

**Personal  
Computer**  
World

**CENTURY**  
SOFTWARE

PCW

# GAMES COLLECTION

FOR THE

**BBC MICRO**

**DIGITALLY REMASTERED EDITION**

---

PCW

---

**GAMES  
COLLECTION**

FOR THE  
**BBC MICRO**

---

PCW

---

# **GAMES COLLECTION**

FOR THE  
**BBC MICRO**



Century Communications  
London

Copyright ©Editors of PCW and James Byrne 1984  
All rights reserved  
First published in Great Britain in 1984 by  
Century Communications Ltd,  
12-13 Greek Street,  
London W1V 5LE

ISBN 0 7126 0621 1

DIGITALLY REMASTERED ON RISC OS COMPUTERS, MARCH 2012.

# CONTENTS

Introduction	1
Typing Tips	2
Robotank	4
Splash!	13
Golden Fleece	23
Pursuit Ship	39
Martian Attack	50
Hunchback	73
Hunger Alley	82
Dogfight	93
Euler's Towing Knight	103
Missile Defender	108
Black Holes	116
Omniopoly	123
Map Quiz	143
Aircraft Lander	150
Micromouse	161
Trank	167
Four in a Line	174
City Smasher	182
Baby Monty	187
Flappa Catcha	195

# INTRODUCTION

Each year *Personal Computer World*, Britain's biggest microcomputer magazine, receives literally thousands of computer programs from its dedicated band of readers. Now, for the first time, we have selected the very best of their collective efforts and proudly offer you this balanced compilation of twenty games. Within these pages you are sure to find something of pleasure, whether your taste is for arcade-action or board games, adventures or brain-teasers.

Each program has been redesigned by a BBC Micro expert to make full use of colour, sound and the other facilities of your machine. All programs have been closely vetted by one of *PCW*'s experienced team of referees to ensure that they are as bug-free as possible.

Full documentation is included so that you can gain a clear understanding of how each program works. You will be able to improve your own programming style, and many of the routines can also be used within your own programs. Who knows, if you spend as many hours in front of your computer as these programmers have, you may get one of your programs published next year!

# TYPING TIPS

If you are a new user of micros, before you start typing in any programs it is important to read through your user manual so that you are familiar with the keyboard of your machine. Most of all, it is vital that you type in the program exactly as it appears in the book. To help you avoid the commonest mistakes, here are some hints.

- 1 Think before you type. Try to understand as much of what the program is trying to do as you can.
- 2 Remember that every character is important to your computer and you should type in exactly what is listed in the book including all spaces. It is not, however, necessary to have any spaces between the line number and the beginning of the text
- 3 Be particularly careful that you distinguish between the following:
  - 1 (one) and l (letter)
  - 0 (zero) and O (letter). All zeros are Ø in the listings
  - : (colon) and ; (semi-colon)
  - , (comma) and . (full stop)
  - " (quote marks) and ' (apostrophe)
  - | and : (colon)quotation-marks with/without a space between them.
- 4 When lower case letters are listed they must be entered as such, not as capitals.
- 5 Always save your program on a tape before running it for the first time. If you have made any typing errors your program may crash and hours of patient work be lost before you can correct it.
- 6 You may come across the | symbol in the listings. This is, in fact, the | character found on the BBC Micro next to the cursor keys.
- 7 If you have a disk system, reset the memory by typing:  
PAGE=&E00  
before typing in any of these programs.

**Note:**

If you don't want the bother of typing in the listings in this book you can get the programs on cassette instead. Simply send away the voucher at the back of the book, with a cheque or postal order for £4.95.

The programs on the cassette are exactly the same as those printed here, so you can still use all the print-outs and program descriptions to help you play the games, and read the listings to understand how the programs were built up.



# ROBOTANK

By Bryn Phillips

Robotank is an original strategy game for the BBC Micro, The idea is to drive your tank wound a battlefield, strewn with a number of targets. You have to destroy as many of them as possible in five missions.

On starting the game the battlefield will be drawn, containing a number of targets including enemy tanks, huts and guns. Your Robotank will be placed at the bottom left of the screen.

The idea of the game is to program the tank with a series of Logo style commands in order to destroy as many targets as possible. The available commands we:

- F - Forward one length
- R - Turn right 90 degrees
- L - Turn left 90 degrees
- S - Shoot
- P - Program complete - GO
- D - Delete last instruction

For example, the following simple program: FFERSLFFS, would make the tank move forward three lengths, turn right 90 degrees, shoot, turn left 90 degrees, move two lengths forward and shoot again.

The maximum Robotank program is 40 commands, and once the tank has been started it cannot be stopped. If the tank crashes into any obstacles, the game will end; otherwise, you have five runs in which to destroy all the targets. The points value of the targets increases with each hit, so careful planning is required to achieve high scores.

## Program Description

**110** Changes mode and turn off cursor.

**130** Call procedure to define characters and envelopes.

**140** Call procedure to print instructions.

**150** Dimension array for commands and set high score to zero.  
**150** Clear screen to a yellow background.  
**190** Initialise variables.  
**200** Print the Robotank on the screen.  
**210-250** Print three guns on the screen, checking that they do not print over something else.  
**250-300** Do the same for five enemy tanks.  
**310-350** Do the same for 20 hills.  
**360-400** Do the same for ten huts.  
**410-420** Print a green border wound the battlefield.  
**430** Set variables for the tank's position.  
**500-550** Input the command string, finishing when P is pressed.  
**580** Set variables and call procedure to play a fanfare.  
**600** Make short delay and 'tank moving' sound.  
**610-670** Decode commands and act on them.  
**680** Blank out tank at old position.  
**690** Test the position on which the tank will land.  
**700** Print tank at new position.  
**710** If the tank has hit an object blow it up and play 'death march'.  
**730** Colour in the command that has just been decoded.  
**750** Test for end of mission or game.  
**760** Add one to number of missions.  
**770** Test for end of game  
**790** Play 'game over' tune.  
**800** Test to see if another game is wanted.  
**820** See if shot has hit the edge of the screen  
**830** Test the shot's new position.  
**840** If it has hit a hill delete shot  
**850** Print shot on screen.  
**860** Make a short delay,  
**870** Remove shot from screen.  
**880** If the shot has hit something blow it up.  
**920** Make 'shot fired' noise  
**930-960** Fire the shot in the direction that the tank is facing.  
**980-1010** Ask if another game is wanted and get reply.  
**1030-1050** Delay for a given time.

**1070-1100** Print score, high score and mission number.  
**1130** Add one to number of hits.  
**1140** Make 'explosion' noise.  
**1170** increase score according to what you hit and how many hits you have made.  
**1190** Flash explosion at the given position.  
**1120-1140** Find the ASCII code of a character at a given position  
**1260-1380** Define characters and envelopes.  
**1400-1630** Print instructions.  
**1650-1150** Play a given tune.  
**1110** Data for 'fanfare'.  
**1180-1190** Data for 'death march'.  
**1800-1820** Data for 'game over' tune.

---

```

10  REM  * * * * *
20  REM  *
30  REM  *  ---==ROBOTANK==--  *
40  REM  *
50  REM  *  BBC  CONVERSION  BY  *
60  REM  *
70  REM  *      JAMES  BYRNE      *
80  REM  *
90  REM  * * * * *
100 REM  =====
110 MODE 1:VDU 23;10,32;0;0;0;
120 VDU 19,3,3;0;19,2,2;0;
130 PROCchars
140 PROCinstructions
150 DIM P$(42):HG=0
160 REM  =====
170 REPEAT
180   COLOUR 131:CLS
190   C=224:SC=0:TM=0:MS=1
200   COLOUR 1:PRINT TAB(1,21)CHR$(C)
210   FOR I=1 TO 3
220     REPEAT:X=RND(38):Y=RND(22)
230     UNTIL FNpeek(X,Y)=32
240     COLOUR 1:PRINT TAB(X,Y)CHR$(2
28)
250     NEXT

```

```

260     FOR I=1 TO 5
270         REPEAT:X=RND(38):Y=RND(22)
280         UNTIL FNpeek(X,Y)=32
290         COLOUR 0:PRINT TAB(X,Y)CHR$(2
29)
300     NEXT
310     FOR I=1 TO 20
320         REPEAT:X=RND(37)+1:Y=RND(21)+
1
330         UNTIL FNpeek(X,Y)=32
340         COLOUR 2:PRINT TAB(X,Y)CHR$(2
30)
350     NEXT
360     FOR I=1 TO 10
370         REPEAT:X=RND(38):Y=RND(22)
380         UNTIL FNpeek(X,Y)=32
390         COLOUR 0:PRINT TAB(X,Y)CHR$(2
31)
400     NEXT
410     COLOUR 130:PRINT TAB(0,0)SPC(40
);TAB(0,23)SPC(40)
420     FOR I=1 TO 22:PRINT TAB(0,I) " "
;TAB(39,I) " ":NEXT:COLOUR 131
430     X=1:Y=21:X1=X:Y1=Y
440     REM =====
450     REPEAT
460         PROCscore
470         COLOUR 0:COLOUR 130:PRINT TAB
(10,25)"ENTER YOUR COMMANDS"
480         COLOUR 129:PRINT TAB(0,27)SPC
(40)
490         I=1:REPEAT
500             *FX 15,1
510             REPEAT:A$=GET$:SOUND 1,-10,
50,2
520                 UNTIL A$="F" OR A$="S" OR
A$="L" OR A$="R" OR A$="D" OR A$="P"
530                 IF A$="D" AND I>1 I=I-1:PRI
NT TAB(I-1,27) " ":UNTIL FALSE ELSE IF A$
="D" AND I=1 UNTIL FALSE
540                 IF I<41 AND A$<>"P" P$(I)=A$:PR
INT TAB(I-1,27)P$(I):I=I+1
550                 UNTIL A$="P"
560         COLOUR 131:PRINT TAB(10,25)SPC(20
):SOUND &11,0,0,0

```

```

570 REM =====
580 TM=0:J=1:COLOUR 1:PROctune(1)
590 REPEAT
600     PROCdelay(20):SOUND &10,-15,6,4
610     IF P$(J)="S" THEN PROCshoot
620     IF P$(J)="L" THEN C=C+1:IF C=22
8 C=224
630     IF P$(J)="R" THEN C=C-1:IF C=22
3 C=227
640     IF P$(J)="F" AND C=224 THEN X=X
+1:IF X>38 X=38
650     IF P$(J)="F" AND C=225 THEN Y=Y
-1:IF Y<1 THEN Y=1
660     IF P$(J)="F" AND C=226 THEN X=X
-1:IF X<1 THEN X=1
670     IF P$(J)="F" AND C=227 THEN Y=Y
+1:IF Y>22 THEN Y=22
680     PRINT TAB(X1,Y1) " "
690     PK=FNpeek(X,Y)
700     PRINT TAB(X,Y)CHR$(C)
710     IF PK<>32 PROCexplode(X,Y,0):PR
INT TAB(10,6)"You crashed your tank":PRO
Cdelay(20):PROctune(2)
720     X1=X:Y1=Y
730     COLOUR 3:COLOUR 129:PRINT TAB(J
-1,27)P$(J)
740     PROCscore
750     COLOUR 1:COLOUR 131:J=J+1:UNTIL
J>I-1 OR PK<>32
760     MS=MS+1
770     UNTIL MS>5 OR PK<>32
780     COLOUR 0:PRINTTAB(16,4)"GAME OVER
"
790     IF MS>5 AND PK=32 PROctune(3)
800     UNTIL FNagain=FALSE:CLS:END
810     REM =====
820     DEF PROCshot:IF X2<1 OR X2>38 OR
Y2>22 OR Y2<1 ENDPROC
830     PK=FNpeek(X2,Y2):Z=0
840     IF PK=134 Z=1:ENDPROC
850     COLOUR 0:PRINTTAB(X2,Y2)".":COLOU
R 1
860     PROCdelay(10)
870     PRINTTAB(X2,Y2) " "
880     IF PK<>32 THEN PROCexplode(X2,Y2,

```

```

PK)
  890   ENDPROC
  900   REM =====
  910   DEF PROCshoot
  920   SOUND 0,-15,5,5
  930   IF C=224 THEN X2=X+1:REPEAT:Y2=Y:
PROCshot:X2=X2+1:UNTIL Z=1 OR X2=X+6:END
PROC
  940   IF C=225 THEN Y2=Y-1:REPEAT:X2=X:
PROCshot:Y2=Y2-1:UNTIL Z=1 OR Y2=Y-6:END
PROC
  950   IF C=226 THEN X2=X-1:REPEAT:Y2=Y:
PROCshot:X2=X2-1:UNTIL Z=1 OR X2=X-6:END
PROC
  960   Y2=Y+1:REPEAT:X2=X:PROCshot:Y2=Y2
+1:UNTIL Z=1 OR Y2=Y+6:ENDPROC
  970   REM =====
  980   DEF FNagain
  990   PRINTTAB(8,8)"Do you want another
game ? "
  1000  REPEAT:A$=GET$:UNTIL A$="Y" OR A$
="N"
  1010  IF A$="Y" =TRUE ELSE =FALSE
  1020  REM =====
  1030  DEF PROCdelay(T%)
  1040  TIME=0:REPEAT UNTIL TIME>T%
  1050  ENDPROC
  1060  REM =====
  1070  DEF PROCscore
  1080  IF SC>HG HG=SC
  1090  COLOUR 3:COLOUR 130:PRINT TAB(1,0
)"HIGH ";HG;TAB(15)"MISSION ";MS;TAB(29)
"SCORE ";SC
  1100  ENDPROC
  1110  REM =====
  1120  DEF PROCexplode(X2,Y2,PK)
  1130  TM=TM+1
  1140  SOUND &10,-15,6,20
  1150  IF PK=132 THEN SC=SC+25*TM
  1160  IF PK=133 THEN SC=SC+15*TM
  1170  IF PK=135 THEN SC=SC+5*TM
  1180  Z=1
  1190  FOR N=1 TO 10:PRINTTAB(X2,Y2)CHR$
(232):PRINT TAB(X2,Y2)" ":NEXT
  1200  ENDPROC

```

```

1210 REM =====
1220 DEF FNpeek(X,Y)
1230 VDU 31,X,Y:A%=135
1240 =( (USR(&FFF4)AND &FFFF)DIV &100)
1250 REM =====
1260 DEF PROCchars
1270 VDU 23,224,0,0,120,252,255,252,12
0,0
1280 VDU 23,225,16,16,56,124,124,124,1
24,56
1290 VDU 23,226,0,0,30,63,255,63,30,0
1300 VDU 23,227,56,124,124,124,124,56,
16,16
1310 VDU 23,228,0,2,4,72,48,240,232,22
4
1320 VDU 23,229,0,0,0,31,24,60,255,255
1330 VDU 23,230,0,0,0,16,60,126,127,25
5
1340 VDU 23,231,0,16,57,125,255,147,14
7,243
1350 VDU 23,232,0,72,130,24,57,4,64,8
1360 ENDPROC
1370 REM =====
1380 DEF PROCinstructions
1390 COLOUR 131:COLOUR 1:CLS
1400 PRINT TAB(10,1)"---==ROBOTANK==
---"
1410 PRINT'" The idea of this game is
to program the tank with a series of
logo-style commands in order to destr
oy as many targets as possible."
1420 PRINT'" At the start, the tank is
positioned at the bottom left corner
of the screen and it can be programmed w
ith the following commands:--"
1430 PRINT'" F = Move forward one leng
th"
1440 PRINT" L = Turn left 90 degrees"
1450 PRINT" R = Turn right 90 degrees"
1460 PRINT" S = Shoot"
1470 PRINT" P = Program complete - GO"
1480 PRINT" D = Delete the previous co
mmand"
1490 PRINT'" The maximum robot tank pr
ogram is 40 commands. If the tank cras

```

```

hes into any obstacle, the game will end, otherwise you will have five runs in which to destroy all the targets."
1500 PRINT" The score of each target increases with each mission, so careful planning is required to achieve high scores."
1510 PRINTTAB(7,30)"Press space to continue"
1520 REPEAT:UNTIL GET=32
1530 CLS:PRINT TAB(10,1)"----==ROBOTANK====-"
1540 PRINT'TAB(10)CHR$(224);" - Your tank"
1550 PRINT'TAB(10)CHR$(228);" - Artillery gun"
1560 COLOUR 0:PRINT'TAB(10)CHR$(229);" - Enemy tank"
1570 PRINT'TAB(10)CHR$(231);" - Hut"
1580 COLOUR 2:PRINT'TAB(10)CHR$(230);" - Hill"
1590 COLOUR 0:PRINT'TAB(9)"Press space to start"
1600 REPEAT:UNTIL GET=32
1610 ENDPROC
1620 REM =====
1630 DEF PROCtune(T%)
1640 IF T%=1 RESTORE 1750
1650 IF T%=2 RESTORE 1760
1660 IF T%=3 RESTORE 1780
1670 READ N%
1680 FOR L%=1 TO N%
1690 READ P%,D%
1700 SOUND 1,-15,P%,D%:SOUND 1,0,0,0
1710 NEXT
1720 REPEAT UNTIL ADVAL(-6)=15
1730 ENDPROC
1740 REM =====
1750 DATA 6,13,2,25,2,45,2,65,3,45,2,65,10
1760 DATA 11,25,12,25,8,25,4,25,12,37,8,33,4
1770 DATA 33,8,25,4,25,8,21,4,25,12
1780 DATA 15,81,4,81,4,81,8,69,4,69,4,69,8

```



1790 DATA 53,4,61,4,53,4,33,16  
1800 DATA 41,4,41,8,49,4,49,8,53,8

# SPLASH

By Jeff Aughton

This a new and different boardgame for two players: the computer and you. The players take turns to 'drop' their counters onto their chosen square in a 7 by 7 grid, but the 'splash' of colour onto surrounding squares may give your opponent the advantage. The effects are predictable and the computer can keep track of what will happen, but can you?

Full instructions are contained within the program. Lines 2000-2330 contain an assembly language routine and you should be especially careful to type these lines exactly as they are shown.

## Program Description

**20-90** Initialize, display rules, print board.

**100-190** Main loop for the game.

**200-1950** Many short procedures that form the basis of the program. The label given to each procedure fully explains its function and is an excellent example of good structured programming. Note especially the humour of labelling the procedure which performs a time delay PROCrastinate!

**1970-2350** PROCass - assembles machine code routines and loads them into memory. Machine code is used to speed up the 'thinking' process for the computer.

