
1445 PRINT " KNIGHT"
1450 FOR T=1 TO 1000:NEXT T:PRINT "HE HA
S IMPRISONED YOU"
1460 INPUT "WHAT NOW":N$:IF N$<>"GRANDOS
" THEN 1480
1463 PRINT "CLOSE, BUT WRONG , YOU SHOUL
D HAVE          THOUGHT ABOUT THE";
1465 PRINT " FIGURE'S NAME":PRINT "THE R
AM OF ANAG...THE ANAG RAM"
1470 IF N$="DRAGONS" AND P=1 THEN GOSUB
SOUND:PRINT "␣":REM CLR SCRIN

```



```

1480 IF N$="DRAGONS" AND P=1 THEN PRINT
"QQQQQ11111THE MAGIC WORD!!"
1481 REM DOWN CRSR RIGHT CRSR
1482 IF N$<>"DRAGONS" THEN 1490
1485 IF P=1 THEN PRINT"QYOU ARE FREE AGA
IN, BUT YOU LOST THE GOLD FIGURE"
1487 REM DOWN CRSR
1488 IF N$="DRAGONS" AND P=1 THEN P=0:B=
1:RETURN
1490 PRINT:PRINT "YOU FAILED AND ARE IM
PRISONED FOREVER":END
1500 GOTO 1380
1510 RETURN
1610 PRINT "YOU ARE ON THE BARON'S LAND,
BE CAREFUL IN THE DISTANCE LIE THE ";
1615 PRINT "SCARED HILLS OF ANGLESARK"
1620 RETURN
1630 PRINT "THE DARK FOREST SURROUNDS YO
U ITS GROTESQUE FORMS SEEM ";
1635 PRINT "TO MOVE CLOSER":PRINT "THIS
PLACE IS EVIL"
1640 GOSUB 3150
1650 RETURN
1660 PRINT "YOU ARE BY THE SIDE OF A SMA
LL STREAM YOU CAN FOLLOW THE RIVER ";
1665 PRINT "BANK BOTH EASTAND WEST OR Y
OU CAN RETREAT TO THE NORTH"
1670 RETURN
1680 PRINT "A TWISTING TRACK CUTS THROUG
H THE TANGLED BUSHES THIS IS ";
1685 PRINT "WILD BOAR":PRINT "COUNTRY"
1690 RETURN
1700 PRINT "YOU ARE BESIDE A FAST FLOWIN
G RIVER A BRIDGE GUARDED BY A ";
1705 PRINT "GREEDY TROLL LIES TOTHE SOUT
H YOU MUST GUESS THE FEE THAT THE ";
1707 PRINT "BEAST CHARGES TO CROSS"

```

```

1710 INPUT F:IF F>INT(40*RND(1))+20 THEN
GOTO 1720
1715 PRINT "HE SCORNS YOUR TINY SUM AND
CHASES YOU AWAY":RETURN
1720 IF M(2)-F>0 THEN GOTO 1730
1725 PRINT "YOU GUESSED RIGHT BUT CANNOT
PAY THE TOLL BE ON YOUR WAY"
1727 RETURN
1730 PRINT:PRINT "THAT IS CLOSE ENOUGH,
CROSS NOW"
1740 M(2)=M(2)-F
1750 RETURN

```

The vital bridge location. If the adventurer wishes to complete the adventure, he must find some way across this bridge. If he offers the wrong amount, he is chased away by the troll that guards the bridge.

```

1850 PRINT "BEFORE YOU IS THE MYTHICAL C
ASTLE OF DIAMONDS DO NOT ";
1853 PRINT "BELIEVE YOUR EYES WHAT YOU
SEE BEFORE YOU IS AN ILLUSION CAST ";
1855 PRINT"BY THE MOST POWERFUL OF ALL":
PRINT"MAGICIANS..THE WARLOCK OF THE ";
1857 PRINT "DARK":PRINT "FOREST"
1860 GOSUB 3150
1870 RETURN
1880 PRINT "BEFORE YOU LIES THE MAGNIFIC
ENT WATCH TOWER OF PENDRAGON, A ";
1885 PRINT "SPLENDID GOTHIC MONUMENT"
1890 GOSUB 3150
1900 RETURN
1910 PRINT "YOU ARE AT THE ENTRANCE TO A
MYSTERIOUS CAVE THE SUN ALWAYS SETS";
1915 PRINT " TO THE RIGHT OF THE CAVE"

```

```

1920 RETURN
1930 PRINT "FROM YOUR POSITION IN A SMALL MARKET SQUARE YOU CAN SEE SOME ";
1935 PRINT "STOCKS AND A PILLORY";PRINT "A REMINDER OF WHAT WILL HAPPEN TO";
1937 PRINT " YOU IF YOU FAIL"
1940 RETURN

```

The Final Confrontation

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

Lines 1960 to 2020 announce the Creature's arrival.

Lines 2200 to 2240 give the adventure with an attribute less than the random figure R, a chance to escape. If the timer is less than 500 when he or she presses a key, then he or she is taken back to location 14, over the other side of the bridge.

```

1950 REM *** END OF GAME ***
1960 PRINT "IT IS VERY DARK INSIDE THE CAVE"
1970 FOR TD=1 TO 1500:NEXT TD:PRINT "YOU HEAR A LOUD NOISE"
1980 GOSUB 3320
1990 PRINT "ITS TH-TH-THE C-C-C-CREATURE

```

```

2000 FOR TD=1 TO 1000:NEXT TD
2020 R=INT(8*RND(1))+1
2200 PRINT "IF ANY OF YOUR ATTRIBUTES ARE LESS THAN";R;"THEN RUN AWAY!!"
2205 PRINT:PRINT "***** PRESS ANY KEY TO CONTINUE *****"
2210 FOR LL=1 TO 500:GET A$
2220 IF A$(<>)" THEN PRINT"PHEW! YOU MADE IT BACK OVER THE BRIDGE";B=14:GOTO 250
2230 NEXT
2240 IF M(1)<R OR M(2)<R OR M(3)<R THEN PRINT "s":GOTO 2910:REM CLR SCRIN

```

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 start 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 over to the greedy troll at the bridge. Therefore the adventurer, after several attempts, will realise that he has to move around the adventure to gain extra strength, wealth and magic so that he can be a reasonable opponent for the Creature.

```

2250 PRINT "LET THE FIGHT BEGIN....."
2255 FOR TD=1 TO 1000:NEXT TD
2260 PRINT "s":REM CLR SCRIN
2270 DIM K(3)
2280 K(1)=INT(50*RND(1))+11
2290 K(2)=INT(45*RND(1))+16
2300 K(3)=INT(12*RND(1))+1

```


The battle itself consists of a loop where all of the three attributes of the Creature and the adventurer are

One 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. This process continues until either the player's or the Creature's attribute 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 the counter reaches two, then the battle has ended and the program goes to the appropriate end game routine. There are two of these short routines, one for the player and one for the Creature.

When it comes down to it, one of these two endings must occur. An ending to a game is as important as any other part of the game. If the adventurer fails, you must give him or her the enthusiasm to have another try. If they win, well they have worked hard and deserve a little more than a 'Well Done' 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 the winning ending, that may be only only seen once every nine or ten games, that it starts to assume giant proportions and takes up

much valuable memory space.

We think our endings have struck a balance between the two extremes.

```
2910 REM *** END OF GAME ***
2920 FOR T=1 TO 24: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, MORTA
L"
3100 GOSUB 3150
3110 PRINT "    FOR YOU MAY STILL SUCCEED

3120 GOSUB 3150
3130 PRINT "    TO TRY AGAIN PRESS ANY KE
Y"
3140 GET A$:IF A$="" THEN GOTO 3140
3145 RUN
```

Our ending if you lose is quite simple but does tempt the player into having another go. The random notes are called up by the GOSUB 3150.

Our winning ending is a little more interesting. A text message is printed up on the screen at first, followed by the player being given an 'official' title.

```
3190 REM *** YOU WIN ***
3200 PRINT "§":GOSUB 3150:REM CLR SCR N
3210 PRINT "QQQ***** WELL DONE
*****":REM DOWN CRSR
3220 PRINT:PRINT "    YOU HAVE CONQUERED T
HE DARK FOREST":REM BLACK
3240 PRINT:PRINT "EYOU OUTWITTED YOUR EN
EMIES, EXPLORED THEFOREST AND FOUND ";
3241 REM WHITE
3245 PRINT "THE CREATURE"
3250 PRINT:PRINT "    YOU HAVE RID THE COUN
TRY OF A GREAT    THREAT AND YOU SHALL "
:
3255 PRINT "BE REWARDED"
3260 PRINT:INPUT "WHAT IS YOUR NAME":NA$
3270 PRINT "§":FOR T=1 TO 24:PRINT:NEXT:
PRINT "ARISE, SIR ":NA$
3275 REM CLR SCR N YELLOW
3280 FOR T=1 TO 20:PRINT:FOR TD=1 TO 200
:NEXT:NEXT
3290 FOR TL=15 TO 1 STEP -1:POKE UO,TL:P
OKE AD,18:POKE SR,132:POKE WL,17
3300 POKE HF,22:POKE LF,227:FOR DD=1 TO
500:NEXT:NEXT
3305 POKE UO,0:POKE WL,0
3310 GOTO 3310
```


We've now completed an analysis of an adventure. Phew! As we already have discussed, there is more

Finally, in this chapter, we have 'The Dark Forest' listing in full:

43

```

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$<>"R" THEN 230
200 CS=INT((B+10)/10):PRINT "110.K. YOU CAN SLEEP HERE BUT IT WILL"
201 REM RIGHT CRSR
205 PRINT "COST YOU";CS;"COIN/S FOR EVERY HOURS UNINTERRUPTED REST":PRINT
210 INPUT "HOW MANY HOURS";H
215 IF H<1 OR M(2)-(H*CS)<0 THEN PRINT "TRY AGAIN SLEEPY!!":GOTO 210
220 M(2)=M(2)-(H*CS):M(1)=M(1)+H:PRINT "sQQQQQQJJJJJJJJJJJJZZZZZZZZZZ"
221 REM CLR SCRN DOWN CRSR RIGHT CRSR
222 FOR DD=1 TO 500*H:NEXT DD:PRINT "s":GOSUB 730:GOTO 250:REM CLR SCRN
230 IF CHK=B THEN PRINT "CANNOT MOVE IN THAT DIRECTION":GOSUB 3150:GOTO 260
240 CHK=B
250 IF B>10 THEN GOTO 254
252 ON B GOSUB 980,1000,1020,1170,1200,1250,1270,1290,1310,1420:GOTO 270
254 B=B-10:ON B GOSUB 1610,1630,1660,1680,1700,1850,1880,1910,1930,1950:B=B+10
260 IF Z(B,5)<>0 THEN 300
270 GOSUB 790
280 PRINT:INPUT "WHAT NOW";N$: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 ROBBERS SEIZE YOU":E=40
330 IF MN=2 THEN GOSUB 3410

```

```

340 IF MN=3 THEN PRINT "SIR RUFUS OF GAS CONY ATTACKS...":E=20
350 IF MN=4 THEN PRINT "A FEROCIOUS BEAR POUNCES":E=20
360 IF MN=5 THEN PRINT "YOU HEAR THE WAR CRY OF THE BLACK KNIGHT":E=60
370 INPUT "WILL YOU FIGHT OR RUN";F$:F$=LEFT$(F$,1)
380 PRINT:IF F$="F" THEN 410
390 IF RND(1)>0.4 THEN INPUT "WHICH DIRECTION, COWARD";N$:M(1)=M(1)-4:GOTO 130
400 PRINT "NO! YOU MUST STAND AND FIGHT"
405 PRINT "YOU LOSE 10 STRENGTH":M(1)=M(1)-10
410 REM *** FIGHT ***
420 Z(B,5)=0:RO=INT(19*RND(1))+1
430 IF Z(RO,5)=0 THEN Z(RO,5)=MN:GOTO 440
435 IF Z(RO,5)<>0 THEN GOTO 420
440 PRINT "WHICH ATTRIBUTE DO YOU WISH TO FIGHT WITH (ENTER WORD)"
445 INPUT 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 2910
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 PRINT "s":IF M(A)>60 THEN DD=M(A)-60:M(A)=60
515 REM CLEAR SCREEN
530 PRINT "sQJJJJJJJJJJRzTHE BATTLE COMMENCES.."

```



```

535 REM CLR SCRN DOWN CRSR RIGHT CRSR RE
V ON LT BLUE
540 PRINT "QQ YOU: ";
541 REM DOWN CRSR PURPLE YELLOW
545 FOR TX=1 TO M(A):PRINT "q";:NEXT
546 PRINT
548 PRINT "QQ ENEMY: ";
550 FOR XT=1 TO E:PRINT "q";:NEXT
560 GOSUB 3390
570 REM *** MAIN FIGHT LOOP ***
580 IF INT(7*RND(1))=4 THEN GOTO 600
590 E=E-1
600 GOSUB 3390
610 IF INT(3*RND(1))=1 THEN GOTO 630
620 M(A)=M(A)-1
630 GOSUB 3390
640 IF E>0 AND M(A)>0 THEN GOTO 530
650 IF M(A)<1 THEN GOTO 2910
660 PRINT "QQ YOU WIN!!QE":REM DOWN CRSR
WHITE
670 GOSUB 3390
680 ER=INT(40*RND(1))+1:BV=INT(7*RND(1))
+1
690 IF BV<4 THEN PRINT "YOU GAIN":ER;"ST
RENGTH":M(1)=M(1)+ER+DD
700 IF BV<7 AND BV>3 THEN PRINT "YOU GAI
N":ER;"MONEY UNITS":M(2)=M(2)+ER+DD
710 IF BV=7 THEN ER=INT(ER/10)+1:PRINT "
YOU GAIN":ER;"MAGIC SPELLS"
715 IF BV=7 THEN M(3)=M(3)+ER+DD
720 FOR T=1 TO 1500:NEXT T:PRINT "g":GOT
O 250:REM CLEAR SCREEN
730 REM *** ATTRIBUTES ***
740 M(1)=M(1)-1:IF M(1)<1 THEN 2910
750 PRINT "STRENGTH: ";INT(M(1))
760 PRINT "MONEY: ";M(2)
770 PRINT "MAGIC: ";M(3)

```

```

780 RETURN
790 IF RND(1)>0.8 AND B<>1 AND B<>15 AND
B<>20 THEN PS=INT(5*RND(1))+1:GOTO 810
800 RETURN
810 IF PS=1 THEN PRINT " AN OLD GLASS BO
TTL IS WITHIN REACH. IT IS COVERED";
815 IF PS=1 THEN PRINT " IN COBWEBS"
820 IF PS=2 THEN PRINT " HERE IS A LARGE
OAK CHEST. A WIERD":PRINT "SIGN IS";
825 IF PS=2 THEN PRINT " CARVED ON THE L
ID"
830 IF PS=3 THEN PRINT " A VELLUM BOOK L
IES BY YOUR FEET"
840 IF PS=4 THEN PRINT " A GOLD LOCKET O
N A CHAIN IS CLOSE TO":PRINT "YOU"
850 IF PS=5 THEN PRINT " BEFORE YOU IS A
BOX OF EXOTIC FRUIT"
860 REM *** POSSESSIONS ***
870 INPUT "DO YOU WANT TO OPEN IT":OP$
880 PRINT:IF LEFT$(OP$,1)<>"Y" THEN RETU
RN
885 IF RND(1)<0.7 THEN GOTO 900
890 PRINT "AN EVIL ORC LEAPS OUT AND ATT
ACKS YOU"
894 FOR TT=1 TO 1000:NEXT TT:PRINT "AFTE
R A BRIEF STRUGGLE IT RUNS AWAY"
896 PRINT "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 THE":PRINT "ESSENCE";
905 IF PS=1 THEN PRINT " GIVES YOU NEW S
TRENGTH AND LIFE":M(1)=M(1)+8
910 IF PS=2 THEN PRINT "MUCH TREASURE IS
CONTAINED IN THE CHEST"
915 IF PS=2 THEN PRINT "YOU ARE A RICH M
AN":M(2)=M(2)+35+INT(15*RND(1))
920 IF PS=3 THEN PRINT "IT CONTAINS 6 PO

```