**2370-2810** PROCrules - prints rules and instructions for game.

**2830-2890** PROCyorn(A\$) - checks for Y key or N key being pressed.

**2900-3060** Data.

---

```
10 REM SPLASH - A GAME BY J.AUGHTON (
FOR P.K.)
20 MODE7:VDU23;8202;0;0;0;
30 PROCrules
40 PROCinitialise
50
60 REPEAT
```

```

70     MODE2:VDU23;8202;0;0;0;
80     PROCstart
90     PROCboard
100    REPEAT
110        M%=3-M%
120        COLOUR7:COLOUR128
130        IF M%=1 THEN PROCyou ELSE PROC
me
140            IF Tot%(M%)>24 THEN Win=1
150            UNTIL Win
160        PROCgover
170        PROCyorn("ANOTHER GAME? (Y/N)")
180        UNTIL No
190    END
200
210    DEFPROCinitialise
220    DIM A%(11,11),Tot%(2)
230    DIM Tab1 24,Tab2 24,Board 121
240    Temp=&70:Tot=&71
250    Blank$=CHR$(30)+STRING$(20," ")+CHR$
R$(30)
260    PROCtable(Tab1)
270    FOR I%=1 TO 11:FOR J%=1 TO 11
280        PROCupdate(I%,J%,-1)
290    NEXTJ%,I%
300    PROCass
310    ENDPROC
320
330    DEFPROCstart
340    CLS:RESTORE 2990
350    PROCtable(Tab2)
360    FOR I%=3 TO 9:FOR J%=3 TO 9
370        PROCupdate(I%,J%,0)
380    NEXTJ%,I%
390    First=1:Win=0:Tot%(1)=0:Tot%(2
)=0
400    PROCyorn("Want to start? (Y/N)
")
410    IF Yes THEN M%=2 ELSE M%=1
420    PROCcrastinate(200)
430    ENDPROC
440
450    DEFPROCboard
460    CLS
470    GCOL0,5:COLOUR6

```

```

480      FOR I%=96 TO 992 STEP 128
490          MOVE I%,64:DRAW I%,736
500      NEXT
510      FOR I%=64 TO 736 STEP 96
520          MOVE 96,I%:DRAW 992,I%
530      NEXT
540      PRINTTAB(2,6)"A B C D E F G"
550      FOR I%=1 TO 7
560          PRINTTAB(0,7+3*I%);I%
570      NEXT
580      FOR C%=0 TO 4
590          I%=11:J%=C%+4:PROCshade
600          PROCupdate(I%,J%,-1)
610      NEXT
620      COLOUR0:COLOUR135
630      PRINTTAB(0,31)"Score: You=0   M
e=0";
640      ENDPROC
650
660      DEFPROCyou
670      PRINTBlank$;"Letter? ";
680      REPEAT
690          X%=GET-62
700          UNTIL (X%>2 AND X%<10) OR X%
=19
710      IF X%=19 THEN PROCquit ELSE PR
OCcont
720      ENDPROC
730
740      DEFPROCcont
750      VDU X%+62
760      PRINTTAB(11,0)"Number? ";
770      REPEAT
780          Y%=GET-46
790          UNTIL Y%>2 AND Y%<10
800      VDU Y%+46
810      IF A%(X%,Y%)>=0 THEN PROCplay(
"Y") ELSE M%=2
820      ENDPROC
830
840      DEFPROCquit
850      PROCyorn(Blank$+"Want to quit?
(Y/N) ")
860      IF Yes THEN Win=2 ELSE M%=2
870      ENDPROC

```

```

880
890      DEFPROCme
900      PRINTBlank$;"It's my move..."
910      PROCcrastinate(150)
920      IF First THEN PROCguess ELSE P
ROCTthink
930      PRINTBlank$;"I will move to ";
940      VDU X%+62,32,Y%+46
950      PROCcrastinate(200)
960      PROCplay("M")
970      ENDPROC
980
990      DEFPROCguess
1000     First=0
1010     REPEAT
1020         X%=RND(9):Y%=RND(9)
1030         UNTIL A%(X%,Y%)>=0
1040     ENDPROC
1050
1060     DEFPROCthink
1070     IF Tot%(1)>20 THEN RESTORE 304
0:PROCTable(Tab2)
1080     IF Tot%(2)>22 THEN RESTORE 299
0:PROCTable(Tab2)
1090     Max%=0
1100     FOR X%=3 TO 9:FOR Y%=3 TO 9
1110         IF A%(X%,Y%)>=0 THEN PROCc
hoose
1120         NEXT Y%,X%
1130         X%=X1%:Y%=Y1%
1140     ENDPROC
1150
1160     DEFPROCchoose
1170     ?(Temp)=11*Y%-11+X%
1180     CALL BEST
1190     IF ?(Tot)>Max% THEN Max%=?(T
ot):X1%=X%:Y1%=Y%
1200     ENDPROC
1210
1220     DEFPROCplay(A$)
1230     PROCsplash(0,200,A$)
1240     PROCsplash(1,100,A$)
1250     PROCsplash(2,100,A$)
1260     ENDPROC
1270

```

```

1280      DEFPROC splash (D%, W%, A$)
1290      FOR I%=X%-D% TO X%+D%
1300          FOR J%=Y%-D% TO Y%+D%
1310              C%=A% (I%, J%)
1315              SOUND 17+D%, -9, 4*RND(20)
+40*D%, 8-2*D%
1320              IF C%>=0 THEN PROCshade
1330              IF C%=4 AND D%=2 THEN PR
OCTake (A$)
1340                  NEXT J%, I%
1350          PROCrastinate (W%)
1360      ENDPROC
1370
1380      DEFPROCshade
1390      IF C%=7 THEN K%=1 ELSE K%=
(C%+1) MOD 8-2*(C%=4)
1400      PROCupdate (I%, J%, K%)
1410      GCOL0, A% (I%, J%)
1420      MOVE128*I%-280, 932-96*J%
1430      MOVE128*I%-168, 932-96*J%
1440      PLOT81, -112, 88:PLOT81, 112,
0
1450      ENDPROC
1460
1470      DEFPROCtake (A$)
1480      COLOUR0:COLOUR135
1490      PRINTTAB(2*I%-4, 3*J%+1); A$
1500      PROCupdate (I%, J%, -1)
1510      Tot% (M%) = Tot% (M%) + 1
1520      PRINTTAB(11, 31); Tot% (1);
1530      PRINTTAB(17, 31); Tot% (2);
1540      ENDPROC
1550
1560      DEFPROCupdate (X%, Y%, K%)
1570      A% (X%, Y%) = K%
1580      ? (Board+11*Y%-11+X%) = K%
1590      ENDPROC
1600
1610      DEFPROCtable (X%)
1620      FOR I%=0 TO 23
1630          READ Y%
1640          ? (X%+I%) = Y%
1650      NEXT
1660      ENDPROC
1670

```

```

1680         DEFPROCcrastinate(W%)
1690         T%=TIME
1700         REPEAT UNTIL TIME>T%+W%
1710         ENDPROC
1720
1730         DEFPROCgover
1740         PROCcrastinate(200)
1750         COLOUR7:COLOUR128:CLS
1760         IF Win=1 THEN PROCcomment
ELSE PROCabandon
1770         ENDPROC
1780
1790         DEFPROCcomment
1800         D%=ABS(Tot%(1)-Tot%(2))
1810         A$="That was a near one!"
1820         IF D%>2 THEN A$="WOW!- a c
lose game!"
1830         IF D%>4 THEN A$="What a go
od game!!"
1840         IF D%>6 THEN A$="An easy w
in!!"
1850         IF D%>8 THEN A$="A MASSACR
E!!!!!!!!!!"
1860         PRINT'A$
1870         IF Tot%(1)>Tot%(2) THEN A$
="YOU WIN" ELSE A$="I WIN"
1880         IF Tot%(1)=Tot%(2) THEN A$
="IT'S A DRAW"
1890         COLOUR15:PRINT' 'A$:COLOUR7
1900         ENDPROC
1910
1920         DEFPROCabandon
1930         PRINT"GAME ABANDONED"
1940         PRINT'"SCORE:You=";Tot%(1)
; " Me=";Tot%(2)
1950         ENDPROC
1960
1970         DEFPROCass
1980         DIM Z% 60
1990         FOR I%=0 TO 2 STEP 2
2000             P%=Z%
2010             [OPT I%
2020             .BEST     LDX #100
2030                     STX Tot
2040                     LDX #23

```

```

2050             .LOOP1 LDA Tab1,X
2060             DEX
2070             .LOOP2 PHA
2080             CLC
2090             ADC Temp
2100             TAY
2110             LDA Board,Y
2120             BMI AGAIN
2130             CLC
2140             ADC Tab1,X
2150             TAY
2160             LDA Tab2,Y
2170             ADC Tot
2180             STA Tot
2190             .AGAIN PLA
2200             BMI TRISH
2210             EOR #255
2220             ADC #1
2230             BNE LOOP2
2240             .TRISH DEX
2250             BPL LOOP1
2260             LDY Temp
2270             LDA Board,Y
2280             TAY
2290             LDA Tab2,Y
2300             CLC
2310             ADC Tot
2320             STA Tot
2330             RTS:]
2340             NEXT
2350         ENDPROC
2360
2370     DEFPROCrules
2380     REPEAT
2390         CLS:RESTORE
2400         VDU141:PRINTTAB(16)"SPLA
SH!"
2410         VDU141:PRINTTAB(16)"SPLA
SH!"
2420         PRINT'"SPLASH! is a bran
d new board game for"
2430         PRINT"two players. It is
played on a 7 by 7"
2440         PRINT"board and the obje
ct of the game is to"

```



```

2450          PRINT"capture more squar
es than the opponent."
2460          PRINT"Thus, the first pl
ayer to capture 25 or"
2470          PRINT"more squares in th
e winner."
2480          PRINT'"We take it in tur
ns to play in any free"
2490          PRINT"(ie uncaptured) sq
uare - playing in a"
2500          PRINT"square affects tho
se surrounding it."
2510          PRINT'"Initially, all sq
uares are black. As the";
2520          PRINT"game progresses th
ey change colour"
2530          PRINT"according to the s
cheme:"'
2540          REPEAT:READ X%:VDU X%:UN
TIL X%=99
2550          PRINT'"If a square is l
eft white at the end of"
2560          PRINT"a move, it is capt
ured and initialled"
2570          PRINT"(Y=you M=me) by th
e moving player."
2580          PRINT'TAB(7)"PRESS SPACE
TO CONTINUE";
2590          REPEAT:X%=RND(7):UNTIL G
ET$=" "
2600          CLS
2610          PRINT'"'"Playing in a sq
uare increases its colour";
2620          PRINT"by 3 steps (the se
quence of colours is"
2630          PRINT"given by the side
of the board to help"
2640          PRINT"you - I don't need
it). The (8) squares"
2650          PRINT"immediately around
it have their colours";
2660          PRINT"increased by 2 ste
ps and the (16) cells"
2670          PRINT"around those are i
ncreased by 1 step."

```

```

2680             PRINT'"Any white cells l
eft at the end of the"
2690             PRINT"move are captured
and cannot be affected";
2700             PRINT"by further plays."
2710             PRINT'"Being a machine -
albeit a clever one! -";
2720             PRINT"my patience is alm
ost inexhaustible and"
2730             PRINT"if a stalemate sit
uation occurs (which"
2740             PRINT"is very unlikely)
I will not surrender"
2750             PRINT"the game; the onus
is on you to do that."
2760             PRINT"If you wish to qui
t the game at any"
2770             PRINT"time, press Q when
asked 'Letter?'"
2780             PRINT'"      Good luck - y
ou'll need it !!!"
2790             PROCyorn(CHR$(129)+"WANT
TO READ THE RULES AGAIN? (Y/N)")
2800             UNTIL No
2810             ENDPROC
2820
2830             DEFPROCyorn(A$)
2840             *FX 15,1
2850             PRINT'A$
2860             REPEAT:A$=GET$
2870             UNTIL A$="Y" OR A$="N"
2880             Yes=(A$="Y"):No=(A$="N")
2890             ENDPROC
2900
2910             DATA 134,221,32,145,47,32,
134,221,32,146,47,32,134
2920             DATA 221,32,147,47,32,134,
221,32,148,47,32,134,221
2930             DATA 32,151,47,32,134,221,
32,145,47,135,101,116,99
2940
2950             DATA 16,1,8,2,8,9,16,10,16
,11,16,12
2960             DATA 8,13,8,20,8,21,8,22,8
,23,8,24

```

```
2970
2980          REM ** NORMAL WEIGHTS **
2990          DATA -1,-1,5,2,1,0,0,0
3000          DATA 0,-1,-3,0,7,0,0,0
3010          DATA -2,-3,-2,6,2,0,0,0
3020
3030          REM ** WEIGHTS FOR DEFENSI
VE ENDGAME **
3040          DATA -1,-2,7,4,-6,0,0,0
3050          DATA 0,-6,4,1,6,0,0,0
3060          DATA -5,-3,4,7,2,0,0,0
```

# GOLDEN FLEECE

By Simon Bryan

Golden Fleece is an interesting adventure game with the added bonus of exciting graphics at each location. The object of your quest is to find the Golden Fleece, which is at one of the 18 locations. You may type in one or two word commands: the commands that the program understands are GO, DIG, GET, DROP, PAY, TAKE, CUT, ROW, KILL, LIGHT, OPEN and INV (inventory - to give you a list of the objects that you are carrying). Directions may be entered in any of three different ways; for example to go north you can type GO NORTH, NORTH or just N.

The way which the program is written makes it easy to extend. Each verb is dealt with by its own procedure, so to add an extra verb all you have to do is add a line to check for it, at line 535 for example, and then add a procedure to deal with it.

## Program Description

- 50** Change mode.
- 60** Change colour 1 to yellow and turn off the cursor.
- 70-90** Redefine the # \$ & % and ' characters.
- 100-260** Print the Introduction.
- 270** Change colour 1 to flashing.
- 280** Wait for a key to be pressed and clear the screen.
- 300** Change colour 0 (the background) to blue.
- 310** Change colour 1 to white.
- 320** Set up the arrays.
- 330** The start of the first loop.
- 340** Set up the variables for a new game.
- 350** The start of the game loop.
- 360** Draw the picture for the room that you are in.
- 370** See if the game has ended.
- 380** Print a list of the objects in the room.
- 390** See if Gandalf will appear.

**400** Get the command string.  
**410-430** See if a direction has been typed in.  
**440-530** Check the list of verbs and act accordingly.  
**540** See if the command was recognized.  
**550** Return to the main game loop.  
**560-590** See if another game is wanted and act accordingly.  
**610-650** PROCobject - print a list of objects in the room.  
**610-100** PROCmove - deal with a movement  
**120-190** PROCinv - print a list of the objects that the player carrying.  
**810-850** PROCdig - deal with the word "DIG".  
**810-940** PROCget - add an object to the player's inventory.  
**960-1030** PROCdrop - leave an object in the room.  
**1050-1080** PROCdelay - delay for a given time unless a key pressed.  
**1100-1140** PROCcut - deal with the verb "CUT"  
**1160-1210** PROCpay - deal with the verb "PAY"  
**1230-1280** PROCrow - deal with the verb "ROW"  
**1300-1320** PROCKill - deal with the verb "KILL"  
**1340-1310** PROClight - deal with the verb "LIGHT"  
**1390-1440** PROCopen - deal with the verb "OPEN"  
**1450-1520** PROCgandalf - print Gandalfs arrival.  
**1540-1560** PROCprint - print a given string with a short delay between each letter.  
**1580-1640** PROCget-command - input a command string split it into two parts - VERB\$ and NOUN\$.  
**1660-1680** FNinstr - perform the INSTR command and avoid the bug in this command found in BASIC 1  
**1100-1110** PROC setup - set up the variables for a new game  
**1190-1810** PROCpicture - print the picture of a given room  
**1890-1910** Data for the arrays.  
**1930-2450** Data for the rooms.

---

```

10  REM  * * * * *
20  REM  *   The Golden fleece   *
30  REM  *       By Simon Bryan  *
```



```

r two word"):PROCprint(" commands: the c
ommands the program"):PROCprint(" unders
tands are GO, DIG, GET, DROP,"):PROCprin
t(" PAY, TAKE, CUT, ROW, KILL, LIGHT,")
252 PROCprint(" OPEN and INV (INVentory
- to give you"):PROCprint(" a list of
objects you are carrying)."):PRINT:PROC
print(" Directions may be entered"):PROC
print(" in any of three different ways;
for")
253 PROCprint(" example, to go north
you can type"):PROCprint(" GO NORTH, NOR
TH or just N.")
260 PRINT'"PRESS ANY KEY TO PLAY THE
ADVENTURE.";
280 D%=GET:CLS
290 REM *****
300 VDU 19,0,4;0;
310 VDU 19,1,7;0;
320 DIM OBJ(9),OBJ$(9),DIR(6),DIR$(6)
, HOLD(9)
330 REPEAT
340 PROCsetup
350 REPEAT
360 PROCpicture(ROOM)
370 IF ROOM=18 THEN E%=TRUE:GOTO
550
380 PROCobject
390 IF RND(20)=1 THEN PROCgandalf
400 F%=FALSE:PROCget_command
410 FOR L%=1 TO 6
420 IF VERB$="GO" AND NOUN$=DIR
$(L%) OR VERB$=DIR$(L%) OR VERB$=LEFT$(D
IR$(L%),1) PROCmove(L%):F%=TRUE
430 NEXT
440 IF VERB$="DIG" PROCdig:F%=TRU
E
450 IF VERB$="GET" OR VERB$="TAKE
" PROCget:F%=TRUE
460 IF VERB$="DROP" PROCdrop:F%=T
RUE
470 IF VERB$="PAY" PROCpay:F%=TRU
E
480 IF VERB$="CUT" PROCcut:F%=TRU
E

```

```

490      IF VERB$="ROW" PROCrow:F%=TRU
E
500      IF VERB$="KILL" PROCKill:F%=T
RUE
510      IF VERB$="LIGHT" PROCLight:F%
=TRUE
520      IF VERB$="OPEN" PROCopen:F%=T
RUE
530      IF VERB$="INV" PROCinv:F%=TRU
E
540      IF NOT F% PRINT"Pardon?":PROC
delay(100)
550      UNTIL E%
560      PRINT"Do you want another game
?"
570      REPEAT:A$=GET$:UNTIL A$="Y" OR
A$="N"
580      UNTIL A$="N"
590      END
600      REM *****
610      DEFPROCobject
620      F=0:FOR L%=1 TO 9
630      IF OBJ(L%)=ROOM AND F=0 PRINT"Y
ou can see...":PRINT"A ";OBJ$(L%):F=1 EL
SE IF OBJ(L%)=ROOM PRINT"A ";OBJ$(L%)
640      NEXT
650      ENDPROC
660      REM *****
670      DEF PROCmove(L%)
680      IF DIR(L%)=0 PRINT"You cannot go
that way.":PROCdelay(200):ENDPROC
690      ROOM=DIR(L%)
700      ENDPROC
710      REM *****
720      DEF PROCinv
730      PRINT"You are carrying...":G%=0
740      FOR L%=1 TO 9
750      IF HOLD(L%)PRINT"A ";OBJ$(L%):G
%=1
760      NEXT
770      IF G%=0 PRINT"Nothing."
780      PROCdelay(400)
790      ENDPROC
800      REM *****
810      DEFPROCdig

```



```

820 IF ROOM<>12 THEN PRINT'"You cannot dig here!":PROCdelay(200):ENDPROC
830 IF HOLD(7)=0 THEN PRINT'"You cannot dig without a spade!":PROCdelay(200):ENDPROC
840 PRINT'"You start to dig, but the ground collapses beneath you. You fall a long way, but fortunately you land safely."':PROCdelay(500)
850 ROOM=15:ENDPROC
860 REM *****
870 DEFPROCget
880 G%=0:FOR L%=1 TO 9
890 IF FNinstr(OBJ$(L%),NOUN$) G%=L
%
900 NEXT
910 IF G%=0 PRINT"I cannot see that here."':PROCdelay(200):ENDPROC
920 HOLD(G%)=1:OBJ(G%)=0
930 PRINT"O.K...":PROCdelay(100)
940 ENDPROC
950 REM *****
960 DEFPROCdrop
970 G%=0:FOR L%=1 TO 9
980 IF FNinstr(OBJ$(L%),NOUN$) G%=L
%
990 NEXT
1000 IF G%=0 OR HOLD(G%)=0 PRINT"You are not carrying that."':PROCdelay(200):ENDPROC
1010 HOLD(G%)=0:OBJ(G%)=ROOM
1020 PRINT"O.K...":PROCdelay(100)
1030 ENDPROC
1040 REM *****
1050 DEFPROCdelay(T%)
1060 *FX 15,1
1070 D%=INKEY(T%)
1080 ENDPROC
1090 REM *****
1100 DEFPROCcut
1110 IF ROOM<>7 THEN PRINT'"I cannot see anything to cut!":PROCdelay(200):ENDPROC
1120 IF HOLD(4)=0 THEN PRINT'"You have nothing to cut with."':PROCdelay(200):ENDPROC

```

```

DPROC
  1130   CUT%=TRUE
  1140   ENDPROC
  1150   REM *****
  1160   DEFPROCpay
  1170   IF ROOM<>13 THEN PRINT'"There is
nothing to pay here!":PROCdelay(200):END
PROC
  1180   IF HOLD(5)=0 THEN PRINT'"I only a
ccept JEWELS!":PROCdelay(200):ENDPROC
  1190   PRINT'"Unfortunately you did not
see the sign that said BRIDGE UNDER REP
AIR and it collapsed underneath your
feet. After falling 4,000 feet y
ou seem to be dead."
  1200   E%=TRUE
  1210   ENDPROC
  1220   REM *****
  1230   DEFPROCrow
  1240   IF ROOM<>14 THEN PRINT'"There is
no water here!":PROCdelay(200):ENDPROC
  1250   IF HOLD(2)=0 THEN PRINT'"You have
not got a boat":PROCdelay(200):ENDPROC
  1260   PRINT'"Unfortunately there was a
hole in the boat.'"Glug...Glug...Glug
...'"Pity you never learned to swim!"
  1270   E%=TRUE
  1280   ENDPROC
  1290   REM *****
  1300   DEFPROCKill
  1310   PRINT'"I can see no one to kill!"
:PROCdelay(200)
  1320   ENDPROC
  1330   REM *****
  1340   DEFPROClight
  1350   IF HOLD(6)=0 THEN PRINT'"I have n
othing to light!":PROCdelay(200):ENDPROC
  1360   PRINT'"The lamp is already lit.":
PROCdelay(200)
  1370   ENDPROC
  1380   REM *****
  1390   DEFPROCopen
  1400   IF ROOM<>3 THEN PRINT'"I cannot s
ee anything to open.":PROCdelay(200):END
PROC

```

```

1410 IF HOLD(9)=0 THEN PRINT"I have no
thing to open it with.":PROCdelay(200):E
NDPROC
1420 IF O%=1 PRINT"It is already open!
":PROCdelay(200):ENDPROC
1430 OBJ(2)=3:O%=1
1440 ENDPROC
1450 REM *****
1460 DEFPROCgandalf
1470 PRINT'"A puff of smoke ..."'
1480 PRINT"Gandalf arrives and says ..
."
1490 PRINT"OH DEAR I SEEM TO HAVE THE
WRONG"
1500 PRINT"ADVENTURE !."
1510 PRINT"A puff of smoke and he is g
one !"
1520 ENDPROC
1530 REM *****
1540 DEF PROCprint(I$)
1550 PRINT:FOR A=1 TO LEN I$:PRINTMID$
(I$,A,1);:PROCdelay(5):NEXT
1560 ENDPROC
1570 REM *****
1580 DEF PROCget_command
1590 *FX 15,1
1600 INPUT'"=>"A$
1610 S%=FNinstr(A$," ")
1620 VERB$=LEFT$(A$,S%-1)
1630 NOUN$=RIGHT$(A$,LEN(A$)-S%)
1640 ENDPROC
1650 REM *****
1660 DEF FNinstr(X$,Y$)
1670 IF LEN(Y$)>LEN(X$) =0
1680 =INSTR(X$,Y$)
1690 REM *****
1700 DEF PROCsetup
1710 ROOM=1:E%=FALSE:CUT%=FALSE:O%=0:R
ESTORE 1890
1720 FOR L%=1 TO 9:READ I:OBJ(L%)=I:NE
XT
1730 FOR L%=1 TO 9:READ I$:OBJ$(L%)=I$
:NEXT
1740 FOR L%=1 TO 6:READ I$:DIR$(L%)=I$
:NEXT

```

[illegible]











ee land on the other side.  
The exit is East., 0, 0, 9, 0, 0

[illegible]

```

      2360    DATA          &
              %            #
              $            '
              '            '

```

```
2380 DATA You find yourself in a sound
proof room.Exits are South and East.,0,
15,17,0,0,0
```





# PURSUIT SHIP

By John Dean

Pursuit Ship is a traditional arcade style space game where you have to locate the enemy ship and blast it into cosmic dust! To aid you in your task you have a long range radar scan which shows the location of the enemy ship relative to yaw sights. All you have to do is to guide it into your sights and fire.

A point to note when typing in this listing is that memory is very limited and so all non-essential spaces should be removed as they are here

## Program Description

**120** Set high score and change mode.

**130** Print instructions.

**140** Change mode.

**160** Initialize the variables.

**170** If the stars are not already drawn then draw them.

**180** Draw the sights.

**210** Initialize the alien ship.

**230** Move the stars.

**240** If there is any space in the sound buffer play another note.

**250** Move the alien ship.

**260** See if a shot needs to be fired.

**270** Use some fuel.

**280** Speed up the game.

**290** Carry on until the alien ship is blown up or the fuel runs out

**300** Carry on until the fuel runs out

**310** Ask whether mother game is wanted.

**340-380** PROCSTARMOVE - display a different set of stars give the impression of movement

**400-530** PROCMOVE - move the alien ship.

**550-580** PROCSHIPSHOW - print the alien ship on the screen and a dot on the radar.

**600-610** PROCFIRE - draw the alien ship's shot, and reduce

amount of fuel.

**630-660** PROCTESTSHOT - test the space bar and the joystick fire button to see if a shot must be fired.

**680-720** PROCSHOOT - fire a shot and test for a hit.

**740-770** PROCHIT blow up the ship and increase the score fuel.

**790-810** PROCUSEFUEL - use up some fuel.

**830-880** PROCSPEEDUP - speed the game up.

**900-950** PROCKEYBOARD - test the four direction keys.

**910-1020** PROCJOYSTICK - test the joystick directions.

**1040-1060** PROCSCORE - print the player's score.

**1080-1120** PROCFUEL - print the amount of fuel left.

**1140-1250** PROCSTATIC - draw the sights and radar on the screen.

**1270-1300** PROCSHIPMAKE - find a random place for the ship.

**1320-1360** PROCSTARS - draw the stars on the screen, while playing the tune.

**1380-1410** PROCSTARMAKE - find a random position for a star.

**1430-1470** PROCSTARSHOW - print a star on the screen.

**1490-1500** PROCSTARNEXT - calculate the position of the next star.

**1520-1530** PROCBLACKOUT - change colours one to six to black and colours nine to 14 to a given colour.

**1550-1660** PROCAGAIN - print the player's score and the high score and see if another game is wanted.

**1680-1150** PROCINIT - define the characters, the envelope and initialize some variables.

**1110-2230** PROCINTRO - print the instructions.

**2280-3210** PROCJOYINST - print the instructions for using a joystick.

**3330-2410** PROCKEYINST - print the instructions for using keyboard

**2430-2450** PROCSPACE - wait for the space bar to be pressed.

**2470** PROCWAIT - wait for a given time

**2490-2520** PROCMUSIC - play the next note of the tune.

**2540-2560** Data for the tune.

---

```

100
110 ONERRORRUN
120 H%=-1:MODE7:VDU23;10,32;0;0;0;
130 PROCINTRO
140 MODE2
150 REPEAT
160 PROCINIT
170 IFH%=-1PROCSTARS(200,1.1):H%=0
180 PROCSTATIC
190 ONERRORPROCAGAIN:RUN
200 REPEAT
210 PROCSHIPMAKE
220 REPEAT
230 PROCSTARMOVE
240 IFADVAL(-8)>3PROCMUSIC
250 PROCSMOVE
260 PROCTESTSHOT
270 PROCUSEFUEL
280 PROCSPEEDUP
290 UNTILHIT OREMPY
300 UNTILEMPY
310 PROCAGAIN
320 UNTILFALSE
330
340 DEFPROCSTARMOVE:IFTIME<STARTIME%EN
DPROC
350 VDU19,STARCOL%,0;0;:STARCOL%=STARC
OL%+1
360 IFSTARCOL%>6 STARCOL%=1
370 VDU19,STARCOL%,7;0;:STARTIME%=TIME
+STARINT%
380 ENDPROC
390
400 DEFPROCSMOVE:LOCALTNX%,TNY%;IFTIME
<SMOVETIME%ENDPROC
410 IFI$="K"PROCKEYBOARD ELSEPROCJOYST
ICK
420 IFRND(5)=1TXA%=RND(MTV%/4)-MTV%/8:
IFRND(2)=1TYD%=-TYD%
430 TXV%=TXV%+TXA%:IFTXV%<-MTV%TXV%=-M
TV%:TXA%=-TXA%
440 IFTXV%>MTV%TXV%=MTV%:TXA%=-TXA%
450 IFTXV%<0TYV%=MTV%+TXV%ELSETYV%=MTV
%-TXV%
460 IFTYD%<0TYV%=-TYV%

```

```

470  TNX%=TX%+TXV%-SXV%:TNY%=TY%+TYV%-S
YV%
480  IFTNX%>1875TNX%=1875:TXV%=-TXV%
490  IFTNX%<-1875TNX%=-1875:TYV%=-TYV%
500  IFTNY%>1500TNY%=1500:TYV%=-TYV%:TY
D%=-TYD%
510  IFTNY%<-1500TNY%=-1500:TYV%=-TYV%:
TYD%=-TYD%
520  PROCSHIPSHOW(TNX%,TNY%):PROCSHIPSH
OW(TX%,TY%):TX%=TNX%:TY%=TNY%
530  SMOVETIME%=SMOVETIME%+SMOVEINT%:EN
DPROC
540
550  DEFPROCSHIPSHOW(X%,Y%):MOVEX%,Y%:I
FX%<575PRINTS$ELSEPRINTCHR$224
560  PLOT69,-490+(X%*150)/1945,-412+(Y%
*100)/1560:PLOT65,8,0:PLOT65,0,4
570  PLOT65,-8,0:IFABSX%<300ANDABSY%<30
0ANDRND(F%/3+15)=1PROCFIRE(X%+64,Y%-16)
580  ENDPROC
590
600  DEFPROCFIRE(X%,Y%):MOVEX%,Y%:DRAW0
,-512:SOUND17,1,100,15
610  PROCFUEL(1):MOVEX%,Y%:DRAW0,-512:E
NDPROC
620
630  DEFPROCTESTSHOT
640  IFI$="K"ANDINKEY(-99)PROCshoot
650  IFI$="J"AND(ADVAL(0)AND3)=1PROCsho
OT
660  ENDPROC
670
680  DEFPROCshoot
690  MOVE-32,16:PRINT"*":SOUND16,-15,4,
2:SOUND17,1,255,10:PROCWAIT(1)
700  MOVE-32,16:PRINT"*"
710  IFTX%>-128ANDTX%<0ANDTY%>0ANDTY%<3
2PROCHIT:PROCSTARMOVE
720  PROCFUEL(1):ENDPROC
730
740  DEFPROCHIT:PROCSHIPSHOW(TX%,TY%):F
ORI%=-15TO0:IFADVAL(-8)>3PROCmusic
750  SOUND0,I%,4,2:PROCSTARMOVE:NEXT:PR
OCSCORE(10):HIT=TRUE
760  IFF%<91PROCFUEL(-20)ELSEIFF%<100P

```

```

ROCFUEL (F%-50)
  770 ENDPROC
  780
  790 DEFPROCUSEFUEL: IFTIME<FUELTIME%END
PROC
  800 IFF%>0PROCFUEL(1):SOUND2,-15,150,1
  810 FUELTIME%=TIME+FUELINT%:ENDPROC
  820
  830 DEFPROCSPPEEDUP: IFTIME<SPEEDTIME%EN
DPROC
  840 IFSMOVEINT%>2SMOVEINT%=SMOVEINT%-1
  850 IFFUELINT%>70FUELINT%=FUELINT%-10
  860 MTV%=MTV%+1:MSV%=MTV%*1.4+5
  870 IFSTARINT%>1STARINT%=STARINT%-1
  880 SPEEDTIME%=TIME+SPEEDINT%:ENDPROC
  890
  900 DEFPROCKEYBOARD
  910 SXV%=0:SYV%=0:IFINKEY(-54)SYV%=MSV
%
  920 IFINKEY(-86)SYV%=-MSV%
  930 IFINKEY(-70)SXV%=MSV%
  940 IFINKEY(-85)SXV%=-MSV%
  950 ENDPROC
  960
  970 DEFPROCJOYSTICK
  980 SXV%=0:SYV%=0:IFADVAL(1)<16380 SXV
%=-MSV%*((16380-ADVAL(1))/16380)
  990 IFADVAL(1)>49140SXV%=MSV%*((ADVAL(
1)-49140)/16380)
  1000 IFADVAL(2)<16380SYV%=MSV%*((16380-
ADVAL(2))/16380)
  1010 IFADVAL(2)>49140SYV%=-MSV%*((ADVAL
(2)-49140)/16380)
  1020 ENDPROC
  1030
  1040 DEFPROCSCORE(X):MOVE124,-412:IFX<>
0 PRINT;S%;:MOVE124,-412:S%=S%*1.2+X
  1050 IFADVAL(-8)>3PROC MUSIC
  1060 PRINT;S%;:ENDPROC
  1070
  1080 DEFPROCFUEL(X):MOVE124,-462:IFX<>0
PRINT;F%;:MOVE124,-462
  1090 IFADVAL(-8)>3PROC MUSIC
  1100 F%=F%-X:PRINT;F%;
  1110 IFF%<1 EMPTY=TRUE ELSEEMPTY=FALSE

```



```

1120 ENDPROC
1140 DEFPROCSTATIC
1150 VDU26,29,0;0;24,0;0;300;200;:GCOL
0,128:CLG:GCOL0,7:MOVE0,0:DRAW300,0
1160 DRAW300,200:DRAW0,200:DRAW0,0:MOVE
116,67:DRAW100,67:DRAW100,83
1170 MOVE184,67:DRAW200,67:DRAW200,83
1180 MOVE100,116:DRAW100,132:DRAW116,13
2:MOVE184,132:DRAW200,132:DRAW200,116
1190 VDU26,5::MOVE444,50:PRINT"Fuel":MO
VE380,100:PRINT"Score":VDU29,640;512;
1200 MOVE60,0:DRAW140,0:MOVE-60,0:DRAW-
140,0:MOVE0,40:DRAW0,120:MOVE0,-40
1210 DRAW0,-120:MOVE120,160:DRAW200,160
:DRAW200,100:MOVE120,-160:DRAW200,-160
1220 DRAW200,-100:MOVE-120,160:DRAW-200
,160:DRAW-200,100
1230 MOVE-120,-160:DRAW-200,-160:DRAW-2
00,-100:GCOL3,8:PROCSCORE(0)
1240 PROCFUEL(0):FORI%=1TO6:SOUND1,1,4,
5:PROC MUSIC
1250 PROCSTARMOVE:NEXT:SOUND17,0,0,1:EN
DPROC
1260
1270 DEFPROCSHIPMAKE:LOCALX%:IFRND(2)=1
X%=1ELSEX%=-1
1280 TX%=1875*X%:TY%=RND(3051)-1526:TXV
%=RND(MTV%)*-X%
1290 TYV%=(MTV%-ABS(TXV%))*X%:TXA%=RND(
3)-2:IFRND(2)=1TYD%=1ELSETYD%=-1
1300 PROCSHIPSHOW(TX%,TY%):HIT=FALSE:EN
DPROC
1310
1320 DEFPROCSTARS(N,D):LOCALI%:B$=" **
PURSUIT SHIP **":I%=RND(-9):STARINT%=0
1330 PROCSTARMAKE:VDU29,640;512;5:GCOL3
,8:MOVE-640,0:PRINTB$
1340 FORI%=1TON:GCOL1,(I%MOD6)+1:PROCST
ARSHOW(SX,SY):PROCSTARSHOW(-SX,-SY)
1350 PROCSTARNEXT(D):PROCSTARMOVE:IFADV
AL(-8)>3 PROC MUSIC
1360 NEXT:STARINT%=10:GCOL3,8:MOVE-640,
0:PRINTB$:I%=RND(-TIME):ENDPROC
1370
1380 DEFPROCSTARMAKE: SX=RND(101)-1:SY=R

```

```

ND(101)-1:IF SX<25ANDSY<25PROCSTARMAKE
1390 IFRND(2)=1 SX=-SX
1400 IFRND(2)=1 SY=-SY
1410 ENDPROC
1420
1430 DEFPROCSTARSHOW(X,Y):X=X-(X MOD 8)
:Y=Y-(Y MOD 8):IFPOINT(X,Y)=0 PLOT69,X,Y
1440 IFPOINT(X,Y+4)=0ANDABSX>160ORABSY>
128PLOT69,X,Y+4
1450 IFPOINT(X+8,Y)=0ANDABSX>320ORABSY>
256PLOT69,X+8,Y
1460 IFPOINT(X+8,Y+4)=0ANDABSX>480ORABS
Y>384PLOT69,X+8,Y+4
1470 ENDPROC
1480
1490 DEFPROCSTARNEXT(D):SX=SX*D:SY=SY*D
:IFABSSX>640ORABSSY>640 PROCSTARMAKE
1500 ENDPROC
1510
1520 DEFPROCBLACKOUT(C):FOR I%=1 TO 6:V
VDU19,I%,0;0;:VDU19,I%+8,C;0;
1530 NEXT:ENDPROC
1540
1550 DEFPROCAGAIN
1560 IFNOTHIT PROCSHIPSHOW(TX%,TY%)
1570 *FX 15,0
1580 VDU19,8,7;0;:GCOL 1,8
1590 IFS%>H%H%=S%:A$="The highest score
!"ELSEA$="High score: "+STR$(H%)
1600 MOVE-640,325:PRINTA$
1610 B$="Another game? (Y/N)":REPEATUNT
ILINKEY$(50)="":PROCBLACKOUT(2)
1620 MOVE-640,250:PRINTB$:REPEATK$=GET$
:UNTILK$="Y"ORK$="N"
1630 IFK$="N" VDU22,7:END:ENDPROC
1640 GCOL 3,8:MOVE-640,250:PRINTB$:PROC
SCORE(0):PROCFUEL(0)
1650 MOVE-640,325:PRINTA$
1660 ENDPROC
1670
1680 DEFPROCINIT:VDU4:GCOL0,7:GCOL0,128
:PROCBLACKOUT(2)
1690 VDU19,8,2;0;19,15,7;0;:IFH%=-1 CLG
1700 VDU23,224,&01&03&7F&FF&7F&3F&0F&03
1710 VDU23,225,&80&C0&FE&FF&FE&FC&F0&C0

```

```

1720  S$=CHR$224+CHR$225:MTV%=12:MSV%=2
4:S%=0:DD%=6:F%=200:STARTIME%=0
1730  STARCOL%=1:SMOVETIME%=0:SMOVEINT%=
25:FUELTIME%=0:FUELINT%=200
1740  SPEEDTIME%=0:SPEEDINT%=1500:SX%=0:
SY%=0:EMPTY=FALSE
1750  ENVELOPE 1,129,-6,-4,-3,16,32,96,6
4,-4,-2,-100,126,100:ENDPROC
1760
1770  DEFPROCINTRO
1780  B$=CHR$141+CHR$131+"P U R S U I T
    S H I P"
1790  PRINTTAB(6,6)B$
1800  PRINTTAB(6,7)B$
1810  PRINTTAB(8,15)CHR$129"AUTHOR : J.A
. Dean 1983"
1820  PRINTTAB(3,16)CHR$133"Do you want
to use a joystick or"
1830  PRINTTAB(3,17)CHR$133"the keyboard
for control? (J/K):";
1840  REPEATI$=GET$:UNTILI$="J"ORI$="K":
PRINTI$
1850  PRINTTAB(3,19)"Do you want instruc
tions? (Y/N):";
1860  REPEATK$=GET$:UNTILK$="Y"ORK$="N":
IFK$="N"ENDPROC
1870  CLS
1880  PRINT''''"You are the pilot of a p
ursuit ship and"
1890  PRINT"it is your task to seek out
and destroy"
1900  PRINT"the alien craft which threat
en Earth's"
1910  PRINT"saftey."
1920  PRINT'''"There is a long range rada
r scan shown"
1930  PRINT"at the bottom left of the sc
reen, this"
1940  PRINT"indicates the position of en
emy craft"
1950  PRINT"relative to the view on the
ship's main"
1960  PRINT"scanner."
1970  PROCSPACE
1980  PRINT'''"You must centre the enemy