```

1230 PRINT "WHAT'S THIS? A SMALL DOOR IN
FRONT OF YOU (N)"
1240 RETURN
1250 PRINT "AROUND YOU IS A SPARSLEY FIL
LED STORE ROOM THIS IS A DEAD END"
1260 RETURN
1270 PRINT "YOU STAND IN THE MIDDLE OF B
ERRY HAMLET IT IS VERY QUIET THE ";
1275 PRINT "PLACE SEEMS DESERTED"
1280 RETURN
1290 PRINT "YOU ARE ON A GRASSY HILL SLO
PING DOWNWARDS INTO A VALLEY";
1295 PRINT " A SMALL HAMLET LIES AT THE
NORTHERN END"
1300 RETURN
1310 PRINT " YOU ARE STANDING OUTSIDE A
WOODCUTTER'S HUT ENTER AT YOUR OWN ";
1315 PRINT "PERIL":GOSUB 3150
1320 PRINT:PRINT "TYPE IN RIGHT CODE TO
ENTER HUT":INPUT E$:PRINT
1330 F$="IVORYMAGICSILVEGRENDRUFUSDIAMOB
ERRYBARON"
1340 FOR T=1 TO INT(8*RND(1))+1:FF$=MID$(
F$,T*5-4,5):NEXT T
1345 IF LEFT$(E$,5)=FF$ OR (E$="DRAGON"
AND P=1) THEN 1360
1350 PRINT "YOU HAVE USED THE WRONG CODE
YOU HAVE BEEN CHASED AWAY BY";
1355 PRINT " THE WOODSMAN INTO ANOTHER
PART OF THE FOREST"
1357 B=INT(3*RND(1))+2:M(1)=M(1)-8:RETUR
N
1360 PRINT "CORRECT CODE!!! ENTER HUT BY
HEADING 'N':RETURN
1370 PRINT N$="N":RETURN
1420 PRINT "␣":POKE 53280,0:POKE 53281,0
:GOSUB 3320:REM CLR SCRIN

```

```

1430 FOR T=1 TO 1000:NEXT T:POKE 53280,0
:POKE 53281,11
1440 PRINT "QQYOU FELL INTO A BEAR TRAP
AND WERE CAUGHT BY THE BLACK";
1445 PRINT " KNIGHT"
1450 FOR T=1 TO 1000:NEXT T:PRINT "HE HA
S IMPRISONED YOU"
1460 INPUT "WHAT NOW":N$:IF N$(<)"GRANDOS
THEN 1480
1463 PRINT "CLOSE, BUT WRONG , YOU SHOUL
D HAVE THOUGHT ABOUT THE";
1465 PRINT " FIGURE'S NAME":PRINT "THE R
AM OF ANAG...THE ANAG RAM"
1470 IF N$="DRAGONS" AND P=1 THEN GOSUB
SOUND:PRINT "␣":REM CLR SCRIN
1480 IF N$="DRAGONS" AND P=1 THEN PRINT
"QQQQQ11111THE MAGIC WORD!!"
1481 REM DOWN CRSR RIGHT CRSR
1482 IF N$(<)"DRAGONS" THEN 1490
1485 IF P=1 THEN PRINT "QYOU ARE FREE AGA
IN, BUT YOU LOST THE GOLD FIGURE"
1487 REM DOWN CRSR
1488 IF N$="DRAGONS" AND P=1 THEN P=0:B=
1:RETURN
1490 PRINT:PRINT "YOU FAILED AND ARE IM
PRISONED FOREVER":END
1500 GOTO 1380
1510 RETURN
1610 PRINT "YOU ARE ON THE BARON'S LAND,
BE CAREFUL IN THE DISTANCE LIE THE ";
1615 PRINT "SCARED HILLS OF ANGLSARK"
1620 RETURN
1630 PRINT "THE DARK FOREST SURROUNDS YO
U ITS GROTESQUE FORMS SEEM ";
1635 PRINT "TO MOVE CLOSER":PRINT "THIS
PLACE IS EVIL"
1640 GOSUB 3150

```

```

1650 RETURN
1660 PRINT "YOU ARE BY THE SIDE OF A SMALL STREAM YOU CAN FOLLOW THE RIVER ";
1665 PRINT "BANK BOTH EASTAND WEST OR YOU CAN RETREAT TO THE NORTH"
1670 RETURN
1680 PRINT "A TWISTING TRACK CUTS THROUGH THE TANGLED BUSHES THIS IS ";
1685 PRINT "WILD BOAR";PRINT "COUNTRY"
1690 RETURN
1700 PRINT "YOU ARE BESIDE A FAST FLOWING RIVER A BRIDGE GUARDED BY A ";
1705 PRINT "GREEDY TROLL LIES TO THE SOUTH YOU MUST GUESS THE FEE THAT THE ";
1707 PRINT "BEAST CHARGES TO CROSS"
1710 INPUT F:IF F>INT(40*RND(1))+20 THEN GOTO 1720
1715 PRINT "HE SCORNS YOUR TINY SUM AND CHASES YOU AWAY";RETURN
1720 IF M(2)-F>0 THEN GOTO 1730
1725 PRINT "YOU GUESSED RIGHT BUT CANNOT PAY THE TOLL BE ON YOUR WAY"
1727 RETURN
1730 PRINT:PRINT "THAT IS CLOSE ENOUGH, CROSS NOW"
1740 M(2)=M(2)-F
1750 RETURN
1850 PRINT "BEFORE YOU IS THE MYTHICAL CASTLE OF DIAMONDS DO NOT ";
1853 PRINT "BELIEVE YOUR EYES WHAT YOU SEE BEFORE YOU IS AN ILLUSION CAST ";
1855 PRINT "BY THE MOST POWERFUL OF ALL THE PRINT MAGICIANS, THE WARLOCK OF THE ";
1857 PRINT "DARK";PRINT "FOREST"
1860 GOSUB 3150
1870 RETURN

```

```

1880 PRINT "BEFORE YOU LIES THE MAGNIFICENT WATCH TOWER OF PENDRAGON, A ";
1885 PRINT "SPLENDID GOTHIC MONUMENT"
1890 GOSUB 3150
1900 RETURN
1910 PRINT "YOU ARE AT THE ENTRANCE TO A MYSTERIOUS CAVE THE SUN ALWAYS SETS";
1915 PRINT "TO THE RIGHT OF THE CAVE"
1920 RETURN
1930 PRINT "FROM YOUR POSITION IN A SMALL MARKET SQUARE YOU CAN SEE SOME ";
1935 PRINT "STOCKS AND A PILLORY";PRINT "A REMINDER OF WHAT WILL HAPPEN TO";
1937 PRINT "YOU IF YOU FAIL"
1940 RETURN
1950 REM *** END OF GAME ***
1960 PRINT "IT IS VERY DARK INSIDE THE CAVE"
1970 FOR TD=1 TO 1500:NEXT TD:PRINT "YOU HEAR A LOUD NOISE"
1980 GOSUB 3320
1990 PRINT "ITS TH-TH-THE C-C-C-CREATURE"

2000 FOR TD=1 TO 1000:NEXT TD
2020 R=INT(8*RND(1))+1
2200 PRINT "IF ANY OF YOUR ATTRIBUTES ARE LESS THAN";R;"THEN RUN AWAY!!"
2205 PRINT:PRINT "***** PRESS ANY KEY TO CONTINUE *****"
2210 FOR LL=1 TO 500:GET A$
2220 IF A$<>" " THEN PRINT "PHEW! YOU MADE IT BACK OVER THE BRIDGE";B=14:GOTO 250
2230 NEXT
2240 IF M(1)<R OR M(2)<R OR M(3)<R THEN PRINT "L";GOTO 2910:REM CLR SCRN
2250 PRINT "LET THE FIGHT BEGIN....."
2255 FOR TD=1 TO 1000:NEXT TD

```



```

2260 PRINT "s":REM CLR SCRIN
2270 DIM K(3)
2280 K(1)=INT(50*RND(1))+11
2290 K(2)=INT(45*RND(1))+16
2300 K(3)=INT(12*RND(1))+1
2310 FOR Y=1 TO 3
2320 FOR T=1 TO 1000:NEXT T
2330 PRINT "Ss";
2331 REM HOME CRSR CLR SCRIN
2335 PRINT "QQTHE BATTLE STARTS...":PRIN
T "QQROUND: ";Y
2336 REM DOWN CRSR
2340 PRINT "QQQ111HUMAN: ";M(Y)
2341 REM DOWN CRSR RIGHT CRSR
2350 PRINT "QQQ111CREATURE: ";K(Y)
2360 IF INT(10*RND(1))=3 THEN 2380
2370 M(Y)=M(Y)-1
2380 IF INT(10*RND(1))=7 THEN 2400
2390 K(Y)=K(Y)-1
2400 IF M(Y)<1 THEN 2560
2410 IF K(Y)<1 THEN 2520
2420 GOSUB 3150
2510 GOTO 2330
2520 FOR T=1 TO 100:NEXT T
2530 PRINT "s":PRINT "QQQQQQQQ1111111111Y
OU WIN!!!!":FOR TD=1 TO 2000:NEXT
2535 REM CLR SCRIN DOWN CRSR RIGHT CRSR
2540 GOSUB 3320
2550 V=V+1:IF V=2 THEN 3190
2555 IF V<>2 THEN NEXT Y
2560 FOR T=1 TO 1000:NEXT T
2570 PRINT "s":PRINT "QQQQQQQQ1111111111T
HE CREATURE WINS!!"
2575 REM CLR SCRIN DOWN CRSR RIGHT CRSR
2580 GOSUB 3320
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)=INT(30*RND(1))+21
2650 M(2)=INT(70*RND(1))+11
2660 M(3)=INT(6*RND(1))+1
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=INT(18*RND(1))+2
2880 IF Z(RO,5)=0 THEN Z(RO,5)=A:GOTO 28
90
2885 GOTO 2870
2890 NEXT A
2892 VO=54296:AD=54277:SR=54278:WL=54276
:HF=54273:LF=54272
2900 RETURN
2910 REM *** END OF GAME ***
2920 FOR T=1 TO 24: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, MORTA
L"
3100 GOSUB 3150
3110 PRINT "    FOR YOU MAY STILL SUCCEED

3120 GOSUB 3150
3130 PRINT "    TO TRY AGAIN PRESS ANY KE
Y"
3140 GET A$: IF A$="" THEN GOTO 3140
3145 RUN
3150 POKE VO,15:POKE AD,18:POKE SR,132:P
OKE WL,33
3155 HX=INT(40*RND(1))+3:LX=INT(100*RND(
1))+50
3160 POKE HF,HX:POKE LF,LX:FOR TD=1 TO 7
5:NEXT TD:POKE HF,0:POKE LF,0:RP=RP+1
3165 IF RP>3 THEN RP=0:POKE WL,0:POKE AD
,0:POKE SR,0:RETURN
3170 GOTO 3155
3180 FOR TD=1 TO 500:NEXT TD:RETURN
3190 REM *** YOU WIN ***
3200 PRINT "␣":GOSUB 3150:REM CLR SCRN
3210 PRINT "QQQXXXXXXXXXXXXXXXXX WELL DONE
XXXXXXXXXXXXXXXXX":REM DOWN CRSR
3220 PRINT:PRINT "    YOU HAVE CONQUERED T
HE DARK FOREST":REM BLACK

```

```

3240 PRINT:PRINT "YOU OUTWITTED YOUR EN
EMIES, EXPLORED THEFOREST AND FOUND ";
3241 REM WHITE
3245 PRINT "THE CREATURE"
3250 PRINT:PRINT "    YOU HAVE RID THE COUN
TRY OF A GREAT    THREAT AND YOU SHALL
:
3255 PRINT "BE REWARDED"
3260 PRINT:INPUT "WHAT IS YOUR NAME":NA$
3270 PRINT "␣":FOR T=1 TO 24:PRINT:NEXT:
PRINT "ARISE, SIR ":NA$
3275 REM CLR SCRN YELLOW
3280 FOR T=1 TO 20:PRINT:FOR TD=1 TO 200
:NEXT:NEXT
3290 FOR TL=15 TO 1 STEP -1:POKE VO,TL:P
OKE AD,18:POKE SR,132:POKE WL,17
3300 POKE HF,22:POKE LF,227:FOR DD=1 TO
500:NEXT:NEXT
3305 POKE VO,0:POKE WL,0
3310 GOTO 3310
3320 POKE VO,15:POKE AD,18:POKE SR,132:P
OKE WL,33
3325 HX=INT(4*RND(1))+2:LX=INT(150*RND(1
))+75
3330 POKE HF,HX:POKE LF,LX
3340 FOR TD=1 TO 550:NEXT
3345 POKE HF,0:POKE LF,0
3350 RP=RP+1: IF RP>6 THEN RP=0:POKE VO,0
:POKE WL,0:RETURN
3360 GOTO 3325
3370 POKE VO,15:POKE AD,9:POKE WL,17
3380 POKE HF,17:POKE LF,37:FOR TD=1 TO 5
00:NEXT:POKE VO,0:POKE WL,0:RETURN
3390 POKE VO,15:POKE AD,17:POKE SR,24:PO
KE WL,129
3400 POKE HF,25:POKE LF,177:FOR TD=1 TO
75:NEXT TD:POKE VO,0:POKE WL,0:RETURN

```



```

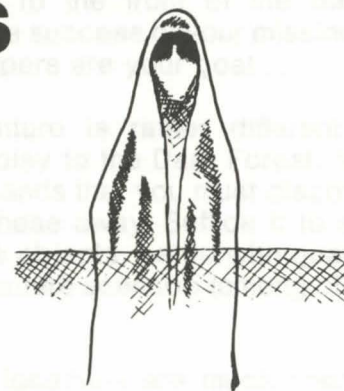
3410 PRINT:PRINT "YOU HAVE COME ACROSS G
RENDAL THE HERMIT"
3420 PRINT "YOU HAVE AWOKEN HIM  HE IS
NOT PLEASED":E=40+INT(50*RND(1))
3430 RETURN

```

CHAPTER THREE

Chapter Three

A Full-Scale Text Adventure, The Nielson Papers



Chapter Three

A Full-Scale Text Adventure, The Nielson Papers



CHAPTER THREE

Moving on from our first adventure, we now have 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 is now making vast sums of money from this invention and rival companies just cannot wait to get their hands on the formula and manufacturing techniques described in these papers. You are employed by the fuel-giant Trithon Ltd to steal the papers and escape safely from the office block.

To complete your task, you will need to find various pieces of equipment, keep yourself out of many traps and solve the complex problems placed around the adventure. Of course, the building is patrolled by security guards who 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. Suffice it to say they allow you to acquire objects, battle with guards and move around the adventure scenario (among other things).

The descriptions of the locations are much shorter than in the previous adventure. Much of the adventure

In many situations, the computer gives more help than a simple "I CAN'T DO THAT", which can be so frustrating in adventures. If you find you are getting nowhere in the adventure, then have a look in the 'HELP!' chapter.

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

Line 25 sets the screen and border colours and then the program displays the title page.

62

Once the title page is displayed, the program then initialises, with the title page still displayed. All the arrays are dimensioned in line 1450. 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 in the game. I\$ is filled with the descriptions of the 30 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. Array P contains the room logic.

The listing for the initialisation section follows. After that, we will take just a couple of sections from the rest of the program, and examine them in detail.