```

```

    ship in your"
1990 PRINT"cross wires before firing to
    destroy it.";
2000 PRINT"When you use the controls, t
he ship will";
2010 PRINT"move in the direction reques
ted, giving"
2020 PRINT"the impression of the target
    moving in"
2030 PRINT"the opposite direction!"
2040 PRINT'"For each alien craft destr
oyed you will"
2050 PRINT"be awarded extra fuel. But n
ote that if"
2060 PRINT"the aliens feel threatened t
hey may well";
2070 PRINT"open fire on you, and mainta
ining your"
2080 PRINT"defensive shield will reduce
    your fuel."
2090 PROCSPACE
2100 IFI$="K"PROCKEYINST ELSEPROCJOYINS
T
2110 PROCSPACE
2120 PRINTTAB(0,3)"There will be a shor
t introduction while";
2130 PRINT"the galaxy is drawn. When it
    is complete";
2140 PRINT"BATTLE STATIONS will be soun
ded and the"
2145 PRINT"game will commence."
2150 PRINT'"Remember: firing your las
er uses fuel,"
2160 PRINT"being hit by the enemy uses
fuel and"
2170 PRINT"fuel is consumed by the norm
al running"
2180 PRINT"of your ship. The only way t
o regain"
2190 PRINT"fuel is to destroy the alien
s."
2200 PRINT"The game ends when your fuel
supply is"
2205 PRINT"exhausted."
2210 PRINT'"Good luck! I've a feeling

```

```

you'll need it"
2220 PROCSPACE
2230 ENDPROC
2240
2250 DEFPROCJOYINST
2260 PRINTTAB(0,7)"Use the left hand jo
ystick."
2270 PRINT"Press the fire button to sta
rt.";
2275 PRINT
2280 REPEAT:UNTIL(ADVAL(0)AND 3)=1
2290 PRINT'"Fine, now you can use the
joystick to"
2295 PRINT"move your ship."
2300 PRINT'"Use the fire button when th
e target ship";
2305 PRINT"is in your sights."
2310 ENDPROC
2320
2330 DEFPROCKEYINST
2340 PRINT'"To move your ship use the
following"
2350 PRINT"keys (one or more at a time)
:"
2360 PRINT'"SPC(14)"'U' ---- Up"
2370 PRINT'" Left ---- 'H'      'J' ---- R
ight"
2380 PRINT'"SPC(14)"'N' ---- Down"
2390 PRINT'"Use the SPACE BAR to fire
when you have"
2400 PRINT"the target in your sights."
2410 ENDPROC
2420
2430 DEFPROCSPACE:REPEAT:UNTILINKEY$(25
)=" "
2440 PRINTTAB(1,24)"Press the SPACE BAR
to continue.";
2450 REPEATUNTILGET=32:CLS:ENDPROC
2460
2470 DEFPROCWAIT(X):WAITTIME%=TIME+X:RE
PEATUNTILTIME>=WAITTIME%:ENDPROC
2480
2490 DEFPROC MUSIC:IFF%/16+1<DD%DD%=F%/1
6+1
2500 READPP%:IFPP%<0RESTORE:READPP%

```

```
2510 SOUND3,-15,PP%,DD%:SOUND3,0,0,1:IF
ADVAL(-8)>3PROC MUSIC
2520 ENDPROC
2530
2540 DATA 0,0,12,12,28,28,12,12,0,0,12,
12,28,28,12,12,20,20,32,32,48,48,32,32
2550 DATA 20,20,32,32,48,48,32,32,28,28
,12,12,0,0,12,12,28,28,12,12,0,0,12,12
2560 DATA 48,48,32,32,20,20,32,32,48,48
,32,32,20,20,32,32,-1
```

# MARTIAN ATTACK

By Nikola Stojanovic

Martian Attack is an arcade style game for a 32K BBC Micro, once again the earth is under attack from strange Martian creatures!

Your task is to defeat the irksome invaders and save the earth. To aid your mission you have a helicopter and a plane under control. The plane is used to shoot down the alien spacecraft while the helicopter attempts to reach the landing pad at the top of the screen. When you reach the platform you pass on to the next level of the game.

As the helicopter nears the platform the aliens change from modules into heads, and the number of points scored for zapping them increases. At each successive level, the number of aliens increases, making it more and more difficult to complete the level.

A point to note when typing in this program is that a lot of the variables are in lower case letters and should be typed in as such. Be careful not to confuse them with capitals or the program may not work properly.

## Program Description

**80** Define the envelope used for the tunes and explosions.

**90** Define the first character set

**110** Change mode and define the text window.

**112** Set colours.

**120** Turn off the cursor.

**130-190** Print the title page on the screen.

**200** Play the Star Wars tune.

**210** Wait for a key to be pressed and clear the screen.

**220** Define the third character set.

**230-410** Print the instructions on the screen.

**480** Define the second character set

**490-520** Print the city on the screen.

**530** Print the landing pad.

**540-560** Reset the score to zero and initialize some variables.  
**570-590** Reset the variables for a new game and print the current high score.  
**700** Print the helicopter at its start position.  
**710** Print the plane at its start position.  
**720** Play a fanfare.  
**750** Print either heads or modules on the screen depending on the helicopter's height  
**760-870** Test the keyboard controls and act accordingly.  
**890-920** Decide whether or not the octopus should be moved.  
**940** Test for the end of a wave or the game.  
**950** Go to either the 'game over' procedure or the 'next wave' procedure.  
**960** If the game is not over, return to the main loop.  
**970** Find out whether another game is wanted and act accordingly.  
**990-1080** PROCnext\_wave  
**1000** Give 100 points bonus.  
**1020** Play the 'next wave' tune.  
**1030** linmove any heads or modules from the screen.  
**1040** If the player has not got beyond the seventh wave increase the speed.  
**1060-1080** See if the player's score is bigger than 9999 - the best possible.  
**1090-1190** PROCgame\_over  
**1100** Print the score.  
**1120-1150** Move the 'earth invasion' message down the screen.  
**1170-1190** See if the player's score has exceeded the best possible  
**1200-1280** FNagain - ask whether another game is wanted and act accordingly.  
**1290-1310** PROCcrash - GOSUB to the appropriate crash routine.  
**1320-1370** The module-helicopter crash.  
**1380-1430** The octopus-helicopter crash.  
**1440-1480** The module-missile crash.  
**1490-1540** The helicopter-missile crash.  
**1550-1590** The head-missile crash.  
**1600-1640** The octopus-missile crash.



**1650-1700** The module-plane crash.  
**1710-1750** The helicopter-plane crash.  
**1160-1810** The head-plane crash  
**1820-1810** The octopus-plane crash.  
**1880-1930** PROCcls - partially clear the screen.  
**1940-2000** PROCscore - print the player's score.  
**2010-2120** PROCdestroy\_plane  
**2020-2040** Make an explosion noise and flash the explosion the screen.  
**2050-2010** Reduce the number of planes and see if the game ended.  
**2080-2120** Play the 'death march' tune and print the plane at its initial position.  
**2130-2240** PROCdestroy\_chopper  
**2140-2160** Make an explosion noise and flash the explosion of the screen.  
**2110-2190** Reduce the number of helicopters and see if the has ended.  
**2200-2240** Play the 'death march' tune and print the helicopter in its initial position.  
**2250-2320** PROCclear\_heads - remove all heads from the screen.  
**2330-2400** PROCc1ear-modules - remove all the modules the screen.  
**2410-2510** PROCexplosion - make an explosion noise and an explosion of a given size at a given position on the screen.  
**2520-2600** PROCoctopus - move the octopus and check that it has not hit anything.  
**2610-2810** PROCmodule - move the modules and check that they have not hit anything.  
**2820-3000** PROChead - move the heads and check that they have not hit anything.  
**3010-3110** PROCeject\_invader - replace a hit invader.  
**3120-3190** PROCplane\_up  
**3160-3220** Move the plane up and left.  
**3230-3290** Move the plane up and right.  
**3300-3420** PROCcheck\_plane - check that the plane has not crashed into anything.  
**3430-3530** PROCchopper\_up - move the helicopter up.

**3540-3620** PROCcheck\_chopper - check that the helicopter not crashed into anything.  
**3630-3800** PROCplane\_down  
**3670-3730** Move the plane down and left.  
**3740-3800** Move the plane down and right.  
**3810-3880** PROCchopper\_down - move the helicopter down.  
**3890-3930** PROCrotate\_plane - change the direction of the plane.  
**3940-4090** PROCchopper\_left - move the helicopter left and check that it has not hit anything.  
**4100-4160** PROCfire - make the firing noise and fire a missile in the direction that the plane is facing.  
**4170-4280** PROCfire\_1left - fire a missile to the left and check for a hit.  
**4410-4560** PROCfire\_right - fire a missile to the right and check for a hit.  
**4410-4560** PROCchopper\_right - move the helicopter to the right and check that it has not hit anything.  
**4570-4670** PROChit\_check1 - check to see if the missile has hit anything.  
**4680-4780** PROChit\_check2 - check to see if the missile has hit anything.  
**4790-4810** PROCdelay - delay for a given time.  
**4820-4840** FNscreen - get the ASCII code of a character on the screen.  
**4850-4880** PRO Ctune - play a given tune.  
**5000** Data for 'fanfare'.  
**5010-5020** Data for 'death march'.  
**5030-5050** Data for 'end of wave' tune.  
**5060-5090** Data for Star Wars tune.  
**5100-5140** PROCpress - wait for a key to be pressed.  
**5150-5250** PROCdefine - define a given character set.  
**5270-5360** Data for the first character set.  
**5370-5440** Data for the second character set.  
**5450-5550** Data for the third character set.

---

```

20  REM *  --==MARTIAN ATTACK==-- *
30  REM *
40  REM *  Converted by J.Byrne *
50  REM *           & A.Ruddle *
60  REM *
70  REM *****
80  ENVELOPE 1,1,0,0,0,0,0,0,127,-1,-
1,-1,126,0
90  PROCdefine(1)
100 REM *** Introduction ***
110 MODE4:VDU28,3,27,35,3,19,1,2;0;0
120 VDU 23,1,0;0;0;0;
130 VDU 31,17,4,235,31,7,5,224,231,32
,227,226,32,228,229,32,233,234,32,232,32
,227,226,32,225,226,13,10,10
140 PRINT" *      * * * * * * * * * *
* * "
150 PRINT" * *      *      *      * * *
* * "
160 PRINT" * * *      *      *      * * *
* * "
170 PRINT"*      *      *      *      * *
* * "
180 PRINT"*      *      *      *      * * *
* * "
190 PRINTTAB(0,19)"Author N.I. Stojan
ovic ";TAB(0,20)"Converted by A.Ruddle
& J.Byrne"
200 PROCTune(4)
210 PROCpress:CLS
220 PROCdefine(3)
230 PRINTTAB(9,3)"MARTIAN ATTACK"
240 PRINT'" In this game you must sa
ve the earth from the invasion of the
marauding martian thingies!"
250 PRINT'" You have a helicopter an
d a plane. The plane is used to shoot
down aliens, while the helicopter";
260 PRINT"attempts to reach the landi
ng padat the top of the screen. When y
ou reach the pad you move on to a more d
ifficult level."
270 PRINT'CHR$233;CHR$234;CHR$235;" -
Your plane"
280 PRINT'CHR$231;CHR$232;" - Your h

```

```

elicopter"
290  PROCpress:CLS
300  PRINTTAB(9,4)"MARTIAN ATTACK"
310  PRINT'TAB(3)CHR$229
320  PRINTTAB(3)CHR$230;" - Module
    25 pts"
330  PRINT'TAB(3)CHR$238;" - Octopus
    50 pts"
340  PRINT'TAB(3)CHR$237;" - Head
    75 pts"
350  PRINT'TAB(2)CHR$231;CHR$232;" - H
elicopter -100 pts"
360  PRINT'"Landing helicopter 100 pt
s bonus"
370  PROCpress:CLS
380  PRINTTAB(9,4)"MARTIAN ATTACK"
390  PRINT'" Controls:-"
400  PRINTTAB(15)"A - up"
410  PRINT'TAB(15)"Z - down"
420  PRINT'TAB(3)"Helicopter: < - left
"
430  PRINT'TAB(15)"> - right"
440  PRINT'TAB(8)"Plane: < - turn arou
nd"
450  PRINT'TAB(15)"> - missile"
460  PRINT'TAB(5)"SPACE BAR - helicopt
er/plane"
470  PROCpress
480  CLS:PROCdefine(2)
490  REM *** Draw city ***
500  VDU 31,1,19,226,227,228,31,7,19,2
29,32,32,32,32,234,32,32,32,32,234,234,3
1,24,19,226,228,234,32,32,229
510  VDU 31,1,20,225,225,225,31,6,20,2
30,231,232,226,228,32,234,32,32,32,32,23
4,234,32,226,227,228,32,225,225,234,32,2
30,231,232
520  VDU 31,0,21,240,225,224,225,237,2
38,239,233,240,225,224,240,234,240,235,2
36,239,234,234,240,225,224,225,240,224,2
25,234,237,238,233,240,239
530  PRINTTAB(15,1)CHR$241;TAB(10,2)"\
___";TAB(17,2)"___/";TAB(14,2)CHR$242;CH
R$243;CHR$244:PROCdefine(3)
540  PRINTTAB(0,0)"Score 0000      3

```

```

3   Hi 0000";TAB(12,0)CHR$231;CHR$232;
TAB(18,0)CHR$228;CHR$227;CHR$226
550   nj%=0
560   DIMr(7):DIMm(7):DIMn(7):DIMs(7):D
IMh(7):DIMg(7):DIMu(7):DIMv(7):DIMk(7)
570   REPEAT
580       a%=3:z=1:c%=3:d%=3:sc%=0:rc%=0:
t%=0:CAP%=1:PRINTTAB(11,23)"Helicopter"
590       VDU 31,28,0
600       IF nj%<1000 PRINT"0";
610       IF nj%<100 PRINT"0";
620       IF nj%<10 PRINT"0";
630       PRINT;nj%
640       REM Initial conditions
650       REPEAT
660           PRINTTAB(15,0);d%;TAB(22,0);c
%:PROCscore(sc%)
670           e%=0:f%=0:y%=0:x%=0:w%=0:b%=0
680           ay%=12:ax%=25:kh=17:khh=4:kg=
1:gx%=0:ey%=0:ex%=0:bm%=1:bg%=0
690           hy%=kh:hx%=khh:gy%=kg
700           PRINTTAB(hx%,hy%)CHR$231;CHR$
232
710           PRINTTAB(ax%,ay%)CHR$233;CHR$
234;CHR$235
720           PROCTune(1):E%=0
730           REM Main Loop
740           REPEAT
750               IF hy%<7 PROChead ELSE PROC
module
760                   a$=INKEY$(0)
770                   *FX 15,1
780                   IF a$="A"ANDCAP%=0 PROCplan
e_up
790                       IF a$="A"ANDCAP%=1 PROCchop
per_up
800                           IF a$="Z"ANDCAP%=0 PROCplan
e_down
810                               IF a$="Z"ANDCAP%=1 PROCchop
per_down
820                                   IF a$=", "ANDCAP%=0 PROCrota
te_plane
830                                       IF a$=", "ANDCAP%=1 PROCchop
per_left
840                                           IF a$="."ANDCAP%=0 PROCfire

```

```

850          IF a$="."ANDCAP%=1 PROCchop
per_right
860          IF a$=" "CAP%=1-CAP%:SOUND
1,-15,100,1
870          IF CAP%=0 PRINTTAB(11,23)"A
eroplane " ELSE PRINTTAB(11,23)"Helicopt
er"
880          IF E%<>0 GOTO 940
890          IF b%=1 PROCCoctopus:GOTO 93
0
900          IF x%>10-a% b%=1:x%=0:kg=3:
gx%=hx%:PROCCoctopus:GOTO 930
910          IF y%>10-a% b%=1:y%=0:kg=ay
%-1:gx%=RND(31):PROCCoctopus:GOTO 930
920          y%=y%+1:x%=x%+1
930          t%=t%+1
940          UNTIL E%<>0
950          IF E%=1 PROCgame_over ELSE PR
OCnext_wave
960          UNTIL E%<>2
970          UNTIL FNagain=FALSE
980          END
990          DEF PROCnext_wave
1000         sc%=sc%+100:PROCscore(sc%)
1010         PRINTTAB(11,10)"** Success **'"
The invasion is suppressed !";TAB(7,
14)"Prepare for next wave."
1020         PROCTune(3)
1030         PROCClear_heads:PROCClear_modules
:PROCcls
1040         IFa%<>7 a%=a%+1:z=z-.2
1050         PROCcls
1060         IFsc%<9999 ENDPROC
1070         PRINTTAB(0,10)"You have the best
possible score!"
1080         E%=1:ENDPROC
1090         DEF PROCgame_over
1100         PRINTTAB(15,0);d%;TAB(22,0);c%:PR
OCscore(sc%)
1110         PROCcls
1120         FORi%=5TO15
1130         PRINTTAB(1,i%-1);SPC(32);TAB(3,
i%)"The Earth has been Invaded !":SOUND
1,-15,RND(255),RND(5)
1140         PROCdelay(8)

```

```

1150     NEXT
1160  PROCcls
1170  IFsc%<9999 ENDPROC
1180  PRINTTAB(0,10)"You have the best
possible score"
1190  ENDPROC
1200  DEF FNagain
1210  IF sc%>9999 =FALSE
1220  PRINTTAB(1,10)"Do you want anothe
r game (y/n)?"
1230  REPEAT:t$=GET$:UNTILt$="Y"ORT$="N
"
1240  IFt$="N" PRINTTAB(5,12)"Maybe som
e other time !";TAB(11,15)"GOODBYE !":=F
ALSE
1260  IFsc%>nj% nj%=sc%
1270  PROCclear_heads:PROCclear_modules
:PROCcls
1280  =TRUE
1290  DEF PROCcrash(N%)
1300  ON N% GOSUB 1330,1390,1450,1500,1
560,1610,1660,1710,1770,1830
1310  ENDPROC
1320  REM Module-Helicopter crash
1330  PRINTTAB(n(i%),m(i%)) " ";TAB(n(i%
),m(i%)+1) " ";
1340  k(i%)=1:FORj%=1TOa%:m(j%)=0:u(j%)
=r(j%):v(j%)=n(j%):NEXT:PROCeject_invale
r:FORj%=1TOa%:r(j%)=u(j%):n(j%)=v(j%):NE
XT
1350  PROCdestroy_chopper
1360  PROCscore(sc%)
1370  RETURN
1380  REM Octopus-Helicopter crash
1390  PRINTTAB(gx%,gy%) " "
1400  PROCdestroy_chopper
1410  kg=0:gy%=0:gx%=0:b%=0
1420  PROCscore(sc%)
1430  RETURN
1440  REM Module-Missile crash
1450  PROCexplode(n(z%),m(z%),1,1,242)
1460  k(z%)=1:FORj%=1TOa%:m(j%)=0:u(j%)
=r(j%):v(j%)=n(j%):NEXT:PROCeject_invale
r:FORj%=1TOa%:r(j%)=u(j%):n(j%)=v(j%):NE
XT

```

```

1470    sc%=sc%+25:PROCscore(sc%)
1480    RETURN
1490    REM Helicopter-Missile crash
1500    IF i%=4 i%=3
1510    sc%=sc%-50:IF sc%<0 sc%=0
1520    PROCscore(sc%)
1530    PROCdestroy_chopper
1540    RETURN
1550    REM Head-Missile crash
1560    PROCexplode(g(z%),h(z%),1,0,244)
1570    k(z%)=1:FORj%=1TOa%:h(j%)=0:u(j%)
=s(j%):v(j%)=g(j%):NEXT:PROCeject_invade
r:FORj%=1TOa%:s(j%)=u(j%):g(j%)=v(j%):NE
XT
1580    sc%=sc%+75:PROCscore(sc%)
1590    RETURN
1600    REM Octopus-Missile crash
1610    PROCexplode(gx%,gy%,1,0,244)
1620    sc%=sc%+50:PROCscore(sc%)
1630    kg=0:gy%=0:gx%=0:b%=0
1640    RETURN
1650    REM Module-plane crash
1660    PRINTTAB(n(i%),m(i%)) " ";TAB(n(i%
),m(i%)+1) " "
1670    k(i%)=1:FORj%=1TOa%:m(j%)=0:u(j%)
=r(j%):v(j%)=n(j%):NEXT:PROCeject_invade
r:FORj%=1TOa%:r(j%)=u(j%):n(j%)=v(j%):NE
XT
1680    PROCdestroy_plane
1690    PROCscore(sc%)
1700    RETURN
1710    REM Helicopter-Plane crash
1720    sc%=sc%-100: IFsc%<0 sc%=0
1730    PROCscore(sc%)
1740    PROCdestroy_plane:PROCdestroy_cho
pper
1750    RETURN
1760    REM Head-Plane crash
1770    PRINTTAB(g(i%),h(i%)) " "
1780    k(i%)=1:FORj%=1TOa%:h(j%)=0:u(j%)
=s(j%):v(j%)=g(j%):NEXT:PROCeject_invade
r:FORj%=1TOa%:s(j%)=u(j%):g(j%)=v(j%):NE
XT
1790    PROCdestroy_plane
1800    PROCscore(sc%)

```



```

1810 RETURN
1820 REM Octopus-Plane crash
1830 PRINTTAB(gx%,gy%) " "
1840 PROCdestroy_plane
1850 kg=0:gy%=0:gx%=0:b%=0
1860 PROCscore(sc%)
1870 RETURN
1880 DEF PROCcls
1890 PRINTTAB(0,1)SPC(15)TAB(16,1)SPC(
16)TAB(0,2)SPC(10)TAB(21,2)SPC(11)
1900 FORL%=3TO18
1910 PRINTTAB(0,L%)SPC(32)
1920 NEXT
1930 ENDPROC
1940 DEF PROCscore(sc%)
1950 rc%=t%
1960 IFsc%<10 PRINTTAB(6,0)"000";sc%:E
NDPROC
1970 IFsc%<100 PRINTTAB(6,0)"00";sc%:E
NDPROC
1980 IFsc%<1000 PRINTTAB(6,0)"0";sc%:E
NDPROC
1990 PRINTTAB(6,0);sc%
2000 ENDPROC
2010 DEF PROCdestroy_plane
2020 SOUND 0,1,6,5:FOR i%=1 TO 5
2030 PROCdelay(3):VDU 31,ax%,ay%,239
,240,241
2040 PROCdelay(3):VDU 31,ax%,ay%,32,
32,32:NEXT
2050 c%=c%-1
2060 IF c%=0 E%=1:ENDPROC
2070 PRINTTAB(22,0);c%
2080 ay%=12:ax%=25:f=0
2090 PRO Ctune(2)
2100 PRINTTAB(ax%,ay%)CHR$233;CHR$234;
CHR$235
2110 x%=0
2120 ENDPROC
2130 DEF PROCdestroy_chopper
2140 SOUND 0,1,6,5:FOR i%=1 TO 5
2150 PROCdelay(3):VDU 31,hx%,hy%,242
,243
2160 PROCdelay(3):VDU 31,hx%,hy%,32,
32:NEXT

```

```

2170   d%=d%-1
2180   IF d%=0 E%=1:ENDPROC
2190   PRINTTAB(15,0);d%
2200   kh=17:hy%=17:kh=4:hx%=4:e%=0
2210   PROCtune(2)
2220   PRINTTAB(hx%,hy%)CHR$231;CHR$232
2230   y%=0
2240   ENDPROC
2250   DEF PROCclear_heads
2260   FORi%=1TOa%
2270       IF h(i%)<>0 PRINTTAB(g(i%),h(i%
) ) " "
2280       s(i%)=0:h(i%)=0:g(i%)=0
2290       NEXT
2300   w%=0
2310   PRINTTAB(0,0)"Score";TAB(25,0)"Hi
";TAB(12,0)CHR$231;CHR$232;TAB(18,0)CHR$
228;CHR$227;CHR$226
2320   ENDPROC
2330   DEF PROCclear_modules
2340   FORi%=1TOa%
2350       IF m(i%)<>0 PRINTTAB(n(i%),m(i%
) ) " ";TAB(n(i%),m(i%)+1) " "
2360       r(i%)=0:m(i%)=0:n(i%)=0
2370       NEXT
2380   w%=0
2390   PRINTTAB(0,0)"Score";TAB(25,0)"Hi
";TAB(12,0)CHR$231;CHR$232;TAB(18,0)CHR$
228;CHR$227;CHR$226
2400   ENDPROC
2410   DEF PROCexplode(X%,Y%,XS%,YS%,C%)
2420   SOUND 0,1,6,5
2430   FOR i%=1 TO 5
2440       PROCdelay(3)
2450       FOR YM%=Y% TO Y%+YS%
2460           PRINTTAB(X%,YM%)STRING$(XS%,C
HR$(C%))
2470       NEXT:PROCdelay(3)
2480       FOR YM%=Y% TO Y%+YS%
2490           PRINTTAB(X%,YM%)STRING$(XS%,
" )
2500       NEXT:NEXT
2510   ENDPROC
2520   DEF PROCoctopus
2530   o%=INT kg/kg=kg+1/z/5:gy%=INT kg:

```

```

IF gy%=3 AND hy%=3 kg=4:o%=4:gy%=4
  2540 IF gy%=o% GOTO2590
  2550 REM Crash check
  2560 IF gy%=hy%AND(gx%=hx% OR gx%=hx%+
1) PRINTTAB(gx%,o%) " ":PROCcrash(2):ENDP
ROC
  2570 IF gy%=ay% AND (gx%=ax% OR gx%=ax
%+1 OR gx%=ax%+2) PRINTTAB(gx%,o%) " ":PR
OCcrash(10):ENDPROC
  2580 IF gy%>18 gy%=18:E%=1:ENDPROC
  2590 PRINTTAB(gx%,o%) " ";TAB(gx%,gy%)C
HR$238
  2600 ENDPROC
  2610 DEF PROCmodule
  2620 IF bm%=0 PROCclear_heads:bm%=1:bg
%=0
  2630 IF w%=a% GOTO 2700
  2640 p%=RND(31)
  2650 IFp%/(10-a%)-INT(p%/(10-a%))<>0 G
OTO2690
  2660 w%=w%+1
  2670 r(w%)=4:m(w%)=4:n(w%)=p%
  2680 PRINTTAB(n(w%),m(w%))CHR$229;TAB(
n(w%),m(w%)+1)CHR$230
  2690 IF w%=0 ENDPROC
  2700 i%=0:REPEAT
  2710 i%=i%+1
  2720 o%=INTTr(i%):r(i%)=r(i%)+1/z/6+(
sc%+t%)/20000:m(i%)=INTTr(i%)
  2730 IF m(i%)=o% THENGOTO2780
  2740 REM Crash check
  2750 IF m(i%)+1=hy% AND (n(i%)=hx% O
R n(i%)=hx%+1) PRINTTAB(n(i%),o%) " ";TAB
(n(i%),o%+1) " ":PROCcrash(1):UNTIL TRUE:
ENDPROC
  2760 IF m(i%)+1=ay% AND (n(i%)=ax% OR
n(i%)=ax%+1 OR n(i%)=ax%+2) PRINTTAB(n(i
%),o%) " ";TAB(n(i%),o%+1) " ":PROCcrash(7
):UNTIL TRUE:ENDPROC
  2770 IF m(i%)>17 m(i%)=17:E%=1
  2780 IF E%=0 PRINTTAB(n(i%),o%) " ";TAB
(n(i%),o%+1) " "
  2790 IF E%=0 PRINTTAB(n(i%),m(i%))CHR$
229;TAB(n(i%),m(i%)+1)CHR$230
  2800 UNTIL E%=1 OR i%=w%

```

```

2810  ENDPROC
2820  DEF PROChead
2830  IF bg%=0 PROCclear_modules:bg%=1:
bm%=0
2840  IF w%=a% THEN 2910
2850  p%=RND(31)
2860  IF p%/(10-a%)-INT(p%/(10-a%))>0
THEN 2900
2870  w%=w%+1
2880  s(w%)=10:h(w%)=10:g(w%)=p%
2890  PRINTTAB(g(w%),h(w%))CHR$237
2900  IF w%=0 ENDPROC
2910  i%=0:REPEAT
2920    i%=i%+1
2930    o%=INT(s(i%)):s(i%)=s(i%)+1.2/z
/6+(sc%+t%)/20000:h(i%)=INT(s(i%))
2940    IF h(i%)=o% THEN 2980
2950    REM Crash check
2960    IF h(i%)=ay% AND (g(i%)=ax% OR
g(i%)=ax%+1 OR g(i%)=ax%+2) PRINTTAB(g(i
%),o%) " ":PROCcrash(9):UNTIL TRUE:ENDPRO
C
2970  IF h(i%)>18 h(i%)=18:E%=1:UNTIL T
RUE:ENDPROC
2980  PRINTTAB(g(i%),o%) " ";TAB(g(i%),h
(i%))CHR$237
2990  UNTIL i%=w%
3000  ENDPROC
3010  DEF PROCeject_invader
3020  w%=w%-1
3030  j%=1
3040  FORi%=1TOa%
3050    IF k(i%)=1 GOTO3080
3060    u(j%)=u(i%):v(j%)=v(i%)
3070    j%=j%+1
3080  NEXT
3090  FORi%=1TOa%:k(i%)=0:NEXT
3100  FORi%=j%+1TOa%:u(i%)=0:v(i%)=0:NE
XT
3110  ENDPROC
3120  DEF PROCplane_up
3130  y%=0
3140  oy%=ay%:ox%=ax%
3150  IF f=1 GOTO3240
3160  REM +left

```

```

3170  ay%=ay%-1:ax%=ax%-1
3180  IF ay%<6 ay%=6:ax%=ox%:ENDPROC
3190  IF ax%<0 ax%=0:ay%=oy%:ENDPROC
3200  PROCcheck_plane:IF F% ENDPROC
3210  PRINTTAB(ox%,oy%) "      ";TAB(ax%,ay
%)CHR$233;CHR$234;CHR$235
3220  ENDPROC
3230  REM +right
3240  ay%=ay%-1:ax%=ax%+1
3250  IF ay%<6 ay%=6:ax%=ox%:ENDPROC
3260  IF ax%>29 ax%=29:ay%=oy%:ENDPROC
3270  PROCcheck_plane:IF F% ENDPROC
3280  PRINTTAB(ox%,oy%) "      ";TAB(ax%,ay
%)CHR$228;CHR$227;CHR$226
3290  ENDPROC
3300  DEF PROCcheck_plane
3310  F%=FALSE:FORi%=0TO2
3320      IF FNscreen(ax%+i%,ay%)<>32 F%=
TRUE
3330      NEXT
3340  IF NOT F% ENDPROC
3350  IF ay%=hy% AND (ax%+2=hx% OR ax%+1
=hx% OR ax%=hx% OR ax%=hx%+1) PRINTTAB(o
x%,oy%) "      ":PROCcrash(8)
3360  i%=0:REPEAT
3370      i%=i%+1
3380      IF (ay%=m(i%)+1 OR ay%=m(i%)) AN
D (ax%=n(i%) OR ax%+1=n(i%) OR ax%+2=n(i
%)) PRINTTAB(ox%,oy%) "      ":PROCcrash(7):
UNTIL TRUE:ENDPROC
3390  IF ay%=h(i%) AND (ax%=g(i%) OR ax
%+1=g(i%) OR ax%+2=g(i%)) PRINTTAB(ox%,o
y%) "      ":PROCcrash(9):UNTIL TRUE:ENDPROC
3400  UNTIL i%=w%
3410  IF ay%=gy% AND (ax%=gx% OR ax%+1=
gx% OR ax%+2=gx%) PRINTTAB(ox%,oy%) "      "
:PROCcrash(10)
3420  ENDPROC
3430  DEF PROCchopper_up
3440  o%=INT kh:kh=kh-.3*z:hy%=INT kh
3450  ad%=1
3460  PROCcheck_chopper:IF F% ENDPROC
3470  IF hx%>8 AND hx%<21 AND hy%=2 hy%
=3:kh=3:ENDPROC
3480  IF hy%=0 hy%=1:kh=1:ENDPROC

```

```

3490 REM Print helicopter
3500 PRINTTAB(hx%,o%) "    "
3510 IF e%=0 PRINTTAB(hx%,hy%)CHR$231;
CHR$232
3520 IF e%=1 PRINTTAB(hx%,hy%)CHR$224;
CHR$225
3530 ENDPROC
3540 DEF PROCcheck_chopper
3550 F%=FALSE
3560 IF hy%=o% F%=TRUE:ENDPROC
3570 IF hy%=ay% AND (hx%=ax%-1 OR hx%=a
x% OR hx%=ax%+1OR hx%=ax%+2) PRINTTAB(hx
%,o%) "    ":PROCcrash(8):F%=TRUE:ENDPROC
3580 G%=FALSE:FORi%=1TOw%
3590 IF G% OR hy%=m(i%)+ad% AND (hx%
=n(i%) OR hx%+1=n(i%)) PRINTTAB(hx%,o%) "
    ":PROCcrash(1):G%=TRUE:NEXT:F%=TRUE:EN
DPROC
3600 NEXT
3610 IF hy%=gy% AND (hx%=gx% OR hx%+1=g
x%) PRINTTAB(hx%,o%) "    ":PROCcrash(2):F%
=TRUE:ENDPROC
3620 ENDPROC
3630 DEF PROCplane_down
3640 y%=0
3650 oy%=ay%:ox%=ax%
3660 IF f=1 GOTO3750
3670 REM +left
3680 ay%=ay%+1:ax%=ax%-1
3690 IF ay%>18 ay%=18:ax%=ox%:ENDPROC
3700 IF ax%<0 ax%=0:ay%=oy%:ENDPROC
3710 PROCcheck_plane:IF F% ENDPROC
3720 PRINTTAB(ox%,oy%) "    ";TAB(ax%,ay
%)CHR$233;CHR$234;CHR$235
3730 ENDPROC
3740 REM +right
3750 ay%=ay%+1:ax%=ax%+1
3760 IF ay%>18 ay%=18:ax%=ox%:ENDPROC
3770 IF ax%>29 ax%=29:ay%=oy%:ENDPROC
3780 PROCcheck_plane:IF F% ENDPROC
3790 PRINTTAB(ox%,oy%) "    ";TAB(ax%,ay
%)CHR$228;CHR$227;CHR$226
3800 ENDPROC
3810 DEF PROCchopper_down
3820 o%=INT kh:kh=kh+.4*z:hy%=INT kh

```

```

3830  ad%=0
3840  PROCcheck_chopper:IF F% ENDPROC
3850  IF hy%=2 AND(hx%=9 OR hx%=10 OR h
x%=20 OR hx%=19) hy%=1:kh=1:ENDPROC
3860  IF hx%>10 AND hx%<19 AND hy%=2 E%
=2:ENDPROC
3870  IF hy%=18 hy%=17:kh=17:ENDPROC
3880  GOTO3500
3890  DEF PROCrotate_plane
3900  f=1-f
3910  IF f=0 THEN PRINTTAB(ax%,ay%)CHR$
233;CHR$234;CHR$235
3920  IF f=1 PRINTTAB(ax%,ay%)CHR$228;C
HR$227;CHR$226
3930  ENDPROC
3940  DEF PROCchopper_left
3950  x%=0:e%=1
3960  o%=INT khh:khk=khk-.7*z:hx%=INT k
hh
3970  IF hx%=o% ENDPROC
3980  REM Crash check
3990  IF hy%=ay% AND hx%=ax%+2 PRINTTAB
(o%,hy%) " " :PROCcrash(1):ENDPROC
4000  i%=0:REPEAT
4010  i%=i%+1
4020  UNTIL i%>=w% OR (hy%=m(i%)+1 OR
hy%=m(i%))AND hx%=n(i%)
4030  IF (hy%=m(i%)+1 OR hy%=m(i%))AND
hx%=n(i%) PRINTTAB(o%,hy%) " " :PROCcrash
(1):ENDPROC
4040  IF hy%=gy% ANDhx%=gx% PRINTTAB(o%
,hy%) " " :PROCcrash(2):ENDPROC
4050  IF hy%=1 ANDhx%=16 hx%=17:khk=17:
ENDPROC
4060  IF hy%=2 ANDhx%=20 hx%=21:khk=21:
ENDPROC
4070  IF hx%<0 hx%=0:khk=0:ENDPROC
4080  PRINTTAB(o%,hy%) " ";TAB(hx%,hy%)
CHR$224;CHR$225
4090  ENDPROC
4100  DEF PROCfire
4110  F%=FALSE
4120  IF t%-rc%<4*z ENDPROC
4130  rc%=t%:SOUND 0,1,5,5
4140  IF f=0 PROCfire_left

```

```

4150 IF f=1 PROCfire_right
4160 ENDPROC
4170 DEF PROCfire_left
4180 ey%=ay%:ex%=ax%-1
4190 IF ex%<0 ENDPROC
4200 eh%=hx%+1:PROChit_check1:IF F% EN
DPROC
4210 PRINTTAB(ex%,ey%) "-"
4220 j%=0:REPEAT
4230 sx%=ex%:ex%=ex%-1:j%=j%+1
4240 PROChit_check2:PRINTTAB(sx%,ey%
) " "
4250 IF ex%>=0 PRINTTAB(ex%,ey%) "-"
4260 UNTIL F% OR ex%<0 OR j%=10
4270 PRINTTAB(ex%,ey%) " "
4280 ENDPROC
4290 DEF PROCfire_right
4300 ey%=ay%:ex%=ax%+3
4310 IF ex%>31 ENDPROC
4320 eh%=hx%:PROChit_check1:IF F% ENDP
ROC
4330 PRINTTAB(ex%,ey%) "-"
4340 i%=0:REPEAT
4350 sx%=ex%:ex%=ex%+1:i%=i%+1
4360 PROChit_check2:PRINTTAB(sx%,ey%
) " "
4370 IF ex%<=31 PRINTTAB(ex%,ey%) "-"
4380 UNTIL F% OR i%=10 OR ex%>31
4390 PRINTTAB(ex%,ey%) " "
4400 ENDPROC
4410 DEF PROCchopper_right
4420 x%=0:e%=0
4430 o%=INT khh:khk=khk+.5*z:hx%=INT k
hh
4440 IF hx%=o% ENDPROC
4450 REM Crash check
4460 IF hy%=ay% ANDhx%+1=ax% PRINTTAB(
o%,hy%) " ":PROCcrash(8):ENDPROC
4470 i%=0:REPEAT
4480 i%=i%+1
4490 UNTIL i%>=w% OR (hy%=m(i%)+1 OR
hy%=m(i%)) AND hx%+1=n(i%)
4500 IF (hy%=m(i%)+1 OR hy%=m(i%)) AND
hx%+1=n(i%) PRINTTAB(o%,hy%) " ":PROCcr
ash(1):ENDPROC