```

1440 REM *** INITIALISATION ***
1450 DIM O$(18),L$(30),P(30,4),L(18),A(18)
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
1495 DATA STOOL,WIRECUTTERS
1500 DATA MONEY,COIN-MACHINE,SILENCER,LA
DDER,KEY,PAPER-CLIPS,ROPE,SECRET-PAPERS
1505 DATA PAPERS
1510 DATA THE MAIN ENTRANCE,THE RECEPTIO
N AREA,THE DOCUMENT ROOM
1515 DATA THE STAFF CANTEEN,THE DELIVERY
ROOM
1520 DATA A CORNER OF THE YARD NEXT TO A
N ELECTRIFIED FENCE
1524 DATA "THE ALLEY BEHIND THE COMPLEX,
A CAR IS WAITING TO TAKE YOU AWAY"
1530 DATA THE COMPUTER ROOM,THE MAINTENAN
CE AREA,A DARK CORRIDOR
1535 DATA THE RECORDS DEPT,AN UNTIDY OFF
ICE
1540 DATA THE TYPING POOL,THE RECREATION
ROOM,A NARROW CORRIDOR
1545 DATA A PLUSH CONFERENCE ROOM,AN UNL
IGHTED OFFICE
1550 DATA THE STRONGROOM NEXT TO THE U
AULT,THE STEEL VAULT,THE ACCOUNTS DEPT
1555 DATA THE EXECUTIVE WASHROOMS
1560 DATA A COLD BARE ROOM,A DARK CORRID
OR,A CLEAN NEW STOREROOM
1565 DATA A LARGE CUPBOARD,A BRIGHT HALL
WAY
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 MOUTAKGETSTEDROREACLICUTCHAO
PEUNLFIRSHOWAIDRISEALOOWEACOMCALDEC"
1605 C$=C$+"HELCLUQUIPUTHITFIGATTKIL"
1610 O$="GUNTORCOFMEMCOMGLOCUPSTOWIRMONC
OISILLADKEYCL IROPSECPAP"
1620 P$="PARDON? WHAT? RUBBISH!REPHRAS
E"
1630 L=1:L(1)=0:K=0:SA=10000:KY=0:GS=0
1640 G1=INT(28*RND(1))+3:IF G1=28 OR G=7
THEN 1640
1650 DC=INT(6*RND(1))+5
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
1675 DATA 21,20,8,13,9,4,0,9,1,0,27,14,1
2,17,0,11,16,16,0,0,9,26,0,18,11,5,8
1680 DATA 0,21,-1,23,12,26,22,12,15,0,19
,20,18,14,0,-1,0,0,18,9,0,0,2,17,8,-1
1685 DATA 0,17,-1,28,16,23,0,17,8,0,26,1
2,0,0,0,27,6,14,0,24,11,-1,25,13,29,16
1690 DATA 0,0,28,0,0,30,0,0,29,0,80,10,0
,1,200,8,3,10,22,0,0,10,40,10,1,27
1695 DATA 1000000,2
1700 VO=54296:WF=54276:AD=54277:SR=54278
:HF=54273:LF=54272
1720 RETURN

```

The 'what can I see' and 'What am I carrying' routine:

```

110 S=0:PRINT:PRINT "YOU HAVE:";
115 REM LT GREEN

```



```

120 FOR D=1 TO 18: IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" ":GOTO 130
125 A$=" 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:";
175 REM PURPLE
180 FOR D=1 TO 18: IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" ":GOTO 190
185 A$=" 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 YOUR MISSIONAND THE ORGANIZATION";
1225 PRINT "HAS KINDLY LET YOU KEEP THE
OBJECTS YOU 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 "EXCERT THE SECRET PAPERS"
1270 FOR T=1 TO 5000:NEXT T
1280 GOSUB 2180:PRINT "00111111111111WEL
L DONE!!!":END

```

If you manage to escape but forget something ...

```

1290 PRINT:PRINT "YOU'RE SAFELY AWAY BUT
YOU DID NOT STEALTHE PAPERS ";
1292 PRINT "THE ORGANIZATION IS NOT
PLEASED... YOU HAVE BEEN SENT BACK ";
1294 PRINT "TO TRY AGAIN":FOR T=1 TO 30
00:NEXT T:RUN

```

If you fail miserably ...

```

1300 FOR T=1 TO 3000:NEXT T:FOR T=1 TO 6
:PRINT:NEXT:PRINT "s"
1305 POKE 53280,11:POKE 53281,0
1310 GOSUB 2140:PRINT "└" ***** **
*** ***** * "
1315 REM YELLOW
1320 PRINT " * * * *
*"
1330 PRINT " **** *
*"
1340 PRINT " * * * *
*"
1350 PRINT " * * * *****
*****"
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 a value 0. 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 0. 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 which display what is in the same location and also explain what you are carrying. These routines are situated at lines 110 to 220. It seems now the best time to look closely at these routines which are displayed following the initialisation section.

Briefly, with these routines the computer checks through the object location array and any object with the value 0 is displayed in the player's inventory. Any object with the value equal to the location number the player is at (held in the variable, L), is printed as being seen by the player. Lines 120 and 180 are there simply for grammatical reasons, and 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 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 concerns the acceptance and processing of the player's commands.

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.

```
240 PRINT:PRINT "WHAT NOW"
242 REM CYAN
245 INPUT N$
250 GOSUB 1370
```

The routine at 1370 is very useful and basically dissects the parts of the command phrase or sentence that are required.

```
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$,CT+1,3)
1425 IF MID$(N$,CT,1)<>" " THEN NEXT CT
1430 RETURN
```

The routine does the following: It checks to see that the command entered is less than three letters, if this is the case it goes back to the main loop, the command entered was an invalid one. Lines 1390, 1400 and 1410 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 primarily used when there is an object manipulation.

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 ... GOSUB's to the line starting the subroutine to operate that particular command. The variable T is used here.

```
270 IFT>960RT<1THENPRINT:PRINTTAB(12)MID
$(P$,(INT(4*RND(1))*8)+1,8):GOTO290
278 IF T>14 THEN GOTO 282
280 ON T GOSUB 340,340,420,420,420,510,5
70,630,650,680,700,700,830,830:GOTO 290
282 IF T>23 THEN GOTO 285
283 T=T-14:ON T GOSUB 920,930,960,960,10
20,1040,1040,1040,1060:T=T+14:GOTO 290
285 T=T-23:ON T GOSUB 1070,1070,1100,113
0,1130,1130:T=T+23
```

We will not discuss all the commands available in this adventure. We have, instead, 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, north, south, east or west. The two words must be separated by a space. The movement mechanism is based on the logic used for linking locations in this adventure. Each location, as you'll see if you look at the location data in the initialisation section, has four pieces of data. Each

refers to the new location in which the player would end up 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, 1 if it is north that you decided to move to and so on. If you are moving in an unblocked direction, the program makes the new location variable, L, equal to the figure in the room array that applies to your present position and the direction in which you wish. For example, if you plan to move N 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 in the data in line 1670, the first four numbers refer to location 1. Moving north is handled by the first number, which is 2. Therefore the new location that you are moved to is location number 2.

If the computer finds the new location is less than 1, then the player cannot go that way. The program then 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
```


We'll now look at one of the 'red herring' commands, HELP. If this is entered, the computer responds with a curt phrase, "HELP YOURSELF, NO CLUES FROM ME". The program then returns. A few of these type of commands, eliciting a humorous reply from the computer, take up little memory and add a little light-heartedness to the whole affair.

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

The READ command is an interesting one to look at and typifies many of the commands in this adventure. Many of the commands fall roughly into the following format: Is the object required to do this currently being carried by the player; Is he or she in the right location to do this; Does the player request the right object to use in conjunction with the command; And are all other conditions (such as extra essential objects) correct for the operation to take place. If there is anything wrong, some form of action, usually a message is displayed, and the game goes back to the main loop. If everything is okay, the program performs the operation, points out that it has performed it, and 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 (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 secret papers) is with the player in that location, then the computer prints a 'nothing to read' message. If the memo is with the

player (L(4) the place in the object array for the memo is equal to the adventurer's present location), a coded message is printed and you are told that it is written in code and needs the computer to decipher it, another object for the papers, he or she is told that they are a 'red herring', just company accounts. The player is reprimanded if he or she tries to read the secret papers. After these actions have been followed, the computer waits for you to press a key to continue the game. The READ routine is as follows:

```
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: TELVSRI TEAHOMHGOEAEHWSRO":PRINT
592 IF L(4)=L THEN PRINT "IT IS WRITTEN IN CODE, YOU NEED THE"
594 IF L(4)=L THEN PRINT "COMPUTER TO DECODE IT"
600 IF L(18)=L THEN PRINT "DON'T WASTE YOUR TIME, THEY ARE COMPANY"
604 IF L(18)=L THEN PRINT "ACCOUNTS, VERY BORING"
610 IF L(17)=L THEN PRINT "YOUR JOB IS TO STEAL THEM, NOT READ THEM"
620 PRINT:PRINT "PRESS RETURN TO CONTINUE"
625 INPUT L$:RETURN
```

We will have a look at one part of another command, UNLOCK. This command, obviously enough, unlocks a

locked door, if you have the key. 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 overuse. It checks that the last word in the sentence was door, so that the command phrases 'UNLOCK THE SOUTHWARDS 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 a 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 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 and generally perform a wide number of actions involving certain conditions that must be met to infuriate and amuse the player with simple 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 that covers many possible conditions.

What we've talked about on commands in this adventure could be said to be just the tip of the iceberg. Though additional commands 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 less than 16K. Because we have kept the program all text, converting it to other computers will not be difficult. There is so much memory space left that this game can be expanded to more than double its present size.

Several final points. Firstly when playing the game, after the computer responds to your command, there is a long pause. If you find the pause too long, 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:GET A$:IF A$="" THE  
N NEXT:GOTO 320
```

Secondly, you may have noticed that none of the adventures in this book feature a game 'save' or 'resurrection' feature. A game save feature simply lets you save your current status onto cassette or disk so that when you return to play the game some time later, you do not have to play the early stages of the game again. A resurrection feature allows a player killed by 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, as they are finely

balanced between the computer and the player. These extra features would make the game much easier to solve. Dark Forest, for example, with its random elements would suffer badly with the adventurer only going through the early stages once.

These two features are usually found on large commercial adventures and if you wish, you can add them to your program. The game save feature can be added by using the cassette or disk file statements and recording 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 to implement. All you need to do is to add a small routine in front of the lose routine at the end of the game. This sort of extra routine would take away a certain amount of such things as treasure and points, let you carry on the game and perhaps move you back to a location closer to the start of the game.

Finally, here is the complete listing:

```

10 REM *** THE NIELSON PAPERS ***
20 REM *** A TEXT ADVENTURE ***
25 PRINT "s":POKE 53280,6:POKE 53281,8
30 GOSUB 1730:GOSUB 1440
40 REM *** MAIN LOOP ***
50 SA=SA-1:IF SA=0 THEN 1300
55 POKE 53281,2:POKE 53280,11
60 PRINT "s":PRINT:PRINT "YOU ARE IN ";
L$(L)
65 REM CLR SCREEN YELLOW
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 TO
P FLOOR"
100 FORM=1TO4:IF P(L,M)=-1 THEN PRINT"LO
CKED DOOR TO THE ";MID$(D$,(5*M)-4,5)
110 S=0:PRINT:PRINT "YOU HAVE:";
115 REM LT GREEN
120 FOR D=1 TO 18:IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" ":GOTO 130
125 A$=" 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:";
175 REM PURPLE
180 FOR D=1 TO 18:IF D=6 OR D=9 OR D=10
OR D>14 THEN A$=" ":GOTO 190
185 A$=" 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 "A SECURIT
Y GUARDE":DC=DC-1:IF DC=0 THEN 1300
235 REM BLACK WHITE
240 PRINT:PRINT "WHAT NOW"
242 REM CYAN
245 INPUT 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
$(P$,(INT(4*RND(1)))*8)+1,8):GOTO 290
278 IF T>14 THEN GOTO 282

```



```

280 ON T GOSUB 340,340,420,420,420,510,5
70,630,650,680,700,700,830,830:GOTO 290
282 IF T>23 THEN GOTO 285
283 T=T-14:ON T GOSUB 920,930,960,960,10
20,1040,1040,1040,1060:T=T+14:GOTO 290
285 T=T-23:ON T GOSUB 1070,1070,1100,113
0,1130,1130:T=T+23
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:GET A$:IF A$="" THE
N NEXT:GOTO 320
315 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: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$(C
$,CT,3)=LEFT$(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 SOM
ETHING FIRST":RETURN
450 IF (CT=8 OR CT=13) AND S>2 THEN GOSU
B 1800:RETURN
470 IF CT=3 OR CT=11 THEN PRINT "DON'T B
E SILLY IT IS TOO HEAVY":RETURN
480 IF L(17)=3 THEN SA=25:GOSUB 2140
482 IF L(17)=3 THEN PRINT "THE ALARMS HA
VE SOUNDED, YOU DON'T HAVE ";

```

```

484 IF L(17)=3 THEN PRINT "MUCH TIME":L(
17)=0:RETURN
490 IF L(CT)=L THEN L(CT)=0:PRINT "YOU A
RE CARRYING THE ";0$(CT):RETURN
500 IF L=3 AND CT=17 THEN PRINT "THEY AR
E OUT OF REACH":RETURN
505 IF L(CT)<>L THEN PRINT "IT IS NOT HE
RE":RETURN
510 REM *** DROP ***
520 DV=0:FOR CT=1 TO 54 STEP 3:IF LEFT$(
CT$,3)=MID$(C$,CT,3) THEN DV=CT
530 NEXT:CT=(DV+2)/3:IF DV=0 THEN T=100:
RETURN
540 IF S=0 THEN PRINT "YOU HAVE NOTHING
TO DROP":RETURN
550 IF L(CT)<>0 THEN PRINT "YOU DON'T HA
VE IT TO DROP":RETURN
560 L(CT)=L:PRINT "YOU HAVE DROPPED THE
";0$(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: TELVSRI
TEAHOMHG0EAENHWSRO":PRINT
592 IF L(4)=L THEN PRINT "IT IS WRITTEN
IN CODE, YOU NEED THE"
594 IF L(4)=L THEN PRINT "COMPUTER TO DE
CODE IT"
600 IF L(18)=L THEN PRINT "DON'T WASTE Y
OUR TIME, THEY ARE COMPANY"
604 IF L(18)=L THEN PRINT "ACCOUNTS, VER
Y BORING"
610 IF L(17)=L THEN PRINT "YOUR JOB IS T
O STEAL THEM, NOT READ THEM"
620 PRINT:PRINT "PRESS RETURN TO CONTINU
E"

```

```

625 INPUT L$:RETURN
630 IF (L(8)=0 OR L(13)=0) AND (CT$="LAD" OR CT$="STO") AND L=3 THEN GOSUB 1820:RETURN
640 IF L=6 THEN PRINT "YOU CANNOT CLIMB OVER THE FENCE":RETURN
650 IF L=6 AND L(6)=35 AND L(9)=0 AND CT$="FEN" THEN GOSUB 1830:RETURN
660 IF L=6 AND L(6)<>35 AND L(9)=0 AND CT$="FEN" THEN GOTO 1850
670 RETURN
680 IF (L=20 AND CT$="MON" AND L(10)=0) THEN GOSUB 1860:RETURN
690 PRINT "THIS MACHINE CHANGES THE NOTE S YOU OWN INTO COINS"
695 PRINT "IT DOES NOTHING ELSE":RETURN
700 DDU=0:FOR M=1 TO 4:IF P(L,M)=-1 THEN DDU=1
710 NEXT M:IF DDU=1 AND L(14)=0 THEN KY=KY+1:IF KY=3 THEN KY=0:GOSUB 1880:RETURN
720 IF DDU<>1 OR L(14)<>0 OR CT$<>"DOO" THEN 790
730 PRINT "WITH A CREAK THE DOOR OPENS..."
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 L=22 THEN P(22,3)=21:P(21,4)=22
780 RETURN
790 IF L=19 THEN PRINT "YOU ARE TRAPPED INSIDE THE VAULT":IF L(14)<>0 THEN 1300
800 IF DDU<>1 THEN PRINT "THERE IS NO DOOR TO UNLOCK, FOOL!!!":RETURN
810 IF L(14)<>0 THEN PRINT "YOU DON'T HAVE 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 GUARD HERE!"
850 IF L(1)<>0 THEN PRINT "YOU HAVE NOTHING TO FIRE WITH"
860 IF LEFT$(CT$,2)<>"GU" THEN PRINT "SHOOT WHAT?"
870 IF L(1)=0 AND G1=L AND LEFT$(CT$,2)="GU" AND GS<>1 THEN GOSUB 2000
880 RETURN
890 IF INT(3*RND(1))+1<2 THEN PRINT "YOU MISSED HIM":RETURN
900 PRINT "YOU KILLED HIM"
905 G1=INT(28*RND(1))+3:IF G1=L OR G1=28 OR G1=7 THEN 905
910 RETURN
920 PRINT "TIME PASSES.....":FOR TT=1 TO 2000: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 HAVE NOTHING TO DRINK FROM":RETURN
950 PRINT "YUK! IT TASTES AWFUL, BUT WHAT DO YOU 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 TOP SHELF":RETURN
980 IF L=21 THEN PRINT "THERE IS A COIN LOCK ON THE TOILET DOOR":RETURN
990 IF L=2 THEN PRINT "THERE IS A LOCKED DOOR TO THE WEST IT IS CONTROLLED"
992 IF L=2 THEN PRINT "VIA A CENTRAL COMPUTER, THE ONLY WAY TO OPEN IT"