```



```

4510 IF hy%=gy% ANDhx%+1=gx% PRINTTAB(
o%,hy%) " ":PROCcrash(2):ENDPROC
4520 IF hy%=1 ANDhx%=13 hx%=12:khk=12:
ENDPROC
4530 IF hy%=2 ANDhx%=9 hx%=8:khk=8:END
PROC
4540 IF hx%>30 hx%=30:khk=30:ENDPROC
4550 PRINTTAB(o%,hy%) " ";TAB(hx%,hy%)
CHR$231;CHR$232
4560 ENDPROC
4570 DEF PROChit_check1
4580 IF FNscreen(ex%,ey%)=32 ENDPROC
4590 IF ey%=hy% ANDex%=eh% PROCcrash(4
):F%=TRUE:ENDPROC
4600 G%=FALSE:H%=FALSE:FORi%=1TOw%
4610 IF(ey%=m(i%)+1 OR ey%=m(i%)) AN
D ex%=n(i%):G%=TRUE:z%=i%
4620 IF ey%=h(i%) AND ex%=g(i%) H%=T
RUE:z%=i%
4630 NEXT
4640 IF G% PROCcrash(3):F%=TRUE:ENDPRO
C
4650 IF H% PROCcrash(5):F%=TRUE:ENDPRO
C
4660 IF ey%=gy% AND ex%=gx% PROCcrash(
6):F%=TRUE:ENDPROC
4670 ENDPROC
4680 DEF PROChit_check2
4690 IF FNscreen(ex%,ey%)=32 ENDPROC
4700 IF ey%=hy% AND ex%=eh% PRINTTAB(s
x%,ey%) " ":PROCcrash(4):F%=TRUE:ENDPROC
4710 G%=FALSE:H%=FALSE:FORi%=1TOw%
4720 IF (ey%=m(i%)+1 OR ey%=m(i%)) A
ND ex%=n(i%) PRINTTAB(sx%,ey%) " ":G%=TRU
E:z%=i%
4730 IF ey%=h(i%) AND ex%=g(i%) PRIN
TTAB(sx%,ey%) " ":H%=TRUE:z%=i%
4740 NEXT
4750 IF G%=TRUE PROCcrash(3):F%=TRUE:E
NDPROC
4760 IF H%=TRUE PROCcrash(5):F%=TRUE:E
NDPROC
4770 IF ey%=gy% AND ex%=gx% PRINTTAB(s
x%,ey%) " ":PROCcrash(6):F%=TRUE
4780 ENDPROC

```

```

4790 DEF PROCdelay(T%)
4800 TIME=0:REPEAT UNTIL TIME>T%
4810 ENDPROC
4820 DEF FNscreen(X%,Y%)
4830 VDU31,X%,Y%:A%=135
4840 =( (USR(&FFF4)AND&FFFF)DIV&100)
4850 DEF PRO Ctune(T%)
4860 IF T%=1 RESTORE 5000
4870 IF T%=2 RESTORE 5010
4880 IF T%=3 RESTORE 5030
4890 IF T%=4 RESTORE 5060
4900 READ N%
4910 IF T%=4 FOR F%=1 TO 2
4920     FOR L%=1 TO N%
4930         READ P%,D%
4940         SOUND 1,1,P%,D%:SOUND 2,1,P%+
48,D%
4950     NEXT
4960     IF T%=4 N%=N%-2:RESTORE 5060:RE
AD D%:NEXT
4970 REPEAT UNTIL ADVAL(-6)=15
4980 ENDPROC
4990 REM *** Tune Data ***
5000 DATA 6,13,2,25,2,45,2,65,3,45,2,6
5,10
5010 DATA 11,25,12,25,8,25,4,25,12,37,
8,33,4
5020 DATA 33,8,25,4,25,8,21,4,25,12
5030 DATA 15,81,4,81,4,81,8,69,4,69,4,
69,8
5040 DATA 53,4,61,4,53,4,33,16
5050 DATA 41,4,41,8,49,4,49,8,53,8
5060 DATA 18,80,18,108,18,100,4,96,4,8
8,4
5070 DATA 128,18,108,9,100,4,96,4,88,4
5080 DATA 128,18,108,9,100,4,96,4,100,
4
5090 DATA 88,18,60,8,60,4
5100 DEF PROCpress
5110 PRINTTAB(2,21)"Press any key..."
5120 *FX 15,1
5130 D%=GET
5140 ENDPROC
5150 DEF PROCdefine(k%)
5160 IF k%=1 RESTORE 5270

```

```

5170 IF k%=2 RESTORE 5370
5180 IF k%=3 RESTORE 5450
5190 FOR chr%=224TO244
5200     VDU 23,chr%
5210     FOR bxt%=0TO7
5220         READ row%:VDU row%
5230     NEXT
5240 NEXT
5250 ENDPROC
5260 REM *** Character Data ***
5270 DATA 49,74,132,132,132,132,132,13
2,140,82,33,33,33,33,33,32,0,0,0,0,0,0,6
4,128
5280 DATA 56,69,131,131,131,131,69,56
5290 DATA 48,73,134,4,4,4,4,4,128,80,3
2,0,0,0,0,0,0
5300 DATA 49,74,132,132,132,132,132,13
2,128,64,32,32,32,32,40,16
5310 DATA 8,16,32,64,64,32,34,28
5320 DATA 31,2,4,4,4,2,2,1,248,0,0,0,0
,16,32,192,0,0,4,8,16,32,64,128
5330 DATA 60,66,129,128,128,129,66,60
5340 DATA 28,34,93,81,93,34,28,0
5350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0
5360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0
5370 DATA 255,195,195,195,195,195,195,195,
195,255,195,195,195,255,255,255,255,15,6
,6,31,63,127,255,255,0,0,0,255,255,255,2
55,255,0,0,0,248,252,254,255,255
5380 DATA 0,4,12,28,60,62,127,255,1,7,
15,15,7,3,1,0,255,255,255,255,255,255,25
5,127,128,192,192,224,192,192,128,0,60,2
8,24,24,24,24,56,60
5390 DATA 24,24,60,60,126,126,255,255
5400 DATA 127,127,127,127,127,127,255,
48,128,240,136,132,254,254,255,12
5410 DATA 0,0,15,16,112,127,255,48,0,0
,128,64,32,254,255,12
5420 DATA 0,24,60,60,60,24,24,24
5430 DATA 0,0,0,255,165,165,165,165
5440 DATA 0,24,60,36,126,66,66,255,15,
31,31,8,7,128,255,255,255,255,255,0,255,

```

0,255,255,240,248,248,16,224,1,255,255  
 5450 DATA 255,4,30,35,67,67,63,12,224,  
 0,0,1,195,255,254,0  
 5460 DATA 0,224,62,255,252,0,0,0,240,1  
 5,255,255,131,56,96,128,3,225,127,63,63,  
 1,2,7  
 5470 DATA 24,36,36,126,255,129,129,255  
 ,255,255,153,153,153,153,153,189  
 5480 DATA 7,0,0,128,195,255,127,0,255,  
 32,120,196,194,194,252,48  
 5490 DATA 0,3,124,255,63,0,0,0,15,240,  
 255,255,193,56,6,1,192,135,254,252,252,1  
 28,64,224  
 5500 DATA 0,0,0,126,0,0,0,0  
 5510 DATA 60,126,219,255,126,102,36,60  
 5520 DATA 60,90,126,60,82,137,133,133  
 5530 DATA 146,64,10,81,4,40,136,4,72,8  
 1,126,241,36,80,0,136,144,32,64,156,0,8,  
 136,68  
 5540 DATA 146,96,67,27,66,4,136,1,32,8  
 ,65,216,0,201,4,2  
 5550 DATA 130,68,40,56,88,20,66,129



# HUNCHBACK

By B.C. Wakelin

Hunchback is to be found in most arcades and it is one of the more unusual games. You control Quazimodo, champion bell ringer of all France, as he scurries across the roof of the abbey to rescue his true love, Esmerelda.

To complete each screen, pull on the bell rope and exit the screen to the right. Here is a table of what happens on each level of the game.

- 1 Arrows and rocks fly across the screen and spears are thrown upwards.
- 2 Same as level one plus a block appears in the middle of the screen over Quazimodo's head.
- 3 Same as level two except that another block appears on the screen and boiling oil falls instead of spears being thrown.
- 4 Same as level three except that another block appears on the right and both oil and spears have to be avoided.
- 5 Similar to level four except that the two left-most blocks now become wider.
- 6 The same as level five except that there is a rope that also must be pulled to finish this screen and release Esmerelda.

When you complete all this, Esmerelda begs poor Quazimodo to rescue her sister from a similar predicament. So, being a love-struck fool, Quazi goes through it all again. This time, however, the time bonus (bottom right of the screen) decreases faster. If it reaches zero then Quazi bites the dust!

Though this version of Hunchback may not offer all the varied levels of the arcade original, it still offers a satisfactory challenge for even the most skilled players.

This program is also an excellent demonstration of just how BBC BASIC can be. On most other computers this game in would be unbearably slow!

# Program Description

**50** Disable the ESCAPE key.

**60-70** Call the routines that set up the screen display and print instructions.

**80** Call the screen display and instructions.

**90** Set up the array variables.

**100** Initiate the high-score table.

**110** Define the sound envelopes.

**120-150** Define the user-defined characters.

**160** Initiate the variables.

**110-180** Set up the display arrays.

**190** Turn off the cursor, print the game screen.

**200** Print, decrease and check the time bonus.

**210** If Quazi is jumping then go to the 'jump update' routine on line 1280.

**220-230** Read the direction of movement from the keyboard.

**240** If jumping, then branch to the 'start jumping' routine.

**250-260** Check if level has been completed.

**270** Stop Quazimodo from walking off the screen.

**280** Check for Quazi being killed.

**290** If Quazi has not moved, bypass, then print line.

**300** Print Quazimodo.

**310-680** The control and crash checking code for the various missiles.

**690-710** Routine used when a level is finished. Add the bonus the score.

**720-770** Level 6 completed. Begin again with a harder skill level.

**780** Start of game-over routine.

**790** Page mode off.

**800** Check for a new high score.

**810** Print a new high score table.

**820-860** Offer the choice of a restart or the instructions.

**870** Procedure to display the title.

**880-930** Enter the new high score.

**940-1150** Instructions.

**1160-1190** Initiate the lances.

**1200-1210** Initiate the boiling oil.

**1220-1240** Print a new screen.

**1250-1260** Print soldiers.

**1270** Initiate the jump routine.

**1280-1330** Routine to calculate the movement of Quazimodo.

**1340** Printing of the special characters used on screen 2.

**1350** Printing of the special characters used on screens 3 and 4.

**1360** Printing of the special characters used on screen 5.

**1370** Printing of the special characters used on screen 6.

**1380-1390** Data for the death tune.

---

```
10  REM      ***  HUNCHBACK  ***
20  REM
30  REM ***  BY  B.  WAKELIN  ***
40  REM
50  *FX229,1
60  *FX213,150
70  *FX214,1
80  MODE7:VDU23;10,32;0;0;0:PROCINSTR
UCTIONS
90  MODE2:DIMM$(1),D%(1),E%(1),TP$(8)
,TP%(8)
100  FORI%=1TO8:TP%(I%)=500:TP$(I%)="P
CW Soft":NEXT
110  ENVELOPE1,1,50,-30,-10,1,3,-2,127
,-9,-4,-3,126,0:ENVELOPE2,1,0,0,0,0,0,0,
127,-3,-2,-1,126,0:ENVELOPE3,2,-40,50,-2
0,-10,5,2,127,-3,-2,-1,126,0:ENVELOPE4,2
,-30,-10,10,5,-5,10,127,-3,-2,-1,126,0
120  VDU23,224,255,4,4,4,255,32,32,32,
23,225,49,121,73,49,121,181,179,177,23,2
26,177,49,49,41,41,69,68,102,23,227,0,33
,99,254,99,33,0,0,23,228,0,44,126,124,62
,126,52,0,23,229,15,31,14,14,12,4,2,2,23
,230,240,248,32,144,8,112,64,128
130  VDU23,231,11,19,39,46,39,19,8,15,
23,232,96,112,112,20,84,80,16,224,23,233
,7,7,14,14,28,0,30,30,23,234,112,112,56,
56,112,0,60,60,23,235,11,27,59,57,60,30,
```



```

15,15,23,236,96,112,0,244,244,0,240,224,
23,237,0,1,3,27,26,24,25,1
140   VDU23,238,160,96,224,192,128,0,22
4,224,23,239,24,24,126,24,24,36,36,66,23
,240,16,56,56,124,124,0,254,56,23,241,8,
28,8,8,8,8,8,8,23,242,66,136,162,2,100,1
00,72,40
150   VDU23,243,8,84,228,211,193,251,25
2,64,23,244,0,60,254,1,122,122,128,127,2
3,245,0,0,3,3,251,251,251,3
160   L%=1:S%=0:LIV%=3:DEC%=2
170   M$(0)=CHR$229+CHR$230+CHR$8+CHR$8
+CHR$10+CHR$231+CHR$232+CHR$8+CHR$8+CHR$
10+CHR$233+CHR$234:M$(1)=CHR$229+CHR$230
+CHR$8+CHR$8+CHR$10+CHR$235+CHR$236+CHR$
8+CHR$8+CHR$10+CHR$237+CHR$238
180   D$=CHR$243+CHR$244+CHR$245:R$="
"+CHR$8+CHR$8+CHR$10+"   "+CHR$8+CHR$8+CH
R$10+"   ":A$=CHR$224+CHR$8+CHR$10:W$=LEF
T$(STRING$(10,A$),28)
190   VDU23,1,0;0;0;0;:PROCscreen:U%=0:
BS%=900:C%=0:O%=0:J%=0:V%=0:M%=0:X%=0:Y%
=18:RB%=0:RR%=512:WW%=-1:ER%=0:COLOUR2:P
RINTTAB(X%,Y%)M$(0)
200   COLOUR3:BS%=BS%-6:PRINTTAB(16,27)
"      ";TAB(16,27);BS%:IF BS%<0THEN360
210   IFJ%=1THEN1280
220   H%=0:IFINKEY(-98)THENVDU7:H%=-1
230   IFINKEY(-67)THENVDU7:H%=1
240   IFINKEY(-74)THENPROCjump
250   IFL%=6ANDER%=0THEN270
260   IFX%+H%=19ANDRB%=1THEN690
270   IFX%+H%<0ORX%+H%>18THENH%=0
280   IF(POINT(X%*64,(32-Y%-V%)*32-104)
=0ORPOINT(X%*64+96,(32-Y%-V%)*32-104)=0)
ANDV%=0THEN360
290   IFH%=0ANDV%=0THEN310
300   COLOUR2:M%=(M%+1)MOD2:X%=X%+H%:Y%
=Y%+V%:PRINTTAB(X%-H%,Y%-V%)R$;TAB(X%,Y%
)M$(M%)
310   IFC%=0THEN400
320   IFA%=0THENC%=0:PRINTTAB(0,B%) "   ":
GOTO350
330   IFPOINT(A%*64-32,(32-B%)*32-16)=2
THEN360

```

```

340  A%=A%-1:COLOURG%:PRINTTAB(A%+1,B%
) " ";TAB(A%,B%)CHR$C%
350  ON L%GOTO430,430,490,550,550,660:
GOTO200
360  LIV%=LIV%-1:PRINTTAB(X%,Y%)R$:COL
OUR2:X%=X%-1:IFX%<0THENX%=0
370  FORI%=Y%+1TO20:PRINTTAB(X%,I%)D$:
FORJ=1TO150:NEXT:PRINTTAB(X%,I%) "    ":NE
XT:PRINTTAB(X%,20)D$:SOUND0,2,6,1:FORI=1
TO1000:NEXT:RESTORE1390:FORI%=1TO8:READN
%,D%:SOUND1,-13,N%,D%:SOUND1,0,N%,1:NEXT
:FORI=1TO1000:NEXT
380  FORI%=1TO42:PRINTTAB(0,32):FORJ=1
TO100:NEXT:NEXT:CLS:IFLIV%=0THEN780
390  GOTO190
400  IFRND(20)>1THEN350
410  A%=19:B%=RND(2)+18:C%=RND(2)+226:
G%=1:IFC%=227THENG%=7
420  GOTO350
430  IFO%=1THEN450
440  IFRND(20)=1THENSOUND0,1,6,1:O%=1:
D%(0)=6:E%(0)=20:D%(1)=13:E%(1)=20ELSEGO
TO200
450  PROClance
460  IFEE%=1THENFORI%=0TO1:PRINTTAB(D%
(I%),E%(I%)+1) " ";TAB(D%(I%),E%(I%)) " ":
NEXT:O%=0:GOTO200
470  IFDD%=1THEN360
480  GOTO 200
490  IF U%=1 THEN520
500  IFRND(20)>1THEN200
510  SOUND2,3,50,1:COLOUR4:U%=1:P%=RND
(17)-1:Q%=1:PRINTTAB(P%,Q%)CHR$242
520  PROCoil:IFDD%=0THEN200
530  IF((POINT(64*P%+32,(32-Q%)*32-40
)=2)AND((P%=X%)OR(P%=X%+1)))THEN360
540  U%=0:PRINTTAB(P%,Q%) " ":GOTO200
550  IFO%=1THEN570
560  IFRND(20)=1THENSOUND0,1,6,1:O%=1:
D%(0)=6:E%(0)=20:D%(1)=13:E%(1)=20ELSEGO
TO600
570  PROClance
580  IFEE%=1THENFORI%=0TO1:PRINTTAB(D%
(I%),E%(I%)+1) " ";TAB(D%(I%),E%(I%)) " ":
NEXT:O%=0:GOTO600

```

```

590 IFDD%=1THEN360
600 IFU%=1THEN630
610 IFRND(20)>1THEN200
620 SOUND2,3,50,1:COLOUR4:U%=1:P%=RND
(17)-1:Q%=1:PRINTTAB(P%,Q%)CHR$242
630 PROCcoil:IFDD%=0THEN200
640 IF((POINT(64*P%+32,(32-Q%)*32-40
)=2)AND((P%=X%)OR(P%=X%+1)))THEN360
650 U%=0:PRINTTAB(P%,Q%) " ":GOTO200
660 IFWW%=-1THENGCOL0,0:MOVE608,RR%:DR
AW608,RR%+32:RR%=RR%+32:IFRR%=896THENWW
%=1
670 IFWW%=1THENGCOL0,6:MOVE608,RR%:DR
AW608,RR%-32:RR%=RR%-32:IFRR%=480THENWW%
=-1
680 GOTO550
690 COLOUR3:PRINTTAB(1,10)"LEVEL ";L%
;" COMPLETED."
700 FORI=1TO5000:NEXT:S%=S%+BS%:L%=L%
+1:IFL%=7THEN720
710 CLS:GOTO190
720 MODE2:COLOUR12:PRINTTAB(7,10)"HUR
RAY"'''TAB(8)"MADE":COLOUR11:PRINTTAB(6
,12)"QUAZIMODO"'''TAB(9)"IT"
730 RESTORE1380:FORI=1TO12:READN%,D%:
IFN%=0THENSOUND1,0,N%,D%:GOTO750
740 SOUND1,-13,N%,D%
750 NEXT
760 FORI=1TO5000:NEXT:DEC%=DEC%+2:IF
DEC%=18 THEN DEC%=16
770 COLOUR5:PRINT'"Esmeralda begs yo
u"'''to resue her sister!":FORI=1TO6000:
NEXT:MODE 5:L%=1:GOTO190
780 COLOUR1:PRINTTAB(5,15)"GAME OVER"
:FORI=1TO4000:NEXT:MODE7
790 *FX15
800 IFS%>TP%(8)THENPROChighscore
810 CLS:PROCtitle:FORI%=1TO8:PRINTTAB
(1,I%*2+2);I%;" ";STRING$(26,".");" ";TP
%(I%);TAB(6,I%*2+2)TP$(I%):NEXT
820 PRINT'"CHR$131"Press 'SPACE' to p
lay again or'"CHR$131"'I' for instructio
ns."
830 *FX15
840 Q%=GET:IFQ%=32THENMODE2:GOTO160