```



```

994 IF L=2 THEN PRINT "IS TO CALCULATE T
HE CODE":RETURN
1000 IF L=28 THEN PRINT "THERE IS A SIGN
SAYING TO GO UP ENTER 'GO N' ";
1004 IF L=28 THEN PRINT "ONCE UP, TO GO
DOWN, ENTER 'GO S' ";
1006 IF L=28 THEN FOR TD=1 TO 1000:NEXT:
PRINT "I DON'T UNDERSTAND IT EITHER"
1008 IF L=28 THEN RETURN
1010 PRINT "I AM LOOKING BUT THERE IS LI
TTLE TO SEE":RETURN
1020 IF L(6)=0 THEN PRINT "THEY FIT YOU
PERFECTLY":G$="YOU ARE WEARING GLOVES"
1025 IF L(6)=0 THEN L(6)=35:RETURN
1030 PRINT "YOU ARE WEARING ALL THAT YOU
OWN":RETURN
1040 IF L(5)=0 AND RIGHT$(CT$,3)="MEM" T
HEN GOSUB 2030:RETURN
1050 IF L(5)=0 AND L=2 THEN GOSUB 2050:R
ETURN
1060 PRINT "HELP YOUR SELF, NO CLUES FRO
M ME":RETURN
1070 PRINT "HAVE YOU REALLY HAD ENOUGH"
1075 INPUT A$
1080 IF LEFT$(A$,1)="Y" THEN PRINT "OK..
...COWARD!":END
1090 RETURN
1100 IF LEFT$(CT$,3)<>"GUN" THEN GOTO 11
10
1102 IF MID$(N$,5,8)="SILENCER"ANDL(12)=0
ANDL(1)=0 THENGS=1:GOSUB 2070:RETURN
1110 IF LEFT$(CT$,3)<>"LOC" THEN GOTO 11
20
1115 IF MID$(N$,5,4)="COIN" AND CC=1 AND
L=21 THEN GOSUB 2080:RETURN
1120 PRINT "YOU CAN'T PUT THAT THERE":RE
TURN

```

```

1130 IF G1=L THEN 1150
1140 PRINT "THERE IS NO ONE NEAR YOU":RE
TURN
1150 INPUT "WITH WHAT":W$
1155 IF LEFT$(W$,4)="WIRE" AND L(9)=0 AN
D RND(1)>0.5 THEN GOSUB 2090
1160 IF W$="ROPE" AND L(16)=0ANDRND(1)>.
5THENPRINT"YOU FINISHED HIM OFF":G1=0
1170 IF LEFT$(W$,4)<>"WIRE" AND W$<>"ROP
E" THEN 1200
1180 IF G1=0 THEN G1=INT(28*RND(1))+3:IF
G1=28 OR G1=L OR G1=7 THEN 1180
1190 RETURN
1200 PRINT "YOU WILL HAVE TO FIGHT HIM W
ITH YOUR HANDS"
1204 IF INT(5*RND(1))=1 THEN PRINT "YOU
HAVE KILLED HIM":G1=0:GOTO 1180
1210 RETURN
1220 PRINT:PRINT"YOU WERE 100% SUCCESSFU
L IN YOUR MISSIONAND THE ORGANIZATION";
1225 PRINT "HAS KINDLY LET YOU KEEP THE
OBJECTS YOU 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 GOSUB 2180:PRINT "QQ11111111111111WEL
L DONE!!!":END
1290 PRINT:PRINT "YOU'RE SAFELY AWAY BUT
YOU DID NOT STEALTHE PAPERS ";
1292 PRINT "THE ORGANIZATION IS NOT
PLEASED...YOU HAVE BEEN SENT BACK ";
1294 PRINT "TO TRY AGAIN":FOR T=1 TO 30
00:NEXT T:RUN

```

```

1300 FOR T=1 TO 3000:NEXT T:FOR T=1 TO 6
:PRINT:NEXT:PRINT "E"
1305 POKE 53280,11:POKE 53281,0
1310 GOSUB 2140:PRINT "E ***** **
*** ***** * "
1315 REM YELLOW
1320 PRINT " * * * *
*"
1330 PRINT " ***** *
*"
1340 PRINT " * * * *
*"
1350 PRINT " * * * *****
*****"
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$,CT+1,3)
1425 IF MID$(N$,CT,1)<>" " THEN NEXT CT
1430 RETURN
1440 REM *** INITIALISATION ***
1450 DIM O$(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
1495 DATA STOOL,WIRECUTTERS
1500 DATA MONEY,COIN-MACHINE,SILENCER,LA
DDER,KEY,PAPER-CLIPS,ROPE,SECRET-PAPERS
1505 DATA PAPERS

```

```

1510 DATA THE MAIN ENTRANCE,THE RECEPTIO
N AREA,THE DOCUMENT ROOM
1515 DATA THE STAFF CANTEEN,THE DELIVERY
ROOM
1520 DATA A CORNER OF THE YARD NEXT TO A
N ELECTRIFIED FENCE
1524 DATA "THE ALLEY BEHIND THE COMPLEX,
A CAR IS WAITING TO TAKE YOU AWAY"
1530 DATA THE COMPUTER ROOM,THE MAINTENA
NCE AREA,A DARK CORRIDOR
1535 DATA THE RECORDS DEPT,AN UNTIDY OFF
ICE
1540 DATA THE TYPING POOL,THE RECREATION
ROOM,A NARROW CORRIDOR
1545 DATA A PLUSH CONFERENCE ROOM,AN UNL
IGHTED OFFICE
1550 DATA THE STRONGROOM NEXT TO THE U
AULT,THE STEEL VAULT,THE ACCOUNTS DEPT
1555 DATA THE EXECUTIVE WASHROOMS
1560 DATA A COLD BARE ROOM,A DARK CORRID
OR,A CLEAN NEW STOREROOM
1565 DATA A LARGE CUPBOARD,A BRIGHT HALL
WAY
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 MOUTAKGETSTEDROREACLICUTCHAO
PEUNLFIRSHOWAIDRISEALOOWEACOMCALDEC"
1605 C$=C$+"HELCLUQUIPUTHITFIGATTKIL"
1610 O$="GUNTORCOFMEMCOMGLOCUPSTOWIRMONC
OISILLADKEYCLIROPSECPAP"
1620 P$="PARDON? WHAT? RUBBISH!REPHRAS
E"
1630 L=1:L(1)=0:K=0:SA=10000:KY=0:GS=0

```



```

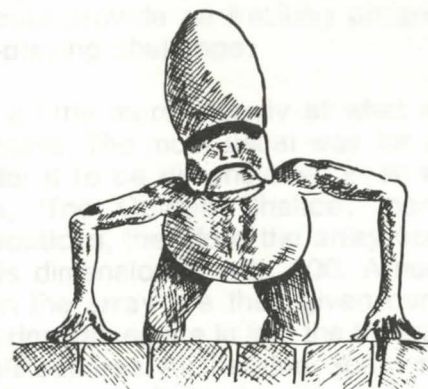
2130 NEXT: NEXT: POKE VO,0: POKE WF,0: POKE
AD,0: POKE SR,0: RETURN
2140 POKE VO,15: POKE AD,18: POKE SR,134: P
OKE WF,33
2150 POKE HF,5: POKE LF,103: FOR TD=1 TO 3
00: NEXT TD: POKE HF,0: POKE LF,0
2155 POKE 53280,10
2160 TG=TG+1: IF TG>3 THEN TG=0: POKE WF,0:
POKE AD,0: POKE SR,0: POKE 53280,11: RETURN
2170 FOR TD=1 TO 100: NEXT TD: POKE 53280,
6: GOTO 2150
2180 POKE VO,15: POKE AD,18: POKE SR,18: PO
KE WF,17
2190 FOR H1=50 TO 200 STEP 10: FOR L1=50
TO 200 STEP 10: POKE HF,H1: POKE LF,L1
2200 NEXT: NEXT: POKE AD,0: POKE SR,0: POKE
WF,0: RETURN
2500 POKE AD,32: POKE SR,132: POKE WF,129
2510 FOR GS=15 TO 1 STEP -1: POKE VO,GS
2520 POKE HF,10: POKE LF,120: FOR DD=1 TO
25: NEXT: NEXT
2530 POKE VO,0: POKE WF,0: RETURN

```

CHAPTER FOUR

Chapter Four

Other Forms of Adventure



CHAPTER FOUR

So far, we have only considered the more standard type of adventures. If you remember, in the first chapter I mentioned that there were many varied styles and formats of adventure. This chapter attempts to cover these other forms of adventure.

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 for the Commodore 64 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 than other types, although they can still provide much 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 100 different locations, therefore the array containing the locations is dimensioned with 100. 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 the 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 its type. For example, in The Golden Chalice, all the squares containing monsters are given the value 113. In this 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 to handle each square. You will have a number of IF ... THEN statements telling the computer that if the square the player is standing on contains a monster (i.e. in this game is equal to 113), 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 will have only four subroutines of this nature to write. Many games, of course, are larger and more complex than this.

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 10 by 10 grid. This does not have to be the case. Any size grid within memory requirements is possible. A diagram of the grid is shown at the top of page 93.

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

If the player presses the key corresponding to rightwards movement on the grid, 1 is added to his position, moving him rightwards. From looking at the diagram above, it is quite simple to see that to move left 1 is deducted; to move up/north 10 is deducted; while 10 is added to the adventurer's position if he asks to move downwards/southwards.

With these pieces written, it is now necessary to add frills to the program. Lines will be needed to stop a player moving off the boundaries of the grid and win and lose routines need to be developed. All other embellishments are really up to the person writing the adventure. 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, an objective to the game, whether it is to accumulate as much treasure as possible or to find the secret code to the IBM mainframe, or whatever.

You can see by now how this form of adventure satisfies all the conditions of an adventure we put forward in the first chapter.

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

Now on specifically to our 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? There's a lot more to it than that. You have around 40 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. We will tell you about two of them. Firstly, you need four pieces of rope to climb out of the openings of the cave roof. Secondly, if you are carrying the map then press 'M' 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, W, remembering that, at the start, you are placed near the centre of the cave system.

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 objects and obstructions are represented by single characters. Tough! We will leave you to sort out the map as yet another problem to confront while trying to find 'The Golden Chalice'.

Here is the listing:

```
10 REM *** THE GOLDEN CHALICE ***
15 PRINT "s":GOSUB 1290:REM CLR SCRIN
20 DIM A(100):P=0:C=P:MK=P:SC=P:M=P
30 FOR T=1 TO 100:ACT)=INT(100*RND(1))+1
:IF ACT)<56 THEN ACT)=46
40 IF ACT)>55 AND ACT)<66 THEN ACT)=122
50 IF ACT)>65 AND ACT)<76 THEN ACT)=113
60 IF ACT)>75 AND ACT)<86 THEN ACT)=119
70 IF ACT)>85 AND ACT)<96 THEN ACT)=32
80 IF ACT)>95 AND ACT)<101 THEN ACT)=166
90 NEXT T
```



```

537 PRINT "IT THIS WILL COST YOU THREE H
OURS IN     TIME"
540 A(MP)=46:GOTO 370
550 REM *** GOLDEN CHALICE ***
560 PRINT TAB(7) "IN THE TREASURE ROOM"
570 PRINT "BESIDE YOU, THE CRYSTAL STATU
E"
580 PRINT "IN ITS HAND THE GOLDEN CHALIC
E"
590 A(GC)=46:GOTO 370
600 REM *** FLOOD ***
610 PRINT TAB(7) "UP TO YOUR NECK IN WAT
ER":PRINT "THE FLOOD HAS COME"
620 PRINT "YOU MANAGE TO SWIM TO SAFETY
BUT YOU     WASTE 10 HOURS"
625 TM=TM-10:GOTO 370
630 REM *** MONSTER ***
640 RESTORE:FOR A=1 TO INT(5*RND(1))+1:R
EAD M$:NEXT
650 DATA SERPENT,BACKBREAKER,GRIZZLY,GIA
NT RAT,WOLFMAN
660 PRINT TAB(7) "IN DEEP TROUBLE, YOU H
AVE AWOKEN THE ":M$
670 PRINT "YOU MUST EITHER FIGHT OR FLEE
, ENTER     YOUR CHOICE ":
675 PRINT "(1 - FIGHT, 2 - FLEE)"
680 INPUT F:IF F=2 THEN X=X+INT(5*RND(1)
+1)-INT(5*RND(1)+1):TM=TM-6
685 IF F=2 THEN PRINT "YOU ARE SAFE FOR
THE MOMENT":FOR TD=1 TO 1000:NEXT TD
687 IF F=2 THEN GOTO 370
690 R=INT(3*RND(1)):IF R=1 THEN PRINT "Y
OU KILLED HIM WITH NO EFFORT"
695 IF R=1 THEN PRINT "WELL DONE!":TM=TM
+10:MK=MK+1:GOTO 370
700 IF R=2 THEN PRINT "AFTER A FIERCE BA
TTLE, HE STARTS TO     RETREAT ":

```

```

705 IF R=2 THEN PRINT "YOU HAVE WON!":TM
=TM-3:MK=MK+1:GOTO 370
710 PRINT "THE BATTLE WAS HARD, YOU FOUG
HT BRAVELY BUT THE ":M$:" WON!"
712 FOR TD=1 TO 1000:NEXT TD
715 GOTO 890
720 REM *** QUICKSAND ***
730 PRINT TAB(7) "IN QUICKSAND AND SINKI
NG FAST!!!"
740 GOSUB 1030
750 GOTO 890
760 REM *** OPENING ***
770 PRINT TAB(7) "STANDING BELOW A SHAFT
OF LIGHT":PRINT "THERE IS AN OPENING"
780 GOSUB 1260
790 IF P=4 THEN GOTO 970
800 GOTO 370
810 REM *** OUT OF TIME ***
820 FOR X=0 TO 8:FOR Y=8 TO 0 STEP -1
830 POKE 53280,X:POKE 53281,Y:FOR TD=1 T
O 50:NEXT:NEXT:NEXT
840 POKE 53280,0:GOSUB 1030
850 PRINT "sQQQQQQ[1111111111]":PRINT "E
TIME IS UP!!!":FOR TD=1 TO 1000:NEXT TD
860 REM CLEAR SCRN CRSR DOWN CRSR RIGHT
WHITE
890 REM *** DEFEAT ***
895 POKE 53280,0:POKE 53281,2
900 PRINT "s":PRINT "uYOU HAVE BEEN DEFE
ATED, THE CAVES MAKE AN IDEAL ":
901 REM YELLOW
905 PRINT "BURIAL CHAMBER FOR YOU AND
OTHERS WHO 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
IS ":SC

```

```

940 GOSUB 1210:END
970 REM *** VICTORY ***
975 POKE 53280,11:POKE 53281,2
980 PRINT "srr00QXXXXXXXXXXXXX R↑WELL DON
ErXXXXXXXXXXXXX"
985 REM CLR SCRN YELLOW CRSR DOWN REV ON
GREEN REV OFF YELLOW
990 PRINT:PRINT "QYOU MANAGED TO ESCAPE
FROM THE CAVES OF DELIRIUM"
995 REM CYAN
1000 IF A(GC)<>46 THEN 1020
1010 PRINT "AND WITH THE PRICELESS GOLDE
N CHALICE":GOSUB 1140
1020 SC=SC+300:GOSUB 1030:PRINT:GOTO 910
1030 POKE V0,15:POKE AD,18:POKE SR,36:PO
KE WF,17
1040 FOR H1=100 TO 10 STEP -10:FOR L1=10
0 TO 10 STEP -10
1050 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 7
5:NEXT
1060 POKE HF,0:POKE LF,0:FOR TD=1 TO 50:
NEXT
1070 NEXT:NEXT
1080 POKE V0,0:POKE WF,0:RETURN
1090 POKE V0,15:POKE AD,33:POKE SR,129:P
OKE WF,17
1100 POKE HF,17:POKE LF,37:FOR TD=1 TO 1
75:NEXT TD
1110 POKE HF,14:POKE LF,107:FOR TD=1 TO
250:NEXT TD
1120 POKE V0,0:POKE WF,0:POKE AD,0:POKE
SR,0
1130 RETURN
1140 POKE V0,15:POKE AD,33:POKE SR,33:PO
KE WF,17
1150 H1=INT(10*RND(1))+15:L1=INT(20*RND(
1))+10

```

```

1170 POKE HF,H1:POKE LF,L1:FOR TD=1 TO I
NT((4*RND(1))+1)*100:NEXT TD
1180 POKE HF,0:POKE LF,0:FOR TD=1 TO 50:
NEXT TD
1190 PG=PG+1:IF PG>3 THEN PG=0:RETURN
1200 GOTO 1150
1210 POKE V0,15:POKE AD,18:POKE SR,33:PO
KE WF,33
1220 POKE HF,4:POKE LF,208:FOR TD=1 TO 5
0:NEXT TD
1230 POKE HF,3:POKE LF,54:FOR TD=1 TO 10
0:NEXT TD
1240 SL=SL+1:IF SL>5 THEN GG=0:POKE V0,0
:POKE WF,0:RETURN
1250 GOTO 1220
1260 POKE V0,15:POKE AD,18:POKE SR,164:P
OKE WF,65:HP=54275:LP=54274
1265 POKE HP,200:POKE LP,15
1270 POKE HF,9:POKE LF,159:FOR TD=1 TO 3
50:NEXT TD
1280 POKE V0,0:POKE WF,0:FOR TD=1 TO 100
0:NEXT TD:RETURN
1290 POKE 53281,0:POKE 53280,0
1300 PRINT "aR00000000111111111THE GOLDEN
CHALICE"
1305 REM ORANGE REV ON DOWN CRSR RIGHT C
RSR
1310 RETURN

```


CHAPTER FIVE

Chapter Five

The City of Atlantis

Many haven't life in your...
cope with the...
under the...
heads, as the...
its energy...

After being asked...
instructions...
location, the...
carrying the...
command...
NORTH...
GET FLO...
the command...
These codes...
and...
quickly. If you...
the command...
all then...
Then, the...



Jm '83

Chapter Five

The City of Atlantis



CHAPTER FIVE

This adventure game was written by Ian Watt. Ian is an experienced adventure programmer who has contributed many adventures 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 by us to the Commodore 64.

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 watched by an Atlantan guard, you must find your way back to the diving ship you left some time ago.

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

After being asked whether or not you require instructions, 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 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. The computer will then prompt you with the word 'OBJECT'. Enter, then, 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 the pause starts and the program will continue.

This program, while being an adventure of the standard kind, is written in a different style to the others in this book and therefore provides an interesting contrast. There is no right way or wrong way to write an adventure program and many methods result in 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 Arena.
- 6) Advanced Part Of City.
- 7) Primitive Part Of City.
- 8) Old Part Of 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 City.
- 17) Passage Out Of 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 Cliff.

Commands:

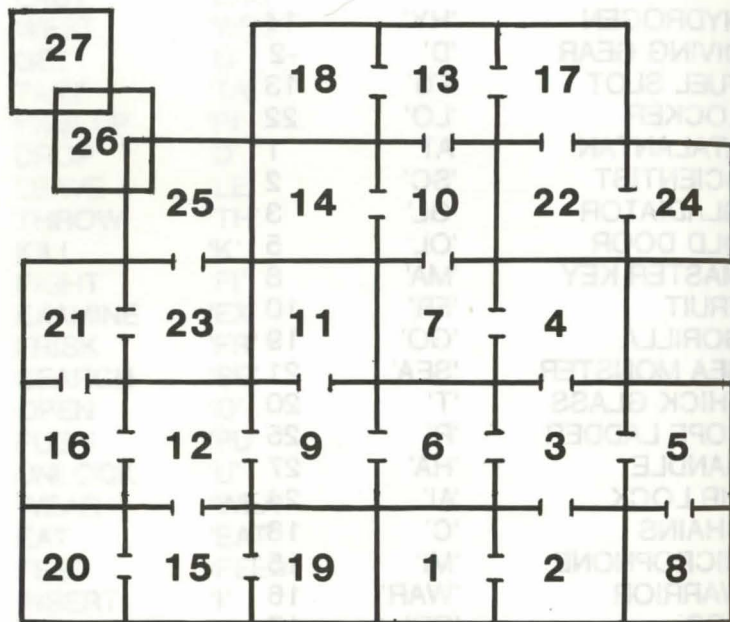
NORTH	'N'
SOUTH	'SO'
EAST	'EAS'
WEST	'WES'
GET	'G'
TAKE	'TA'
PICK UP	'PI'
DROP	'D'
LEAVE	'LE'
THROW	'TH'
KILL	'K'
FIGHT	'FI'
EXAMINE	'EX'
FRISK	'FR'
SEARCH	'SE'
OPEN	'O'
PUSH	'PU'
UNLOCK	'U'
WEAR	'WEA'
EAT	'EAT'
FEED	'FEED'
INSERT	'I'
CLIMB	'C'
QUIT	'Q'
RAISE	'R'
PLANT	'PL'
VAULT	'V'

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
ATALANTAN	'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

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



The listing for The City of Atlantis starts at the top of page 111.

```

10 REM *****
20 REM *** CITY OF ATLANTIS ***
30 REM *****
40 X=0:PRINT "s"
45 REM CLR SCRIN
50 GOSUB 1520
60 GOSUB 1630
70 PRINT "QDO YOU WANT INSTRUCTIONS (Y/
N)";:INPUT Z$:IF Z$="Y" THEN 80
71 REM DOWN CRSR
75 IF Z$<>"Y" THEN 160
80 PRINT "s":PRINT "WHILE ON A DIVING EX
PEDITION YOU WERE CAPTURED BY ";
82 PRINT "PEOPLE FROM THE SEA AND YOU AR
E NOW A PRISONER OF THESE PEOPLE"
90 FOR Z=1 TO 3000:GET A$:IF A$<>" " THEN
GOTO 100
95 NEXT
100 PRINT "s":PRINT "YOU HAVE TO DEPEND
ON GIVING THE COMPUTER ";
101 REM CLR SCRIN
102 PRINT "INSTRUCTIONS SO THAT YOU MAY
ESCAPE"
110 FOR Z=1 TO 3000:GET A$:IF A$<>" " THE
N GOTO 120
115 NEXT
120 PRINT "s":PRINT "ENTER A COMMAND OR
AN OBJECT AS THE COMPUTER ";
121 REM CLR SCRIN
122 PRINT "PROMPTS YOU - YOU WILL FIND
THAT, FOR EACH WORD YOU WILL ";
124 PRINT "ENTER AT LEAST THE NUMBER O
F LETTERS THAT WILL DETERMINE ";
126 PRINT "ONE WORD FROM ANOTHER"
130 FOR Z=1 TO 3000:GET A$:IF A$<>" " THE
N GOTO 140

```

```

135 NEXT
140 PRINT "s_";PRINT "IF ONE COMBINATION
OF COMMAND AND OBJECT DOESN'T WORK ";
141 REM CLR SCRIN
142 PRINT "THEN TRY ANOTHER"
150 FOR Z=1 TO 3000:GET A$:IF A$<>" " THE
N GOTO 160
155 NEXT
160 PRINT "s_";A=1:W=0:U=0:U=0
165 REM CLR SCRIN
170 W=W+1:RESTORE:FOR T=1 TO 1000:GET A$
:IF A$<>" " THEN GOTO 180
175 NEXT
180 PRINT "s_";IF E(6)=-2 AND U<6 THEN U=
U+1
185 REM CLR SCRIN
190 IF (A=6 OR A=11) AND E(11)=0 THEN GO
SUB 1800:GOTO 720
200 IF W>35 AND E(1)<>-2 THEN GOSUB 1810
:GOTO 720
210 IF A=21 AND E(12)=A AND E(11)=0 THEN
E(11)=-2:E(12)=-2:GOSUB 1820
220 IF E(20)=A AND E(24)=-1 THEN GOSUB 1
830:E(20)=-2:E(21)=A
230 IF E(25)=A AND E(26)=0 THEN E(26)=-2
:E(27)=0:GOSUB 1840
240 FOR B=1 TO A:READ A$:NEXT B:PRINT "1
1";A$:REM RIGHT CRSR
250 PRINT "EXITS:- :";IF A(A)<>0 THEN P
RINT " _NORTH: ";
255 REM PURPLE RED
260 IF B(A)<>0 THEN PRINT " _SOUTH: ";
265 REM BLACK RED
270 IF C(A)<>0 THEN PRINT " _EAST: ";
275 REM GREEN RED
280 IF D(A)<>0 THEN PRINT " _WEST: ";
285 REM BLUE RED

```

```

290 PRINT:PRINT "OBJECTS:-a";
295 REM ORANGE
300 H=0:FOR C=1 TO 30:IF E(C)<>A THEN NE
XT:GOTO 310
305 PRINT " ";H$(C);:NEXT
310 PRINT:PRINT " [-----
-----]";REM RED
320 PRINT:PRINT "INVENTORY:- z";:REM LT
BLUE
330 F=0:FOR C=1 TO 30:IF E(C)<>0 AND E(C
)<>-1 OR F=6 THEN NEXT:GOTO 340
335 PRINT " ";H$(C);:NEXT C
340 PRINT:PRINT:PRINT " [-----
-----]";REM RED
350 IF A=16 AND E(17)<>0 THEN GOSUB 1850
:GOTO 720
360 PRINT "COMMAND";:D$="":M=0:N=0
370 IF F>1 AND A=26 THEN PRINT "THE ROPE
LADDER HAS BROKEN...":GOTO 720
380 IF A=16 AND E(17)<>0 THEN GOSUB 1850
:GOTO 720
390 IF A=16 THEN PRINT "QTHE WARRIOR SEE
S YOUR CHAINS AND THINKS YOU WILL ";
391 REM DOWN CRSR
395 IF A=16 THEN PRINT "NOT GO FAR"
400 IF E(18)=0 THEN GOSUB 1860:GOTO 720
410 IF A=21 AND E(12)=A THEN PRINT "QTHE
SEA MONSTER HAS CAUGHT YOU!!!":GOTO 720
420 INPUT D$:DD$=LEFT$(D$,3):M=0
422 FOR C=1 TO 27
424 IF LEFT$(DD$,1)=B$(C)ORLEFT$(DD$,2)=B
$(C)ORDD$=B$(C)THENM=C:C=27:NEXT C
426 IF M>0 THEN 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
445 GOTO 470

```



```

450 PRINT "I'M SORRY BUT I DO NOT KNOW H
OW TO ":PRINT D$;" SOMETHING":GOTO 170
460 PRINT "I'M SORRY BUT I DO NOT KNOW W
HAT A ":PRINT D$;" IS":GOTO 170
470 PRINT "OBJECT ":D$="":N=0
480 INPUT E$:EE$=LEFT$(E$,3):FOR C=1 TO
30
482 IFEE$=G$(C)ORLEFT$(EE$,1)=G$(C)ORLEF
T$(EE$,2)=G$(C)THENN=C:C=30:NEXT:GOTO490
483 REM PUT CRSR BACK INTO LINE BEFORE P
RESSING RETURN
485 NEXT C
490 PRINT C$(M);:IF M>0 AND M<5 OR M=24
OR M=28 OR M=29 THEN 500
495 PRINT " ";H$(N)
500 PRINT
505 IF M>17 THEN GOTO 515
510 ONMGOTO590,590,590,590,740,740,740,7
90,790,790,840,920,850,850,850,960,960
515 M=M-17:ONMGOTO 960,1050,1100,1130,11
70,1210,1240,1250,1310,1350:M=M+17
520 PRINT "I CANNOT DO THAT":GOTO 170
530 PRINT "I AM ALREADY CARRYING IT":GOT
O 170
540 PRINT "I AM CARRYING TOO MUCH":GOTO
170
550 PRINT "O.K.":GOTO 170
560 PRINT "I DO NOT SEE IT HERE":GOTO 17
0
570 PRINT "I AM NOT CARRYING IT":GOTO 17
0
580 PRINT "I DO NOT SEE A PLACE TO PUT I
T":GOTO 170
590 IF (M<>3 AND A=1) THEN V=1:GOSUB 187
0:GOTO 170
600 IF M=3 AND A=1 AND V=0 THEN GOSUB 18
80:GOTO 170

```

```

610 IF V=1 THEN V=0:GOSUB 1890
620 IF E(7)=A THEN PRINT "THE GLADIATOR
WILL 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 O
RG IS IN THE WAY":GOTO 170
650 IF E(29)=A AND M=1 THEN GOSUB 1900:G
OTO 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
715 GOTO 170
720 PRINT:PRINT "PRESS SPACE BAR TO STAR
T AGAIN"
730 GET A$:IF A$<>" " THEN 730
735 RUN
740 IFN<>1ANDN<>2ANDN<>9ANDN<>10ANDN<>15
ANDN<>17ANDN<>18ANDN<>21ANDN<>23THEN745
742 GOTO 750
745 IFN<>24ANDN<>26ANDN<>28ANDN<>30THEN5
20
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 DON'T HAVE ENOUGH ENERGY T
O DO THAT IN THE THIN AIR":GOTO 170
850 IF E(N)<>A AND E(N)<>0 AND E(N)<>-1
THEN 560
855 IF N<>6 AND N<>13 AND N<>16 AND N<>2
2 AND N<>23 THEN GOSUB 1910:GOTO 170
860 IF N=22 THEN PRINT "THE GROUND IS PR
EPARED TO PLANT SEEDS":GOTO 170
870 IF N=23 THEN PRINT "THERE IS A BUTTO
NHOLE IN THE JACKET":GOTO 170
880 IF N=16 THEN PRINT "TO OPEN THE AIRL
OCK IT MUST BE RAISED":GOTO 170
890 IF N=13 AND E(N)<>A THEN 560
895 IF N=3 THEN PRINT "SOME FISH ARE SWI
MMING ABOUT ON THE OTHER SIDE":GOTO 170
900 IF E(N)<>A THEN PRINT "I DON'T SEE H
IM HERE":GOTO 170
910 E(6)=-2:GOSUB 1920: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 N
O MATCH FOR THE GLADIATOR":GOTO 720
950 E(7)=-2:PRINT "YOU HAD JUST ENOUGH EN
ERGY TO DEFEAT THEGLADIATOR":GOTO 170
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 DON'T HAVE EN
OUGH ENERGY":GOTO 170
1000 E(N)=-2:PRINT "YOU HAVE OPENED THE
DOOR":GOTO 170
1010 IF E(9)<>0 THEN PRINT "I CANNOT DO
THAT...YET":GOTO 170
1020 PRINT "THE LOCKER OPENS...AND CLO
SES"
1030 IF E(2)=-2 THEN E(2)=A:PRINT "SOME
DIVING 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
1075 IF N=24 AND E(23)<>-1 THEN 580
1080 IF N=24 THEN GOSUB 1930:E(N)=-1:GOT
O 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 MEA
NT TO BE FOR SOME 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 NOTH
ING TO FEED IT WITH":GOTO 170
1160 E(10)=-2:E(11)=0:GOSUB 1940:GOTO 17
0
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:GOSUB 1950:GOTO 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
1232 IF A=26 THEN A=27:E(N)=A:GOTO 550
1234 IF A=25 THEN A=26:E(N)=A:GOTO 550
1236 IF A=27 THEN A=26:E(N)=A:GOTO 550
1240 PRINT "QQQQARE YOU SURE? (Y/N)":
1241 REM DOWN CRSR
1245 GET A$:IF A$="" THEN 1245
1247 IF A$="Y" THEN 720
1249 GOTO 170