```

```

850  IFQ%<>73THENVVDU7:GOTO840
860  MODE7:VDU23;10,32;0;0;0:PROCINSTR
UCTIONS:MODE2:GOTO160
870  DEFPROCtitle:FORI%=0TO1:PRINTTAB(
7,I%)CHR$141CHR$129"Hunchback - By B.Wak
elin":NEXT:ENDPROC
880  DEFPROChighscore:PRINT'"CHR$136CH
R$129"NEW HIGH SCORE !!!'"CHR$134" Plea
se enter your name."
890  INPUT'N$:IFLEN(N$)>20THENVVDU7:PRI
NT'CHR$129"Too long, try again.":GOTO890
900  I%=9:REPEAT:I%=I%-1:UNTILI%=1ORTP
%(I%-1)>S%
910  FORJ%=8TOI%STEP-1:TP$(J%)=TP$(J%-
1):TP%(J%)=TP%(J%-1):NEXT
920  TP$(I%)=N$:TP%(I%)=S%
930  ENDPROC
940  DEFPROCINSTRUCTIONS:PROCtitle
950  PRINT'"CHR$130"      You have to hel
p Quazimodo across"
960  PRINTCHR$130"the castle ramparts
to save Esmerelda."
970  PRINTCHR$130"This is no easy job
as Esmerelda is"
980  PRINTCHR$130"guarded by soldiers
with bows and"
990  PRINTCHR$130"spears, rock hurling
giants and mean"
1000 PRINTCHR$130"oil droppers. To add
to his trouble"
1010 PRINTCHR$130"he has to avoid lump
s of stone above"
1020 PRINTCHR$130"his head, and get to
the end of each"
1030 PRINTCHR$130"level before his ene
rgy runs out!"
1040 PRINTCHR$133"      You control Quazi
modo using 'Z' and"
1050 PRINTCHR$133"'X' for left and rig
ht and 'RETURN' to"
1060 PRINTCHR$133"jump."
1070 PRINTCHR$134"      At the end of eac
h level you must"
1080 PRINTCHR$134"ring the bell and wa
lk to the edge of"

```

```

1090 PRINTCHR$134"the rampart to conti
nue."
1100 PRINTCHR$129"    To rescue Esmerel
da you must pull"
1110 PRINTCHR$129"the middle rope on l
evel 6."
1120 PRINT'CHR$131"PRESS 'SPACE' TO PL
AY. ";
1130 *FX15
1140 Q%=GET:IFQ%<>32THENVDU7:GOTO1140
1150 ENDPROC
1160 DEFPROC lance:COLOUR5:DD%=0:EE%=0:
FORI%=0TO1:E%(I%)=E%(I%)-1:IFE%(I%)=1THE
NEE%=1
1170     IFE%(I%)<Y%+3ANDE%(I%)>Y%-1ANDD
%(I%)>X%-1ANDD%(I%)<X%+2THENDD%=1
1180     PRINTTAB(D%(I%),E%(I%)+1) " ";TA
B(D%(I%),E%(I%))CHR$241
1190     NEXT:ENDPROC
1200 DEFPROC coil:DD%=0:Q%=Q%+1:IF(POINT
(64*P%+32,(32-Q%)*32-40))THEN DD%=1
1210 COLOUR1:PRINTTAB(P%,Q%-1) " ";TAB(
P%,Q%)CHR$242:ENDPROC
1220 DEFPROC screen:COLOUR3:PRINTTAB(6,
10)"LEVEL ";L%:COLOUR6:COLOUR129:FORI%=1
9TO0STEP-1:PRINTTAB(I%,21)W$;;FORJ=1TO15
0:NEXT:NEXT
1230 COLOUR128:PRINTTAB(6,10)SPC(8):ON
L%GOSUB1250,1340,1350,1350,1360,1370
1240 COLOUR128:GCOL0,3:COLOUR3:PRINTTA
B(17,2)CHR$240:MOVE1112,920:DRAW1112,512
:COLOUR7:PRINTTAB(0,0)"SCORE ";S%;TAB(10
,27)"BONUS 900":COLOUR2:PRINTTAB(13,0)ST
RING$(LIV%-1,CHR$239):ENDPROC
1250 COLOUR5:FORI%=5TO12STEP7:PRINTTAB
(I%,21) " ";CHR$225; " ";TAB(I%,22) " ";CHR
$226; " ":NEXT
1260 RETURN
1270 DEFPROC jump:J%=1:K%=0:V%=-1:SOUND
1,1,100,1:ENDPROC
1280 IFK%=5THENV%=0:J%=0:GOTO250
1290 K%=K%+1:IFK%<3THENV%=-1ELSEV%=1
1300 IFX%+H%>15ANDX%+H%<18ANDY%+V%<16A
NDRB%=0THENSOUND2,2,150,1:SOUND3,2,140,1
:S%=S%+500*L%:COLOUR7:RB%=1:PRINTTAB(6,0

```

```

);S%
1310  IFL%=6ANDER%=0ANDX%+H%=8ANDY%+V%=
16ANDRR%<513THENSOUND1,4,100,1:S%=S%+100
0:COLOUR7:COLOUR128:PRINTTAB(6,0);S%:ER%
=1
1320  IFPOINT((X%+H%)*64,(32-Y%-V%)*32-
8)=1ORPOINT((X%+H%)*64+96,(32-Y%-V%)*32-
8)=1THEN360
1330  GOTO250
1340  GOSUB1250:COLOUR6:COLOUR129:FORI%
=14TO16:PRINTTAB(9,I%)CHR$224:NEXT:RETUR
N
1350  GOSUB1340:FORI%=14TO16:PRINTTAB(2
,I%)CHR$224:NEXT:RETURN
1360  GOSUB1350:FORI%=14TO16:PRINTTAB(1
6,I%)CHR$224;TAB(10,I%)CHR$224;TAB(1,I%)
CHR$224:NEXT:RETURN
1370  GOSUB1360:COLOUR128:FORI%=14TO16:
PRINTTAB(9,I%) " ":NEXT:GCOL0,6:MOVE608,9
56:DRAW608,512:RETURN
1380  DATA89,4,89,4,109,4,109,4,117,3,1
21,3,117,3,121,3,117,6,0,3,129,6,137,3
1390  DATA89,6,89,6,97,4,101,6,89,6,89,
6,81,6,101,3

```

# HUNGER ALLEY

By N.S. Norat

No book of games would be complete without an appearance of the king of the arcade - PAC-MAN. In this version, you have to move around a maze gobbling up food pills and avoid poisoned skulls - unless you have just eaten a power pill. After eating one of these, you flash, change colour and are allowed a time of 500 units to devour the nasty skulls. You score 10 points for each pill eaten, 30 for each power pill and 50 for each skull.

The game has nine different coloured screens and you around by operating four different keys:

Z - Left  
X = Right  
\* = Up  
? = Down

Remember, speed is the key word in this game, so gobble up those pills as fast as you can!

## Program Description

**80** Disable ESCAPE key.

**90** Display the instructions.

**100-110** Set up the high-score table.

**180** Beginning of a new game loop.

**190** Join text and graphics cursor.

**200-240** Set up. **250** Call the game loop procedure.

**260** Turn off page mode.

**210** Clear the screen.

**280** Print the high score.

**290** Go to the beginning of a new game.

**300-380** Procedure to define the sound envelopes, variables, colour of the maze and characters.

**390** Procedure to draw the maze.

**400** Restore data for maze shape.  
**410-510** Draw the maze.  
**520-620** Data for drawing the maze.  
**630** Beginning of the procedure to print the score.  
**640-710** print the values to the screen.  
**720-870** Procedure for printing the white food dots on the screen, using the commands MOVE and VDU.  
**880-920** Procedure to print the Gobbler at X%, Y%.  
**930** Beginning of the main program loop.  
**950-1000** Read the keyboard and go to the relevant command procedures. Note that \*FX,210 turns the sound on and off.  
**1010** Various calls to check and update variables.  
**1020** Check if the end-of-game conditions have been fulfilled.  
**1030** If player is dead then end the game.  
**1040** Go to the next screen.  
**1050** Begin play again.  
**1060** Procedure for the end of the game.  
**1070-1160** Procedure to move left. Check whether player has hit an obstacle and call other procedures if this is the case.  
**1170-1270** As above, but for movement to the right.  
**1280-1370** As above, upward movement  
**1380-1460** As above, for downward movement.  
**1470-1580** Update the score and the time of the power pill.  
**1590-1610** Check for the appearance of mother skull.  
**1620-1800** Play a tune. The screen is cleared, variables are updated and a new screen is displayed.  
**1810-1860** Print mother skull onto the screen.  
**1870-1930** Procedure for the death of the Gobbler.  
**1940-2220** Routines to update and then print the high-score table.  
**2230-2270** Procedure that is called when an energy pill is eaten.  
**2280-2300** Check whether the current energy pill is still active.  
**2310-2650** Print the instructions.

---

```

10  REM  | | | | |  HUNGER ALLEY  | | | | |
20  REM  | | | | |           BY    | | | | |
30  REM  | | | | |  Anwar  Sidat  | | | | |
  
```



```

40  REM  |||||  Sadiq  Norat  |||||
50  REM  |||||  Soyab  Norat  |||||
60  REM  |||||  Conversion  |||||
70  REM  |||||  Adam  Ruddie  |||||
80  *FX 229,1
90  PROCINSTRUCTIONS
100  DIM H%(3),H$(3)
110  H$(1)="PCW Soft"
120  H$(2)="PCW Soft"
130  H$(3)="PCW Soft"
150  H%(1)=1000
160  H%(2)=800
170  H%(3)=400
180  REPEAT MODE5
190      VDU 5
200      PROCSETUP
210      PROCMAZE
220      PROCFOOD
230      PROCSCORE
240      PROCMAN
250      PROCINPUT
260      *FX 15,0
270      *FX 12,0
280      PROCTABLE
290      UNTILFALSE
300  DEFPROCSETUP
310  ENVELOPE1,10,0,0,0,0,0,0,127,-14,
-3,-2,126,0:ENVELOPE2,1,50,-30,10,100,-5
0,5,127,-5,-4,-2,126,0
320  BX%=FALSE:RA%=1: PL%=94:SCR%=0:CO
L%=1:S%=0:LIV%=3:K%=FALSE:BA=45
325  VDU19,1,COL%+1,0,0,0
330  VDU 19,2,COL%,0,0,0
340  VDU 23,240,&1C,&3E,&3B,&7E,&78,&3
E,&1F,&0E
350  VDU 23,254,&FF,&FF,&FF,&FF,&FF,&F
F,&FF,&FF
360  VDU 23,255,60,126,219,255,255,126
,36,60
370  VDU 23,239,0,0,0,24,24,0,0,0
380  ENDPROC
390  DEFPROCMAZE
400  RESTORE 520
410  GCOL 0,2
420  Q=0

```

```

430 REPEAT
440     Q=Q+1
450     READ X1,Y1,X2,Y2
460     FOR J=1 TO 2
470         MOVE X1,Y1:DRAW X2,Y2
480         X1=1200-X1:X2=1200-X2
490     NEXT J
500     UNTIL Q=31
510 ENDPROC
520 DATA 0,50,600,50,0,950,600,950,10
0,750,200,750,100,850,200,850
530 DATA 100,750,100,850,200,750,200,
850,100,650,200,650
540 DATA 0,450,300,450,0,550,300,550,
0,250,100,250
550 DATA 100,350,200,350,200,350,200,
150,100,150,300,150
560 DATA 400,850,400,950,400,550,400,
750,500,750,300,750
570 DATA 300,750,300,850,300,250,500,2
50,400,50,400,250
580 DATA 300,350,400,350,400,350,400,
450
590 DATA 500,850,600,850,600,850,600,
750
600 DATA 500,150,600,150,600,250,600,
350
610 DATA 600,350,500,350,500,350,500,
650
620 DATA 500,650,600,650,0,50,0,450,0
,550,0,950,200,650,300,650
630 DEFPROCSCORE
640 VDU 4,28,0,2,19,0
650 COLOUR 3:PRINTTAB(0,0)"SCORE "
660 PRINTTAB(0,1)"HIGH "
670 PRINTTAB(12,0);"TIME "
680 PRINTTAB(12,1);"LIVES "
690 PROCUPSCORE
700 VDU 5:GCOL0,3:MOVE 400,40:PRINT"S
CREEN ";RA%
710 ENDPROC
720 DEFPROCFOOD
730 X=25
740 GCOL 0,3
750 FOR Y=115 TO 915 STEP 100

```

```

760      REPEAT
770          MOVE X,Y:VDU 239
780          X=X+100
790          IF ( X=125 OR X=1025) AND Y=81
5 THEN X=X+100
800          IF (Y=415 OR Y=515 OR Y=615) A
ND X=525 THEN X=X+200
810          UNTIL X>1150
820          X=25
830          NEXT Y
840      GCOL 0,2
850      MOVE 25,315:VDU 239:MOVE 25,815:VDU 2
39
860      MOVE 1125,315:VDU 239:MOVE 1125,81
5:VDU 239
870      ENDPROC
880      DEFPROC MAN
890      VDU 5
900      X%=525:Y%=215
910      MOVE X%,Y%:GCOL 0,2:VDU 240
920      ENDPROC
930      DEFPROC INPUT
940      REPEAT
950          IF INKEY(-98) THEN PROCLEFT
960          IF INKEY(-67) THEN PROCRIGHT
970          IF INKEY(-73) THEN PROCUP
980          IF INKEY(-105) THEN PROCDOWN
990          IF INKEY(-17) THEN *FX210,1
1000         IF INKEY(-82) THEN *FX210,0
1010         PROCUPSCORE:PROCHECK:PROCHECK
PILL
1020         UNTIL PL%=0 OR LIV%=0
1030         IF LIV%=0 THEN VDU 22,7:PRINT "SCO
RE=";S%; " " :ENDPROC
1040         PROCNEWSSCREEN
1050         GOTO 940
1060         ENDPROC
1070         DEF PROCLEFT
1080         IF (POINT(X%-25,Y%)=2) AND ( POINT
(X%-25,Y%-1)=2 OR POINT(X%-25,Y%+1)=2) T
HEN ENDPROC
1090         IF POINT(X%-75,Y%-15)=3 THEN S%=S
%+10:SOUND 1,-15,128,1:PL%=PL%-1
1100         MOVE X%,Y%:GCOL 0,0:VDU 254
1110         IF POINT(X%-75,Y%-15)=2 THEN S%=S

```

```

%+30:SOUND 1,-15,20,1:PROCENERGYPILL
1120  X%=X%-100
1130  IF X%<=0 THEN X%=1125
1140  MOVE X%,Y%:GCOL 0,2:VDU 240
1150  IF POINT(X%,Y%-10)=1 ANDK%=FALSE
THEN PROCDIE ELSEIFPOINT(X%,Y%-10)=1 AND
K%=TRUE THEN S%=S%+50:SOUND0,-15,70,1:P
ROCCKECHKPILL
1160  ENDPROC
1170  DEF PROCRIGHT
1180  IF (POINT(X%+75,Y%)=2) AND( POINT
(X%+75,Y%-1)=2 OR POINT(X%+75,Y%+1)=2) T
HEN ENDPROC
1190  IF POINT(X%+125,Y%-15)=3 THEN S%=
S%+10:SOUND 1,-15,128,1:PL%=PL%-1
1200  MOVE X%,Y%:GCOL 0,0:VDU 254
1210  IF POINT(X%+125,Y%-15)=2 THEN S%=
S%+30:SOUND 1,-15,20,1:PROCENERGYPILL
1220  IF X%>=1100 THEN X%=-75
1230  X%=X%+100
1240  MOVE X%,Y%:GCOL 0,2:VDU 240
1250  IF POINT(X%,Y%-10)=1 ANDK%=FALSE
THEN PROCDIE ELSEIFPOINT(X%,Y%-10)=1 AND
K%=TRUE THEN S%=S%+50:SOUND0,-15,70,1:P
ROCCKECHKPILL
1260  PROCCKECHKPILL
1270  ENDPROC
1280  DEFPROCUP
1290  IF (POINT(X%,Y%+35)=2) AND(POINT(
X%+1,Y%+35)=2 OR POINT(X%-1,Y%+35)=2) TH
EN ENDPROC
1300  MOVE X%,Y%:GCOL 0,0:VDU 254
1310  IF POINT(X%+25,Y%+85)=2 THEN S%=S
%+30:SOUND 1,-15,20,1:PROCENERGYPILL
1320  IF POINT(X%+25,Y%+85)=3 THEN S%=S
%+10:SOUND 1,-15,128,1:PL%=PL%-1
1330  Y%=Y%+100
1340  MOVE X%,Y%:GCOL 0,2:VDU 240
1350  IF POINT(X%,Y%-10)=1 ANDK%=FALSE
THEN PROCDIE ELSEIFPOINT(X%,Y%-10)=1 AND
K%=TRUE THEN S%=S%+50:SOUND0,-15,70,1:P
ROCCKECHKPILL
1360  PROCCKECHKPILL
1370  ENDPROC
1380  DEFPROCDOWN

```

```

1390 IF (POINT(X%,Y%-65)=2) AND (POINT(
X%+1,Y%-65)=2 OR POINT(X%-1,Y%-65)=2) TH
EN ENDPROC
1400 IF POINT(X%+25,Y%-115)=3 THEN S%=
S%+10:SOUND 1,-15,128,1:PL%=PL%-1
1410 MOVE X%,Y%:GCOL 0,0:VDU 254
1420 IF POINT(X%+25,Y%-115)=2 THEN S%=S
%+30:SOUND 1,-15,20,1:PROCENERGYPILL
1430 Y%=Y%-100
1440 MOVE X%,Y%:GCOL 0,2:VDU 240
1450 IF POINT(X%,Y%-10)=1 AND K%=FALSE
THEN PROC DIE ELSE IF POINT(X%,Y%-10)=1 AND
K%=TRUE THEN S%=S%+50:SOUND 0,-15,70,1:P
ROC CHECKPILL
1460 PROC CHECKPILL
1470 DEF PROC UPSCORE
1480 VDU 4,23;8202;0;0;0;
1490 S$=STR$(S%):H$=STR$(H%(1))
1500 P=LEN S$:Q=LEN(H$)
1510 IF P<5 THEN S$="0"+S$:GOTO 1500
1520 IF Q<5 THEN H$="0"+H$:GOTO 1500
1530 COLOUR 1:PRINT TAB(6,0);S$
1540 PRINT TAB(6,1);H$
1550 IF K%=TRUE THEN PRINT TAB(17,0);TI
ME
1560 PRINT TAB(18,1);LIV%
1570 VDU 5
1580 ENDPROC
1590 DEF PROC CHECK
1600 IF RND(BA)<3 THEN PROC TRAPS
1610 ENDPROC
1620 DEF PROC NEWS SCREEN
1630 DATA 115,2,100,5,115,2,100,5,75,2
,60,3,60,2,75,3,75,2,60,3
1640 FOR I=1 TO 10
1650 READ NO%,LE%
1660 SOUND 1,1,NO%,LE%:NEXT
1670 RESTORE
1680 K%=FALSE:RA%=RA%+1:BA=BA-5
1690 S%=S%+150
1700 IF BA<=5 THEN BA=5
1710 CLG
1720 MOVE X%,Y%:GCOL 0,0:VDU 240
1730 X%=525:Y%=215:PL%=94
1750 COL%=COL%+1