```



```

1250 IF N<>16 THEN 520
1260 IF A<>24 THEN 560
1270 IF E(15)<>0 THEN GOSUB 1960:GOTO 170
1280 IF E(2)<>-1 THEN GOSUB 1970: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"
1305 PRINT:PRINT "QQSCORE: ";Y;"11BEST SCORE: ";X:GOTO 720
1307 REM DOWN CRSR RIGHT CRSR
1310 IF N<>27 THEN 520
1320 IF E(N)<>0 THEN PRINT "I AM NOT CARRYING 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:GOSUB 1980:GOTO 170
1350 IF N<>29 THEN 520
1360 IF A<>7 THEN 560
1370 IF E(28)<>0 THEN PRINT "I HAVE NOTHING TO VAULT WITH":GOTO 170
1380 A=10:PRINT "YOU HAVE MANAGED TO VAULT THE WALL":GOTO 170
1390 DATA DETENTION AREA,INDOCTRINATION AREA,SPORTS ARENA,BARREN GROUND
1392 DATA BLOCKED EXIT FROM ARENA,ADVANCED PART OF CITY,PRIMITIVE PART OF CITY
1394 DATA OLD PART OF CITY,GARDEN WALKWAY,FOOD GROWING AREA,CENTRAL DOME
1396 DATA ATLANTAN LECTURE THEATRE
1400 DATA NUCLEAR FUSION POWER REACTOR,WATER HYDROLYSIS AREA,RADIO STATION
1402 DATA BORDER OF CITY,PASSAGE OUT OF KINGDOM,SLAVE LABOUR AREA
1404 DATA ANIMAL SPECIMEN CENTRE,UNDERWATER OBSERVATION POINT,FORBIDDEN ZONE

```

```

1406 DATA EQUIPMENT STORAGE AREA
1410 DATA MURKY LAGOON,AIR LOCK,STEEP INCLINE,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,S,O,E,A,S,W,E,S,G,T,A,P,I,D,L,E,T,H,K,F,I,E,X,F,R,S,E,O,P,U,U,W,E,A,E,A,T,F,E,I,C,Q
1465 DATA R,P,L,U
1470 DATA NORTH,SOUTH,EAST,WEST,GET,TAKE,PICK UP,DROP,LEAVE,THROW,KILL,FIGHT
1475 DATA EXAMINE,FRISK,SEARCH,OPEN,PUSH,UNLOCK,WEAR,EAT,FEED,INSERT,CLIMB,QUIT
1477 DATA RAISE,PLANT,VAULT
1480 DATA H,Y,D,F,U,L,O,A,T,S,C,G,L,O,L,M,A,F,R,G,O,S,E,A,T,R,H,A,A,I,C,M,I,W,A,R,O,R,S,L,G,R,J,F,L
1482 DATA L,E,B,S,E,E,P,W,A,L,M,E
1490 DATA HYDROGEN,DIVING GEAR,FUEL SLOT,LOCKER,ATLANTAN,SCIENTIST,GLADIATOR
1492 DATA OLD DOOR,MASTER KEY,FRUIT,GORILLA,SEA MONSTER,THICK GLASS,ROPE LADDER
1494 DATA HANDLE,AIR LOCK,CHAINS,MICROPHONE,WARRIOR,ORG,SLEEPY ORG,GROUND
1496 DATA 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
1515 DATA -2,-2,7,23
1520 REM *** ARRAYS ***
1530 DIM L$(27):FOR T=1 TO 27:READ L$(T):NEXT

```

```

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
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
1620 RETURN
1630 PRINT "s":POKE 53280,6:POKE 53281,3
1640 PRINT "00000000111111111111";
1641 REM DOWN CRSR RIGHT CRSR
1645 PRINT "RCITY OF ATLANTIS"
1650 REM RED REV ON
1660 FOR T=1 TO 1500:NEXT:RETURN
1800 PRINT "THE GORILLA IS HUNGRY AGAIN.
":RETURN
1810 PRINT "THE POWER REACTOR HAS BECOME
UNSTABLE AND HAS BLOWN UP!!":RETURN
1820 PRINT "THE GORILLA HAS RUN AFTER TH
E SEA MONSTER":RETURN
1830 PRINT "THE FLOWER MAKES THE ORG SLE
EPY":RETURN
1840 PRINT "THE CRAZY LECTURER TAKES THE
BOOKLET ANDGIVES YOU SEEDS":RETURN
1850 PRINT "THE WARRIOR CATCHES YOU AS Y
OU TRY TO ESCAPE FROM THE CITY"
1855 RETURN
1860 PRINT "SOME ATLANTANS HEAR YOU ON T
HEIR RADIOS AND INVESTIGATE":RETURN

```

```

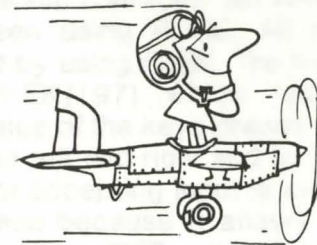
1870 PRINT "THE ATLANTAN LAUGHS AT YOU W
ALKING INTO THE WALL":RETURN
1880 PRINT "THE ATLANTAN WILL NOT LET YO
U PASS":RETURN
1890 PRINT "THE ATLANTAN IS DISTRACTED A
ND DOES NOT SEE YOU LEAVE":RETURN
1900 PRINT "THE WALL IS TOO SOLID FOR YO
U TO GO THROUGH IT":RETURN
1910 PRINT "I SEE NOTHING WORTH TAKING N
OTE OF":RETURN
1920 PRINT "HE GIVES YOU SOMETHING TO HE
LP YOU WITH THE ATMOSPHERE":RETURN
1930 PRINT "THE FLOWER GIVES OFF A PUNGE
NT ODOR BUT DOESN'T HARM YOU":RETURN
1940 PRINT "THE GORILLA FOLLOWS YOU IN T
HE HOPE OF GETTING MORE FOOD":RETURN
1950 PRINT "THE ADDITION OF MORE FUEL TO
THE REACTORHAS STABILIZED IT":RETURN
1960 PRINT "I HAVE NOTHING WITH WHICH I
CAN OPERATE THE RAISING MECHANISM"
1965 RETURN
1970 PRINT "YOU HAVE DROWNED IN THE WATE
R THAT HAS FLOODED THE AIRLOCK"
1975 RETURN
1980 PRINT "THE SEEDS GROW QUICKLY AND F
ORM TALL POLE VAULT SHAPED PLANTS"
1985 RETURN

```


CHAPTER SIX

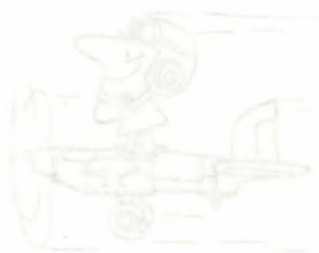
Chapter Six

Other Games for your Computer



Chapter Six

Other Games for your Computer



CHAPTER SIX

You may ask: why other games in an adventure book? 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 adventures. Secondly and perhaps the main reason, 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.

Raiders from Mars

This is the only moving graphics game in the book. It uses items from the graphic character set and moves them around the screen using POKE. All relevant collisions are detected by using PEEK. The keyboard is scanned using PEEK(197) which reads the keyboard buffer. The value of the keys pressed is then used to move your base left and right and to fire your missiles. This method of accepting input is invaluable in moving graphic games because it allows for key repeats, it is faster than using GET and it also rejects all inputs other than the ones you are using.

To move your firebase to the right use the '.' key, value 44 in line 65. To move it to the left, press the ',' key, value 47. The line also makes sure that your firebase doesn't move off the screen. These two keys were chosen because the 'less than' and 'greater than'

symbols on the top of the keys point in the correct directions!

In line 90, if K equals 21, the value of the 'F' key — then we go to the missile firing routine, provided a missile isn't already on the screen.

This program has all the basic elements required for a moving graphics game and puts them all together most effectively.

It has:

1. Moving targets.
2. A moving firebase.
3. Laser or missile firing.
4. The aliens shoot back.
5. Good explosions.
6. An end routine with score and high score features.
7. A start routine.

The only feature it doesn't have is sound effects during the playing of the game. These were left out because they slowed down the action and, in a moving graphics game, the action is the most important element.

```
10 REM *** RAIDERS FROM MARS ***
30 GOSUB 1200
40 GOSUB 2000
50 L=3: X=18: C=54272: POKE 53280,0: POKE 53281,0
55 PRINT "SCORE: "; S; "SHIPS: "; L
56 REM CLR SCRN HOME RIGHT CRSR YELLOW N
EXT LINE HOME DOWN CRSR
```

```
57 PRINT "SCORE: "; S; "SHIPS: "; L
58 PRINT "REMAINS: "; REM; "KEYS: "; KEYS
59 REM COMM KEY AND @ KEY
60 FOR I=0 TO 39: POKE 1944+I,99: POKE 1944+C+I,1: NEXT
65 XI=X: K=PEEK(197): X=X+(K=42 AND X>0)-(K=50 AND X<37): IF XI<>X THEN ER=1
70 IF ER=1 THEN POKE 1904+XI,32: POKE 1905+XI,32: POKE 1906+XI,32: ER=0
80 POKE 1904+X,112: POKE 1905+X,113: POKE 1906+X,110
85 POKE 1904+X+C,1: POKE 1905+X+C,10: POKE 1906+X+C,1
90 IF K=10 THEN IF SF=0 THEN SF=1: BX=X+1-40: BY=22
100 IF SF=1 THEN POKE 1024+BX+40*BY,32: BY=BY-1: PE=PEEK(1024+BX+40*BY)
102 IF SF=1 THEN IF PE=107 OR PE=88 OR PE=115 THEN A1=0: GOSUB 500
107 IF SF=1 THEN IF PE=40 OR PE=42 OR PE=41 THEN A2=0: GOSUB 500
110 IF SF=1 THEN IF BY<=3 THEN SF=0
120 IF SF=1 THEN POKE 1024+BX+40*BY,93: POKE 1024+BX+40*BY+C,4
130 IF A1=0 AND RND(1)>.5 THEN GOSUB 600: X1=X5: Q=AS: M1=M5: A1=1
140 IF A2=0 AND RND(1)>.5 THEN GOSUB 600: X2=X5: Q2=AS: M2=M5: A2=1
150 IF A1=1 THEN A=Q+X1: M=M1: F=1: GOSUB 700: X1=A-Q: A1=F: IF X1<0 OR X1>36 THEN A1=0
155 IF A1=1 THEN POKE Q+X1,107: POKE Q+X1+1,88: POKE Q+X1+2,115
157 IF A1=1 THEN POKE Q+X1+C,7: POKE Q+X1+1+C,14: POKE Q+X1+2+C,7
160 IF A2=1 THEN A=Q2+X2: M=M2: F=1: GOSUB 700: X2=A-Q2: A2=F: IF X2<0 OR X2>36 THEN A2=0
165 IF A2=1 THEN POKE Q2+X2,40: POKE Q2+X2+1,42: POKE Q2+X2+2,41
```

```

167 IF A2=1 THEN POKE Q2+X2+C,4:POKE Q2+
X2+1+C,8:POKE Q2+X2+2+C,4
170 RS=RND(1):IF RS<.05 THEN IF A2=1 THE
N N=Q2+X2+41:GOTO 800
180 IF RS>.95 THEN IF A1=1 THEN N=Q+X1+4
1:GOTO 800
190 GOTO 65
500 EX=1024+BX+40*BY:POKE EX,102:POKE EX
-1,102:POKE EX+1,102
502 POKE EX+C,10:POKE EX-1+C,10:POKE X+1
+C,10
505 POKE EX+40,102:POKE EX-40,102
507 POKE EX+40+C,10:POKE EX-40+C,10
510 S=S+50:PRINT "SSCORE:";S
530 POKE EX,32:POKE EX-1,32:POKE EX+1,32
:POKE EX+40,32:POKE EX-40,32
532 POKE EX+2,32:POKE EX-2,32
540 SF=0:RETURN
600 Y=INT(RND(1)*17)+2:AS=1024+40*Y:Z=IN
T(RND(1)*2):IF AS=Q OR AS=Q2 THEN 600
610 IF Z=1 THEN X5=37:M5=-1
620 IF Z=0 THEN X5=0:M5=1
630 RETURN
700 POKE A,32:POKE A+1,32:POKE A+2,32:A=
A+M
710 IF PEEK(A)=93 OR PEEK(A+2)=93 THEN G
OSUB 500:F=0
720 RETURN
800 O=N
810 POKE N,93:N=N+40:IF PEEK(N)=112 OR P
EEK(N)=113 OR PEEK(N)=110 THEN 850
820 IF N>1983 THEN FOR CL=0 TO N STEP 40
:POKE CL,32:POKE CL+C,1:NEXT:GOTO 65
825 IF N<1983 THEN POKE N-40,113:POKE N-
40+C,2:GOTO 810
830 FOR CL=0 TO N STEP 40:POKE CL,22:NEX
T

```

```

850 POKE UO,15:POKE AD,17:POKE SR,64:POK
E WF,33:FOR HN=1 TO 6
860 POKE HF,3:POKE LF,155:FOR TD=1 TO 25
:NEXT:POKE HF,7:POKE LF,53
865 FOR TD=1 TO 50:NEXT TD:NEXT HN
870 POKE UO,0:POKE WF,0
880 FOR I=0 TO N STEP 40:POKE I,32:POKE
I+C,1:NEXT
890 L=L-1:PRINT "S111111111111SHIPS:";L
:IF L=0 THEN 950
895 REM HOME RIGHT CRSR
900 GOTO 65
950 E=1905+X:C=56177+X:POKE E,160:POKE C
,2
960 POKE E-1,160:POKE E+1,160:POKE E-40,
160:POKE C-1,4:POKE C+1,4
965 POKE C-40,4:POKE E-2,102:POKE E+2,10
2:POKE E-41,160:POKE E-39,160
970 POKE E-80,102:POKE C-2,3:POKE C+2,3:
POKE C-41,3:POKE C-39,3
975 FOR TD=1 TO 3000:NEXT
1000 PRINT "SQQQQQ1111111111R *** GAME
OVER ***"
1005 REM HOME DOWN CRSR RIGHT CRSR
1010 PRINT "QQQ111111111111YOU SCORED";
S
1015 REM DOWN CRSR RIGHT CRSR CYAN
1020 IF S>HS THEN HS=S:PRINT "Q111zYOU GO
T THE HIGH SCORE!! WELL DONE!!"
1025 REM DOWN CRSR RIGHT CRSR LT BLUE
1030 PRINT "Q111111111111HIGH SCORE";HS
1035 REM DOWN CRSR RIGHT CRSR
1040 FOR TD=1 TO 5000:NEXT TD
1050 A2=0:A1=0:S=0:SF=0:GOTO 50
1200 UO=54296:AD=54277:SR=54278:WF=54276
:HF=54273:LF=54272:RETURN
2000 POKE 53280,0:POKE 53281,4

```



```

2010 PRINT "S";
2020 POKE VO,15:POKE AD,17:POKE SR,32:PO
KE WF,17
2030 FOR TL=1 TO 24
2040 H1=INT(50*RND(1))+150):L1=INT(50*RND
(1))+150)
2050 POKE HF,H1:POKE LF,L1:FOR TD=1 TO 1
5:NEXT
2060 PRINT "R***** RAIDERS FROM M
ARS *****";
2065 REM REV ON BLUE REV OFF
2070 NEXT TL
2080 POKE VO,0:POKE WF,0
2090 FOR TD=1 TO 500:NEXT TD
2100 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 in the middle. In this game you move by entering the co-ordinates of the peg you wish to move and then the co-ordinates of the position you wish to move to. The computer will reject any illegal moves. You enter the side number and then the top number, e.g. 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.

```

10 REM *** SOLITAIRE ***
20 GOSUB 400:REM INITIALISE
30 GOSUB 250:REM PRINT DISPLAY
40 REM *** MOVE ***
50 PRINT:PRINT "WHICH PEG DO YOU WISH T
O MOVE";:REM BLUE
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 "aMOVE TO WHICH HOLE";:R
EM ORANGE
110 INPUT B:GOSUB 640
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 LEF
T ON THE BOARD"
220 IF CO<>1 THEN 40
230 IF A(44)=79 THEN PRINT:PRINT "YOU HA
VE COMPLETED THE PUZZLE IN JUST"
235 IF A(44)=79 THEN PRINT MV;"MOVES";GO
SUB SOUND:END
240 PRINT:PRINT "THE GAME IS OVER AND YO
U HAVE FAILED":GOSUB 690:END
250 REM *** DISPLAY ***
260 PRINT "s":GOSUB 590:REM CLEAR SCREEN
270 PRINT TAB(5) "ENTER SIDE CO-ORDINAT
E FIRST":PRINT:REM PURPLE
280 PRINT TAB(6) "vENTER 99 TO CONCEDE D
EFEAT":PRINT:REM LT RED

```

```

290 PRINT:PRINT TAB(12) "w1 2 3 4 5 6 7"
:REM GREY 1
300 PRINT TAB(12);
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(12);:GOTO 350
340 PRINT CHR$(A(D));" ";
350 NEXT D:PRINT " 7"
360 PRINT:PRINT "NUMBER OF MOVES SO FAR:
";MV
370 RETURN
400 REM *** INITIALISE ***
405 POKE 53280,6:POKE 53281,3
410 PRINT "s":DIM A(87):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
495 V0=54296:AD=54277:SR=54278:WF=54276:
HF=54273:LF=54272
500 RETURN
510 REM *** DATA ***
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
590 POKE V0,15:POKE AD,17:POKE SR,132:PO
KE WF,17
600 POKE HF,51:POKE LF,97:FOR TD=1 TO 15
0:NEXT TD

```



```

610 POKE HF,0:POKE LF,0:FOR TD=1 TO 75:N
EXT TD
620 POKE HF,43:POKE LF,52:FOR TD=1 TO 20
0:NEXT TD
630 POKE V0,0:POKE WF,0:RETURN
640 POKE AD,18:POKE SR,132:POKE WF,17
650 FOR TT=15 TO 1 STEP -1:POKE V0,TT
660 POKE HF,14:POKE LF,107:FOR TD=1 TO 7
5:NEXT TD
670 NEXT
680 POKE V0,0:POKE WF,0:FOR TD=1 TO 750:
NEXT TD:RETURN
690 POKE V0,15:POKE AD,132:POKE SR,132:P
OKE WF,33
700 POKE HF,4:POKE LF,73:FOR TD=1 TO 500
:NEXT TD
710 POKE WF,0:POKE V0,0:RETURN

```

Four By Four

This is a family favourite, where you must attempt to get four of your counters in a row either horizontally, diagonally or vertically. Your pieces are represented by the yellow circles and the computer's by the purple 'spades' symbols. 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 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 TAB(9) "STAND BY FOR MY
MOVE...":REM LT RED
130 GOSUB 1220: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 FOUR 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 MV=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 MV=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 MV=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 MV=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 MV=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 MV=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 MV=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 MV=B+10
320 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 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 MV=B-11:GOTO 650

```

```

340 IF A(B+9)=X AND A(B+18)=X AND A(B-9)
=E AND A(B+1)<>E THEN MU=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 MU=B+2:GOTO 650
390 IF A(B+1)=X AND A(B-1)=E AND A(B+9)<
>E THEN MU=B-1:GOTO 650
400 IF A(B-1)=X AND A(B-2)=E AND A(B+8)<
>E THEN MU=B-2:GOTO 650
420 IF A(B+10)=X AND A(B-10)=E AND A(B)<
>E THEN MU=B-10:GOTO 650
440 IF A(B+9)=X AND A(B-9)=E AND A(B+1)<
>E THEN MU=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 MU=B-11:GOTO 650
460 GOTO 180
490 FOR N=1 TO 3
500 M(N)=0
510 NEXT N
520 CO=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 CO=CO
0+1:M(CO)=B+1
560 IF A(B-1)=E AND A(B+9)<>E THEN CO=CO
+1:M(CO)=B-1
570 IF A(B-10)=E AND A(B)<>E THEN CO=CO+
1:M(CO)=B-10
580 IF A(B-11)=E AND A(B-1)<>E THEN CO=CO
0+1:M(CO)=B-11
590 IF A(B-9)=E AND A(B+1)<>E THEN CO=CO
+1:M(CO)=B-9
600 NEXT B
610 IF CO<>0 THEN 640
620 PRINT:PRINT TAB(9) "I THINK IT IS A
DRAW!!":GOSUB 1240:END:REM BLACK
640 MU=M(INT(CO*RND(1)))+1)

```

```

650 A(MU)=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 TAB(9) "YOU HAVE
BEATEN ME!!!":REM LT GREEN
830 IF X=C THEN PRINT TAB(7) "I HAVE BE
ATEN YOU, HUMAN!!!":REM YELLOW
835 GOSUB 1240:END
850 REM *** PRINT BOARD ***
860 PRINT "S":PRINT "Z***** FOUR
BY FOUR *****":PRINT
865 REM CLEAR SCREEN LT BLUE
870 FOR K=10 TO 70 STEP 10
890 PRINT:PRINT TAB(13);
900 FOR J=1 TO 7:IF A(K+J)=113 THEN PRIN
T "I":REM YELLOW
905 IF A(J+K)=97 THEN PRINT "L":REM PUR
PLE
907 IF A(J+K)=46 THEN PRINT "U":REM GRE
EN
910 PRINT CHR$(A(K+J));" ";

```



```

920 NEXT J:NEXT K
935 PRINT:PRINT:PRINT TAB(13) "a1 2 3 4
5 6 7":PRINT:REM ORANGE
960 RETURN
970 REM *** HUMAN MOVE ***
990 PRINT TAB(12) "IT IS YOUR MOVE":REM
  CYAN
1000 PRINT "WHICH COLUMN DO YOU WISH TO
MOVE INTO..."
1010 INPUT 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":GOSUB
1240
1070 GOTO 1000
1080 REM *** INITIALISE ***
1100 PRINT "s":POKE 53280,10:POKE 53281,
11:REM CLR SCRN
1110 DIM A(109),M(30),P(6)
1120 V0=54296:AD=54277:SR=54278:WF=54276
1130 HF=54273:LF=54272
1140 E=46:H=113:C=97:REM ASC CODES FOR P
RINTING BOARD
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
1215 REM *** SOUND ROUTINES ***
1220 POKE V0,15:POKE AD,18:POKE SR,132:P
OKE WF,17
1230 POKE HF,9:POKE LF,159:FOR TD=1 TO 3
00:NEXT TD:POKE V0,0:POKE WF,0:RETURN

```

```

1240 POKE V0,15:POKE AD,33:POKE SR,132:P
OKE WF,33
1250 POKE HF,34:POKE LF,75:FOR TD=1 TO 2
50:NEXT TD
1260 POKE HF,0:POKE LF,0:FOR TD=1 TO 75:
NEXT TD
1270 TG=TG+1:IF TG>3 THEN TG=0:POKE V0,0
:POKE WF,0:RETURN
1280 GOTO 1240

```

Sound and
Graphics
for your own
Adventures

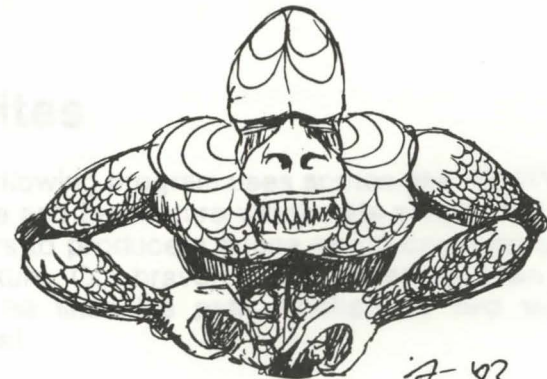
CHAPTER SEVEN

Chapter Seven

Sound and Graphics for your own Adventures

Sprites

The first feature of an adventure game is the sprite. A sprite is a small, two-dimensional image that is used to represent a character or object in the game. Sprites are typically created using a graphics editor and are then loaded into the game's memory. The game engine then uses the sprite to draw the character or object on the screen.



Jim '83

Chapter Seven

Sound and Graphics for your own Adventures



CHAPTER SEVEN

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 (including us) who believe that an occasional graphic display heightens the enjoyment that a user receives from playing the game, providing three factors are noted. The graphic displays should not be used in locations that are not often visited, as a player does not want to come across the same display too often. There should not be too many such displays in an adventure which is primarily text. Finally, the pictures should not take too long to generate. A picture of great complexity may look wonderful, but much of the magic is lost if it takes six or seven minutes to draw.

Sprites

The following program uses sprites, the REVERSE ON feature and the feature which sets screen and border colours to produce a scene which could occur in an adventure. Our brave hero is travelling down a road when he finds his path blocked by two fearsome wraiths!

```

10 REM *** GRAPHIC DISPLAY ***
20 PRINT "s";POKE 53281,0:POKE 53280,11
:GOSUB 2000:REM CLR SCREEN
30 V=53248:POKE V+21,28
40 POKE 2042,194:POKE 2043,195:POKE 2044
,196
50 FOR S1=12416 TO 12478:READ Q:POKE S1,
Q:NEXT
60 FOR S2=12480 TO 12542:READ Q2:POKE S2
,Q2:NEXT
70 FOR S3=12544 TO 12606:READ Q3:POKE S3
,Q3:NEXT
80 POKE V+43,14:POKE V+42,2:POKE V+41,10
90 POKE V+4,175:POKE V+5,200
100 POKE V+6,200:POKE V+7,150
110 POKE V+8,150:POKE V+9,150
130 GOTO 130
200 DATA 0,0,16,0,24,8,0,36,4,0,36,2,0,2
4,3,0,24,7,0,230,14,1,129,252
210 DATA 3,129,248,15,129,240,14,66,0,28
,66,0,56,60,0,112,62,0,224,85,0
220 DATA 64,5,0,1,5,0,2,5,0,5,5,0,10,5,0
,10,7,128
230 DATA 0,16,2,0,40,6,0,84,14,0,186,30,
0,84,62,0,40,126,0,56,254,0,223,230
240 DATA 1,215,226,255,239,192,255,95,12
8,127,224,0,96,222,0,97,156,0
250 DATA 97,172,0,66,216,0,2,184,0,7,88,
0,12,176,0,31,240,0,31,240,0
260 DATA 0,16,2,0,40,6,0,84,14,0,186,30,
0,84,62,0,40,126,0,56,254,0,223,230
270 DATA 1,215,226,255,239,192,255,95,12
8,127,224,0,96,222,0,97,156,0
280 DATA 97,172,0,66,216,0,2,184,0,7,88,
0,12,176,0,31,240,0,31,240,0,
2000 FOR XX=1 TO 400

```

```

2010 PRINT "R ";:REM REV ON BLUE ONE SP
ACE
2020 NEXT
2025 XY=14
2030 FOR XX=1424 TO 1984 STEP 40
2040 FOR X=XX TO XX+XY
2050 POKE X,160:POKE X+54272,5
2060 NEXT
2070 XY=XY-1
2080 NEXT
2085 XY=14
2090 FOR XX=1449 TO 2023 STEP 40
2100 FOR X=XX TO XX+XY
2110 POKE X,160:POKE X+54272,5
2130 NEXT
2135 XX=XX+1:XY=XY-1
2140 NEXT
3000 RETURN

```

Line 20 clears the screen, sets the screen and border colours and then calls up the routine which begins at line 2000. Lines 2000 to 2020 use the REVERSE ON feature to draw in the blue sky. Lines 2025 to 2080 draw in the green fields to the left of the black road and lines 2085 to 2140 do the same for the right hand side of the road.

Line 30 sets V equal to the beginning of the video RAM and then turns on sprites 2, 3, and 4. (POKE V+28). Line 40 then sets the memory pointers for these sprites. Lines 50 to 70 READ the sprite DATA into the relevant memory locations. We then set the sprite colours in line 80. The three sprites are positioned on the screen by lines 90 to 110. Line 130 sets up an endless loop to keep the scene on the screen.

Lines 200 to 280 contain all the DATA to draw the sprites.

Title Pages

By using the REVERSE ON feature and a combination of the two graphic characters from the asterisk and L keys with a series of reversed spaces, we can produce very impressive title screens for our programs. The short routine below demonstrates how to do this.

```
10 REM *** TITLE PAGE ***
20 POKE 53280,2:POKE 53281,0
30 PRINT "a":REM ORANGE
50 PRINT "s":REM CLR SCRIN
60 REM "R" IS REV ON
70 REM "r" IS REV OFF
80 REM "□" IS COMMODORE KEY PLUS * KEY
90 REM "r" IS SHIFT KEY PLUS & KEY
100 PRINT "QQQQ"
105 REM DOWN CRSR
110 PRINT TAB(6) "R  r  R  r  R  r  R  r"
120 PRINT TAB(6) "R  r  R  r  R  r  R  r"
130 PRINT TAB(6) "R  r  R  r  R  r  R  r"
140 PRINT TAB(6) "R  r  R  r  R  r  R  r"
150 PRINT TAB(6) "R  r  R  r  R  r  R  r"
160 PRINT "QQ"
170 PRINT TAB(8) "R  □  R  □  R  □  R  □"
180 PRINT TAB(8) "R  r  R  r  R  r  R  r"
190 PRINT TAB(8) "R  r  R  r  R  r  R  r"
```

```
190 PRINT TAB(8) "R  r  R  r  R  r  R  r"
200 PRINT TAB(8) "R  r  R  r  R  r  R  r"
210 PRINT TAB(8) "R  r  R  r  R  r  R  r"
220 GOTO 220
```

Sound Effects

Now we turn our attention to sound. The Commodore 64 has a very good three-voice sound synthesiser which can be used to great effect in adventure programs. Most of the examples below use only one of the voices, however, the routine called 'Scales 3' uses voices one and three together. The short routines below are just examples of the type of effects you can create with your Commodore 64. Use these routines as starting points for your own experimentation. The titles in the REM statements are generally self-explanatory.

```
10 REM *** SCALES 1 ***
20 V0=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 POKE V0,15:POKE AD,18:POKE SR,64:POKE
WF,17
40 FOR H1=2 TO 102 STEP 5:FOR L1=10 TO 2
00 STEP 10
50 POKE HF,H1:POKE LF,L1
70 NEXT:NEXT
80 FOR H1=107 TO 7 STEP -5:FOR L1=210 TO
20 STEP -10
```

```

90 POKE HF,H1:POKE LF,L1
100 NEXT:NEXT
110 POKE WF,0:POKE LF,0

```

```

10 REM *** SCALES 2 ***
20 U0=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 POKE U0,15:POKE AD,18:POKE SR,64:POKE
WF,17
40 FOR H1=102 TO 2 STEP -5:FOR L1=10 TO
200 STEP 10
50 POKE HF,H1:POKE LF,L1
70 NEXT:NEXT
80 FOR H1=7 TO 107 STEP 5:FOR L1=210 TO
20 STEP -10
90 POKE HF,H1:POKE LF,L1
100 NEXT:NEXT
110 POKE WF,0:POKE U0,0

```

```

10 REM *** SCALES 3 ***
20 U0=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 A3=54291:S3=54292:W3=54290:H3=54287:L
3=54286
40 POKE U0,15:POKE AD,17:POKE A3,15:POKE
SR,132:POKE S3,132
50 POKE WF,33:POKE W3,17
60 FOR H1=4 TO 44 STEP 2:FOR L1=50 TO 25
0 STEP 10
70 POKE HF,H1:POKE LF,L1:POKE H3,H1-1:PO
KE LF,L1-10
80 NEXT:NEXT
90 POKE U0,0:POKE WF,0:POKE W3,0

```

```

10 REM *** RANDOM NOTES ***
20 U0=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272:HP=54275:LP=54274
30 POKE U0,15:POKE AD,17:POKE SR,132:POK
E WF,65:POKE HP,4:POKE LP,200
35 FOR TT=1 TO 6
40 H1=INT(20*RND(1))+2:L1=INT(180*RND(1)
)+10
50 POKE HF,H1:POKE LF,L1
60 T1=INT(3*RND(1)+1)*100
70 FOR TD=1 TO T1:NEXT
80 POKE HF,0:POKE LF,0
90 FOR TD=1 TO 50:NEXT
100 NEXT
110 POKE U0,0:POKE WF,0

```

```

10 REM *** FADING NOTE ***
20 U0=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 POKE AD,32:POKE SR,255:POKE WF,17
40 FOR TG=15 TO 1 STEP -1
50 POKE U0,TG
60 POKE HF,12:POKE LF,216
70 FOR TD=1 TO 100:NEXT TD
80 NEXT TG
90 POKE U0,0:POKE WF,0

```

```

10 REM *** PHOTON TORPEDOES ***
20 U0=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 POKE AD,15:POKE SR,147:POKE WF,129
40 FOR TG=15 TO 1 STEP -1:POKE U0,TG
50 POKE HF,2:POKE LF,6

```



```

60 FOR TD=1 TO 50:NEXT TD
70 NEXT TG
90 FOR TD=1 TO 100:NEXT:POKE VO,0:POKE A
D,0:POKE WF,0
100 R=R+1:IF R>3 THEN R=0:END
110 GOTO 30

```

```

10 REM *** WHIZZ ***
20 VO=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272:HP=54275:LP=54274
25 POKE AD,17:POKE SR,132:POKE WF,33:POK
E VO,15
30 H1=5:L1=10
40 POKE HF,H1:POKE LF,L1
50 FOR TD=1 TO 2:NEXT
60 H1=H1+5:L1=L1+10
70 IF L1<250 THEN GOTO 40
80 POKE VO,0:POKE WF,0

```

```

5 REM *** JET ***
10 VO=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
20 POKE VO,15:POKE AD,17:POKE SR,17:POKE
WF,129
30 FOR H1=20 TO 70:FOR L1=2 TO 200 STEP
4
40 POKE HF,H1:POKE LF,L1
60 NEXT:NEXT
70 POKE VO,0:POKE WF,0

```

```

10 REM *** BUZZER ***
20 VO=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 POKE VO,15:POKE AD,18:POKE SR,132:POK
E WF,33
40 FOR TT=1 TO 8
50 POKE HF,4:POKE LF,73:FOR TD=1 TO 50:N
EXT
60 POKE HF,7:POKE LF,53:FOR TD=1 TO 50:N
EXT
70 NEXT
80 POKE VO,0:POKE WF,0

```

```

10 REM *** FAILURE ***
20 VO=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 POKE VO,15:POKE AD,32:POKE SR,255:POK
E WF,33
40 POKE HF,2:POKE LF,220
50 FOR TD=1 TO 1500:NEXT
60 POKE VO,0:POKE WF,0

```

```

10 REM *** WARNING SIREN ***
20 VO=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272
30 POKE VO,15:POKE AD,18:POKE SR,146:POK
E WF,33
40 FOR TG=1 TO 6
50 POKE HF,14:POKE LF,107:FOR TD=1 TO 35
0:NEXT:POKE HF,0:POKE LF,0
60 FOR TD=1 TO 125:NEXT
70 POKE HF,9:POKE LF,159:FOR TD=1 TO 350
:NEXT:POKE HF,0:POKE LF,0
80 FOR TD=1 TO 125:NEXT
90 NEXT

```

Symphonies

From sound effects, we move onto tunes. By using DATA statements it is a very simple matter to play any piece of music on your computer. The example program shows the basic method to use. The program contains a central loop from line 40 to line 100. This loop READs the DATA, POKEs the values into the high frequency and low frequency registers and provides the time delay to create the length of the note. Line 50 is used to turn off the sound registers when the song has finished. This line gets its signal from dummy pieces of DATA at the end of the DATA lines.

Lines 20 and 30 initialise the sound generator. VO is the volume, AD the attack/decay, SR the sustain/release, WF the waveform, HF the high frequency, LF the low frequency, HP the high pulse rate for waveform 65, the pulse waveform, and LP is the low pulse rate.

The DATA in lines 200 to 260 is used in groups of three. The first number is for the high frequency register (READ H) and the second number is for the low frequency (READ L). The third piece of DATA is for the time delay (READ D), which is used in the FOR/NEXT loop in line 70.

Lines 500 to 530 set the border and screen colours and PRINT the title of the piece of music while it is playing.

To produce this short program, I selected a suitable piece of music and then set to work. Reading from the sheet music, I obtained the values for each note from

the 'Music Note Values' list in the User's Manual. The **Programmer's Reference Guide** will give you the exact values for the length of the notes. When you are doing this, don't forget the sharps and flats or else you may find some very odd-sounding notes in your tune.

A final warning. Most popular songs are covered by copyright, so if you are using a piece of music in a program that will be used other than for your own use at home, make sure you choose music that is not covered by copyright.

```
10 REM *** PLAYING A TUNE ***
15 GOSUB 500
20 VO=54296:AD=54277:SR=54278:WF=54276:H
F=54273:LF=54272:HP=54275:LP=54274
30 POKE VO,15:POKE AD,65:POKE SR,65:POKE
WF,65:POKE HP,200:POKE LP,5
40 READ H:READ L:READ D
50 IF H=999 THEN POKE VO,0:POKE WF,0:END
60 POKE HF,H:POKE LF,L
70 FOR TD=1 TO D:NEXT
80 POKE HF,0:POKE LF,0
100 GOTO 40
200 DATA 28,49,412,28,49,412,29,233,412,
33,135,412,33,135,412,29,233,412
210 DATA 28,49,412,25,30,412,22,96,412,2
2,96,412,25,30,412,28,49,412
220 DATA 28,49,600,25,30,200,25,30,850
230 DATA 28,49,412,28,49,412,29,233,412,
33,135,412,33,135,412,29,233,412
240 DATA 28,49,412,25,30,412,22,96,412,2
2,96,412,25,30,412,28,49,412
250 DATA 25,30,600,22,96,200,22,96,412
260 DATA 999,999,999
```



```

500 POKE 53280,0:POKE 53281,2
510 PRINT "S"
520 PRINT TAB(14) "QQQQQQQQ ODE TO JOY"
525 REM CRSR DOWN PURPLE
530 RETURN

```

You may now be saying: The tune is very nice, but what use is it to me in writing an adventure? A tune can be useful. It can be a theme tune played 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.

This is just one of the many things that you can do with sound in an adventure. Generally sound is a good feature to add. It is relatively cheap on memory and if an adventurer does not like it then he can turn his television down!

Chapter Eight

Help! A Guide to the Adventures in this Book





CHAPTER EIGHT

You need only read this chapter if you have failed to solve the adventures in this book. We deliberately withheld as much information about the adventures as 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, there are still many little traps for you to fall into. Here, we will give you a few tips to get you further into the adventures in this book.

Golden Chalice

The first adventure, that we 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:

Roof opening	=	CHR\$ 32	=	space
Monster	=	CHR\$ 133	=	solid circle
Human	=	CHR\$ 120	=	clubs symbol
Rope	=	CHR\$ 41	=	right bracket
Torch	=	CHR\$ 122	=	diamond
Flood	=	CHR\$ 166	=	checkerboard
Quicksand	=	CHR\$ 119	=	open circle
Map	=	CHR\$ 188	=	cross
Chalice	=	CHR\$ 36	=	asterisk
Empty Cave	=	CHR\$ 46	=	full stop

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 whole map at once. 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 we can give, other than to say: watch the screen at all times.

Dark Forest

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

Here are some tips:

- ★ Do not go North from the beginning; you will land in prison.
- ★ You have a better than average chance of benefiting from the opening of a possession, in all but extreme circumstances.
- ★ 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.
- ★ When fighting, choose your strongest attribute. If you lost, the game is over. Note: that whatever battle level of the Creature, the magic level is divided by 10. Magic is often the best attribute to use.

- ★ 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).
- ★ 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.
- ★ The cave's entrance is to the south; the clue you are given is just to slow you down considering the answer.
- ★ 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, 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.
- ★ 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!

- ★ 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 enemies and are particularly effective against the Creature.
- ★ 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 and actions and try to construct a map. These will all aid you in trying to solve the adventure.

The Nielson Papers

This is a much tougher adventure than the previous one, and will keep you adventuring for some time. Here are some hints to aid you in your progress. Be warned, we have not told you everything.

- ★ From the opening location, you can only move north.
- ★ 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).
- ★ There are 30 or so commands for you to find, understand and use. Here is a list of them:

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

TAKE
GET
STEAL — Collect a movable object.
DROP — Self explanatory, allows you to discard an object.
READ — Read certain messages.
CLIMB — Climb up to shelf with papers on, 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 executive washroom.
OPEN
UNLOCK — If carrying key, unlocks doors but must be used discriminatingly, very old.
FIRE
SHOOT — If carrying gun, allows you to fire at 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, then the normal description. Must have torch to do so.
WEAR
COMPUTE
CALCULATE — Allows you to work out the coded memo to the computer-controlled lock in the document room.

DECODE

- 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 will allow you to vanquish a security guard. If you do not have rope or wirecutters, then you will fight with your hands.

- ★ There are 18 objects and only a couple like paper-clips and Papers are totally useless and have been put in as 'red herrings'.
- ★ The key which is in the dark corridor opens all the locked doors.
- ★ 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.
- ★ Once in the document room, you will need a stool or ladder to climb to the top shelf.
- ★ Try to decode the memo as its advice is pretty important.
- ★ Once in the maintenance area, going west will just take you further into it.
- ★ Be careful when around the vault. If you get locked in there, without the key, your game is over.

- ★ Plan your escape carefully. Once you have taken the papers, the alarms sound and your remaining time is very short.
- ★ Finally, as with 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.

City Of Atlantis

A lot of help 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 are a few extra points of note:

- ★ If you bump into a wall in room 1, the Atlantan is distracted with laughter and you can then go East to room 2.
- ★ In the Radio Station, do not stay for too long, some Atlantans may hear you over the radio and come and investigate.
- ★ 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.
- ★ Try to steer 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.
- ★ A cryptic clue to finish with, remember your horticultural knowledge, it will come in useful at a point . .

CHAPTER NINE

Chapter Nine

Some Ideas for your own Adventures



Chapter Nine

Some ideas for your own Adventures



CHAPTER NINE

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

The Plot

Obvious ones include stealing the treasure, battling against the creature to save planet Earth and rescuing a prisoner. 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 its 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, e.g. space, medieval, underground, everyday, desert, ocean, recent history and the like. The present time is a fascinating area in which to place your adventure. Most people expect a 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! Our 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 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 idea 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 story gives you many of the locations and characters needed.

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 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 can use. They are fairly non-specific and are short so there is room for you to add details if you need 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.

The Characters

Names of creatures both good and bad are not that hard to create, but why work yourself when you have a powerful computer nearby? The program below generates random creature names. The combinations of letters are not totally random, 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 PRINT "s":NAME$="":A=0:L$=""
30 PRINT:FOR N=1 TO INT(2*RND(1)+2)
40 A=INT(26*RND(1)+65)
50 IF A=81 AND N<>1 THEN 40
55 GOSUB 130
60 NAME$=NAME$+CHR$(A)+L$
70 NEXT N
100 PRINT TAB(10) NAME$:PRINT:PRINT:PRINT
110 PRINT "PRESS ANY KEY FOR ANOTHER NAME"
```



```

120 GET A$:IF A$="" THEN 120
125 RUN
130 RESTORE:FOR T=1 TO INT(8*RND(1)+1):R
EAD L$
140 NEXT T
150 RETURN
160 DATA A,E,I,O,U,AE,CH,TH

```

For those of you with a printer, why not type in the following program that dumps twenty names onto paper.

```

10 REM *** NAME GENERATOR ***
15 OPEN 1,4:CMD1
16 REM PRINTER PLOTTER SHOULD BE OPEN 1,
6
17 FOR X=1 TO 20
20 NAME$="":A=0:L$=""
30 PRINT:FOR N=1 TO INT(2*RND(1)+2)
40 A=INT(26*RND(1)+65)
50 IF A=81 AND N<>1 THEN 40
55 GOSUB 130
60 NAME$=NAME$+CHR$(A)+L$
70 NEXT N
100 PRINT NAME$:
110 NEXT X:CLOSE 1:END
130 RESTORE:FOR T=1 TO INT(8*RND(1)+1):R
EAD L$
140 NEXT T
150 RETURN
160 DATA A,E,I,O,U,AE,CH,TH

```

A large percentage of the names formulated will be unusable, but you only need to gain a few names to have made the exercise worthwhile. Here is a sample printout of some words generated. We have ringed those which may be suitable. You may have different ideas, and prefer others in the list.

ZUVAE	LUMAE	SAEJAEYI
KEFE	KEYIFE	OAEI
UCHJAENI	HUAETE	WARA
PAEATHUU	BOXCH	XCHCAUCH
MAEIOIA	OCHUUUTH	NUSUKAE
DUIEHAE	CCHTA	DAZU
LCHWU	UTHCEEA	SASI
RCHHE	FOXOBCH	CAPAECYCH
FINO	ZUZUKI	ATHHO
VAETO	VAACH	SUTTH
PIJTH	LUKTH	MECOPI
CCHUOKTH	BETE	OAELEYI
UICCHDTH	VAYOJO	QTHPI
WALTH	AUME	WEBAFCH
WUOEVAE	YFAI	IEEA
SCHUI	ZEJAEST	ZCHAUGH
HUDETA	SEKAEXTH	BTHZAE
KTHCOTTH	MOXTHMAE	SANU
UAESOPU	BAIT	KOWCH
IIOA	JOMCH	CCHCAOI

Here, at last, is the key to creating and playing adventure programs on your Commodore 64. Written by two highly-experienced authors and adventurers, this book will take you further into the fabulous world of computer adventuring than you had dared hope.

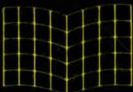
You will be shown how to write program lines which allow your Commodore 64 to understand words like "GO", "PICK UP", "UNLOCK", "GET" and all the rest which are needed so that the computer will follow player's commands.

Much more is given relating to the **mechanics** of writing adventure games. But, as well as this, you'll find a host of information to help you solve and enjoy commercial adventures.

And, best of all, there are four **complete** adventure programs, just begging to be typed into your Commodore 64 and solved.

Time to buckle on your armour, and enter the fascinating world of Adventure gaming.

Another great book from

INTERFACE
PUBLICATIONS 

£5.25

ISBN 0-907563-49-X



9 780907 563495