```

```

1760 IF COL%>8 THEN COL%=0
1770 VDU 19,6,COL%,0,0,0
1780 PROCMAZE:PROCSCORE
1790 PROCFOOD:PROCMAN
1800 ENDPROC
1810 DEFPROCTRAPS
1820 P=RND(11)*100+50:Q=(RND(8)+1)*100
1830 IF P=150 OR P=550 OR P=650 OR P=1
050 THEN 1820
1840 IF POINT(X%,Y%)=2 THEN 1820 ELSE
IF POINT(P,Q)=3 THEN PL%=PL%-1
1850 MOVE P-25,Q+15:GCOL0,1:VDU 255
1860 ENDPROC
1870 DEFPROC DIE
1880 LIV%=LIV%-1
1890 MOVE X%,Y%:GCOL 0,0:VDU 254
1900 SOUND 0,2,6,1
1910 X%=525:Y%=215
1920 MOVE X%,Y%:GCOL 1,2:VDU 240
1930 ENDPROC
1940 DEFPROCTABLE
1950 VDU 22,7
1960 VDU 23;8202;0;0;0;0;0;
1970 PRINTTAB(8,5);CHR$141;CHR$134;"YO
UR SCORE WAS ";S$:PRINTTAB(8,6);CHR$141;
CHR$134;"YOUR SCORE WAS ";S$
1980 IF S%>H%(1) THEN H%(3)=H%(2):H$(3)=
H$(2):H%(2)=H%(1):H$(2)=H$(1):H%(1)=S%:Q
=1:GOTO 2610
1990 IF S%>H%(2) THEN H%(3)=H%(2):H$(3)=
H$(2):H%(2)=S%:Q=2:GOTO 2610
2000 IF S%>H%(3) THEN H%(3)=S%:Q=3:GOTO
2610
2010 PRINTTAB(7,1);CHR$129;CHR$157;CHR
$141;CHR$135;"TOP SCOFFERS ";CHR$156:
PRINTTAB(7,2);CHR$129;CHR$157;CHR$141;CH
R$135;"TOP SCOFFERS ";CHR$156
2020 PRINTTAB(3,9);CHR$147;
2030 FOR U=4 TO 34
2040 PRINTTAB(U,9)"|";
2050 NEXT
2060 PRINTTAB(3,15);CHR$147;
2070 FOR U=4 TO 34
2080 PRINTTAB(U,15)"|";
2090 NEXT

```

```

2100   FOR U=10 TO 14
2110       PRINTTAB(3,U);CHR$147;"|"
2120       PRINTTAB(33,U);CHR$147;"|"
2130   NEXT
2140   FOR X=1 TO 3
2150       PRINT TAB(5,X+10);CHR$(128+X);H
$ (X);TAB(16,X+10);".....";TAB(
28,X+10);H%(X)
2160   NEXT
2170   PRINTTAB(5,20);CHR$134;"Press spa
ce bar to continue."
2180   A$=GET$:IF A$=" " THEN ENDPROC
2190   IF A$="Q" THEN *FX210,1
2200   IF A$="S" THEN *FX210,0
2210   IF A$<>" " OR A$<>"Q" OR A$<>"S" T
HEN GOTO 2180 ELSE ENDPROC
2220   ENDPROC
2230   DEFPROCENERGYPILL
2240   PC%=13:K%=TRUE:VDU 19,2,13,0,0,0
2250   TIME=0
2260   SOUND0,-15,7,1
2270   ENDPROC
2280   DEFPROCCKECHKPILL
2290   IF TIME>500 THEN VDU4:COLOUR0:PRI
NTTAB(17,0);"      ":VDU5:K%=FALSE:VDU 19,2
,COL%,0,0,0
2300   ENDPROC
2310   DEFPROCINSTRUCTIONS
2320   VDU22,7
2330   PRINTTAB(11,2);CHR$129;CHR$157;CH
R$141;CHR$135;"HUNGER ALLEY      ";CHR$156
2340   PRINTTAB(11,3);CHR$129;CHR$157;CH
R$141;CHR$135;"HUNGER ALLEY      ";CHR$156
2350   VDU23;8202;0;0;0;0;
2360   PRINTTAB(9,12);"SOUND ON _ S"
2370   PRINTTAB(9,14);"SOUND OFF _ Q"
2380   A$=GET$:IFA$="S" THEN *FX210,0
2390   IFA$="Q" THEN *FX210,1
2400   IF A$<>"S" AND A$<>"Q" THEN 2380
2410   PRINTTAB(5,21);CHR$131;"DO YOU WA
NT INSTRUCTIONS ?"
2420   PRINTTAB(5,7);CHR$131;"This is an
";CHR$134;"A-MAZE-ING";CHR$131;"game"
2430   A$=GET$
2440   IF A$="N" THEN ENDPROC

```

```

2450 IF A$<>"Y" THEN 2430
2460 CLS
2470 PRINTTAB(11,2);CHR$129;CHR$157;CHR$141;CHR$135;"HUNGER ALLEY ";CHR$156
2480 PRINTTAB(11,3);CHR$129;CHR$157;CHR$141;CHR$135;"HUNGER ALLEY ";CHR$156
2490 PRINTTAB(0,5);CHR$134;"The idea of the game is to eat the food";CHR$134;"You get 10 points for each pill you eat";CHR$134;"You must avoid eating the poisoned"CHR$134"skulls"
2500 PRINTCHR$134;"However if you eat the flashing energy"CHR$134"pill you will gain energy to eat these"CHR$134;"skulls."CHR$134;"You get 30 points for each energy pill"CHR$134;"and 50 points for each skull."
2510 PRINTCHR$134;"You also get 150 points for each screen"CHR$134;"that you complete"
2520 PRINTTAB(3,21);CHR$131;"Press spacebar to continue"
2530 REPEAT UNTIL GET$=" ":CLS
2540 PRINTTAB(11,2);CHR$129;CHR$157;CHR$141;CHR$135;"HUNGER ALLEY ";CHR$156
2550 PRINTTAB(11,3);CHR$129;CHR$157;CHR$141;CHR$135;"HUNGER ALLEY ";CHR$156
2560 PRINTTAB(3,7);CHR$134;"Use the following keys to move:"
2570 PRINTTAB(13,10);CHR$133;"Z - Left"
2580 PRINTTAB(13,12);CHR$133;"X - Right";TAB(13,14);CHR$133;"* - Up";TAB(13,16);CHR$133;"? - Down"
2590 PRINTTAB(3,21);CHR$131;"Press spacebar to play"
2600 REPEAT UNTIL GET$=" ":CLS:ENDPROC
2610 PRINT TAB(5,8);CHR$141;CHR$133;"ITS A HIGH SCORE!!!!";TAB(5,9);CHR$141;CHR$133;"ITS A HIGH SCORE!!!!":PRINTTAB(4,14);CHR$129;"PLEASE ENTER YOUR NAME.":INPUT "H$(Q)"
2620 IF LEN H$(Q)>10 THEN CLS:GOTO2610
2630 IF H$(Q)=" " THEN 2610
2640 CLS

```





# DOGFIGHT

BY E.J. Barfield

Dogfight is an excellent 'shoot 'em up' type of game, but it is somewhat unusual because it involves two players playing against each other.

The game is an air battle between two planes - a dogfight, in fact. The aim is to score 10 hits against your opponent before you run out of fuel. Each player has a choice of either a jet or a biplane; the jet is faster but has a larger turning circle and consumes more fuel.

Having selected their planes the players must choose the amount ammunition they wish to take with them on their sortie; this should be a value between 10 and 99. You will also be asked if you wish to fly in a clear sky or one with clouds. Novice pilots may need the security of clouds because the opposing flyer cannot shoot down while you are hidden from view. The controls are:

PLAYER 1 (blue)

W=Left

E=Right

A=Fire

PLAYER 2 (black)

O=Left

P=Right

K=Fire

## Program Description

**50** Define the envelope for the shooting sounds.

**70** Dimension the arrays that are to be used within the program.

**80-90** Change the graphics mode and turn off the cursor.

**95** Print the game's instructions. `^w`

**100** Request information concerning the type of plane and the amount of ammunition.

**110** Change mode.

**120-230** Set the colours.

**240-430** Define the characters that make up the planes' shape

**440** Turn the cursor off.

**450-490** Draw the screen.  
**510** Initialize the variables.  
**530** Draw the clouds if this option has been selected.  
**540-510** Reset the colours.  
**580** Print the players' scores.  
**590-680** Read in the data for the direction of movement for the planes.  
**690-740** The main game loop. Test for key presses and move planes accordingly.  
**810-980** Print the number of the plane that has won. Wait for a new game to be selected.  
**1000-1160** PROCPLANE - move player one's plane.  
**1110-1330** PROCPLANE2 - move player two's plane.  
**1340-1500** PROCLOUD - draw a cloud.  
**1510-1650** PROCBOMB - move player one's bomb and check for a hit.  
**1610-1800** PROCBZ - move player two's bomb and check for a hit  
**1810-1810** PROCEXPLODEZ - flash player two's plane.  
**1880-1940** PROCEXPLODE - flash player one's plane.  
**1950-2020** PROCQUES ask user to enter type of plane, a of ammunition required and whether clouds are wanted.  
**2030-2060** Data for the planes' directions.  
**2010-2090** Error routine.  
**2100-2340** PROCINSTRUCT - the instructions for the game.

---

```

10 REM*****
20 REM          Dogfight          *
30 REM*    By E.J. Barfield    *
40 REM*****
50 ENVELOPE1,0,0,0,0,0,0,0,100,-5,0,-
5,100,0
60 ONERRORGOTO2070
70 DIMX(10),Y(10),x(10),y(10),dx(10),
dy(10)
80 MODE7
85 VDU19,1,2;0;0
90 VDU23;8202;0;0;0;

```

```

95  PROCINSTRUCT
100  PROCQUES
110  MODE2
120  REM*****SET  COLOURS*****
130  VDU19,0,6,0,0,0
140  VDU19,1,0,0,0,0
150  VDU19,2,7,0,0,0
160  VDU19,3,6,0,0,0
170  VDU19,4,4,0,0,0
180  VDU19,5,4,0,0,0
190  VDU19,6,7,0,0,0
200  VDU19,10,6,0,0,0
210  VDU19,11,6,0,0,0
220  VDU19,14,4,0,0,0
230  VDU19,15,6,0,0,0
240  REM*****DEFINE  CHARACTERS*****
250  VDU23,224,0,32,48,120,62,120,48,32
260  VDU23,225,0,1,126,62,62,14,14,2
270  VDU23,226,0,8,8,28,62,127,20,0
280  VDU23,227,0,64,63,62,62,56,56,32
290  VDU23,228,0,4,12,30,124,30,12,4
300  VDU23,229,0,32,56,56,62,62,63,64
310  VDU23,230,0,20,127,62,28,8,8,0
320  VDU23,231,0,2,14,14,62,62,126,1
330  VDU23,233,0,6,134,206,255,206,134,
6
340  VDU23,234,52,114,60,28,30,231,98,3
2
350  VDU23,235,8,8,127,127,28,8,28,62
360  VDU23,236,4,78,60,56,120,231,70,4
370  VDU23,237,0,96,97,115,255,115,97,9
6
380  VDU23,238,4,70,231,120,56,60,78,4
390  VDU23,239,62,28,8,28,127,127,8,8
400  VDU23,240,32,98,231,30,28,60,114,3
2
410  VDU23,241,128,128,224,245,183,254,
224,0
420  VDU23,242,15,255,255,255,109,255,2
55,60
430  VDU23,243,0,3,127,255,219,255,127,
0
440  VDU23;8202;0;0;0;
450  GCOL0,3:MOVE0,1024:MOVE1280,1024:P
LOT85,0,924:PLOT85,1280,924:MOVE0,100:MO

```

```

VE1280,100:PLOT85,0,0:PLOT85,1280,0
  460 GCOL0,15:MOVE0,924:DRAW1280,924:MO
VE0,100:DRAW1280,100
  470 MOVE100,50:MOVE100,10:PLOT85,500,5
0:PLOT85,500,10
  480 GCOL0,11:MOVE1180,10:MOVE1180,50:P
LOT85,780,10:PLOT85,780,50
  490 MOVE515,40:VDU5:GCOL0,0:PRINT"FUEL
":VDU4
  500 REM*****SET VARIABLES
  510 sc%=0:sc2%=0:c%=0:c2%=0:t=5:t2=1:X
1%=550:X2%=650:Y2%=527:Y1%=527:OX1%=1400
:OX2%=1300:OY1%=1227:OY2%=1227:ot=5:ot2=
1:cot1%=0:F1=200:F2=200
  520 REM*****CLOUDS***
  530 IF C=1 THENPROCLOUD(200+RND(200),
300+RND(50),150+RND(150),100+RND(50),3,1
5):PROCLOUD(850+RND(100),650+RND(50),20
0+RND(100),100+RND(75),3,15)
  540 VDU19,10,0,0,0,0
  550 VDU19,3,7,0,0,0
  560 VDU19,15,4,0,0,0
  570 VDU19,11,0,0,0,0
  580 COLOUR15:COLOUR130:PRINTTAB(1,1);a
%;:PRINTTAB(6,1);sc%;:COLOUR11:PRINTTAB(
17,1);a2%;:PRINTTAB(13,1);sc2%;
  590 REM*****READ IN DATA*****
  600 FORw=1 TO 8
  610   READX(w),Y(w)
  620   NEXT
  630 FORB=1TO8
  640   READx(B),y(B)
  650   NEXT
  660 FORB=1TO8
  670   READdx(B),dy(B)
  680   NEXT
  690 REM*****INPUT*****
  700 REPEAT
  710   IF INKEY(-55) THENt2=t2+T2:F2=F2
-FR2:IFt2>8+T2 THEN t2=1
  720   IF INKEY(-34) THENt=t+T:F1=F1-FR
1:IFt>8+T THEN t=1
  730   IF INKEY(-56) THENt2=t2-T2:F2=F2
-FR2:IFt2<1 t2=8+T2
  740   IF INKEY(-35) THENt=t-T:F1=F1-FR

```

```

1: IF t < 1 THEN t = 8 + T
   750   IF c% = 1 THEN PROC BOMB ELSE IF INKEY
Y(-66) AND a% > 0 THEN SOUND&10,1,6,10: co% = 0:
c% = 1: a% = a% - 1: ch% = 0: dx = x(t): dy = y(t): x% = X1
% + dx(t): y% = Y1% + dy(t): ox% = x%: oy% = y%: PROC B
OMB: VDU4: COLOUR15: PRINTTAB(1,1); a%; " "
   760   IF c2% = 1 THEN PROC B2 ELSE IF INKEY
(-71) AND a2% > 0 THEN SOUND&10,1,6,10: a2% =
a2% - 1: co2% = 0: c2% = 1: ch2% = 0: dx2 = x(t2): dy2 =
y(t2): x2% = X2% + dx(t2): y2% = Y2% + dy(t2): ox2%
= x2%: oy2% = y2%: PROC B2: VDU4: COLOUR11: PRINT
TAB(17,1); a2%; " ";
   770   PROC PLANE
   780   PROC PLANE2
   790   UNTIL sc% = 10 OR sc2% = 10 OR F1 <= 0
OR F2 <= 0
   800   REM*****END*****
   810   TIME = 0: REPEAT UNTIL TIME > 100
   820   MODE 6
   830   VDU23; 8202; 0; 0; 0;
   840   PRINTTAB(0,0)
   850   *FX15,1
   860   IF F2 <= 0 AND F1 <= 0 THEN win = 3: GOTO 91
0
   870   IF F1 <= 0 THEN win = 2: GOTO 920
   880   IF F2 <= 0 THEN win = 1: GOTO 920
   890   IF sc% = sc2% THEN win = 3: GOTO 910
   900   IF sc% > sc2% THEN win = 1 ELSE win = 2
   910   IF win = 3 THEN PRINTTAB(10,9) "It's a
draw ! !": GOTO 940
   920   PRINTTAB(10,9) "Player No."; win; " W
ins ! ! "
   940   PRINTTAB(0,16) "Press any key to re
start except 'N'.."
   950   *FX15,1
   960   G = GET
   970   IF G = ASC "N" THEN MODE 4: END
   980   RESTORE: PROC QUES: GOTO 110
   990   REM*****PROCEDURES*****
1000   REM*****DRAW FIRST PLANE***
1010   DEF PROC PLANE
1020   GCOL 3,4
1030   MOVE OX1%, OY1%: VDU5: VDU223 + ot + TY
1040   VDU4
1050   MOVE X1%, Y1%: VDU5: VDU223 + t + TY: VDU4

```

```

1060  OX1%=X1%:OY1%=Y1%
1070  ot=t
1080  X1%=X1%+(X(t)*S):Y1%=Y1%+(Y(t)*S)
1090  IFX1%<-30 THENX1%=1310
1100  IFX1%>1310 THEN X1%=-30
1110  IFY1%<75 THENY1%=930
1120  IFY1%>930 THENY1%=75
1130  F1=F1-FR
1140  IF F1<0 THEN F1=0
1150  GCOL0,0:MOVE100+(F1*2),10:DRAW100+
(F1*2),50
1160  ENDPROC
1170  REM****DRAW SECOND PLANE****
1180  DEFPROCPLANE2
1190  GCOL3,1
1200  MOVEOX2%,OY2%:VDU5:VDU223+ot2+TY2
1210  VDU4
1220  MOVEX2%,Y2%:VDU5:VDU223+t2+TY2:VDU
4
1230  OX2%=X2%:OY2%=Y2%
1240  ot2=t2
1250  X2%=X2%+(X(t2)*S2):Y2%=Y2%+(Y(t2)*
S2)
1260  IFX2%<-30 THENX2%=1310
1270  IFX2%>1310 THEN X2%=-30
1280  IFY2%>930 THENY2%=75
1290  IFY2%<75 THEN Y2%=930
1300  F2=F2-FR3
1310  IFF2<0 THENF2=0
1320  GCOL0,0:MOVE780+(F2*2),10:DRAW780+
(F2*2),50
1330  ENDPROC
1340  REM****DRAW CLOUD*****
1350  DEFPROCCLLOUD(X%,Y%,SX%,SY%,C1%,C2%
)
1360  VDU29,X%;Y%;
1370  L%=6+RND(8)
1380  MOVE0,0:MOVESX%+SX%/L%,0
1390  X3%=SX%+SX%/10:Y3%=0
1400  FORI=0TO6.3STEP0.1
1410      X%=SX%*COS(I)+SX%/L%*COS(I*L%)
1420      Y%=SY%*SIN(I)+SY%/L%*COS(I*L%)
1430      GCOL0,C1%
1440      MOVE32,12:PLOT85,X%,Y%
1450      MOVEX3%,Y3%

```

```

1460      GCOL0,C2%:DRAWX%,Y%
1470      X3%=X%:Y3%=Y%
1480      NEXT
1490      VDU29,0;0;
1500      ENDPROC
1510      REM****PLAYER 1'S BOMB*****
1520      DEFPROCBOBOMB
1530      GCOL3,4
1540      IFch%=1 THENPLOT69,ox%,oy% ELSE ch
%=1
1550      PLOT69,x%,y%
1560      ox%=x%:oy%=y%
1570      y%=y%+dy:x%=x%+dx
1580      IFx%<-30 THENx%=1310
1590      IFx%>1310 THENx%=-30
1600      IFy%<75 THENy%=930
1610      IFy%>930 THENy%=75
1620      IF(x%>X2%-12ANDx%<X2%+72)AND(y%>Y2
%-42ANDy%<Y2%+12) THEN PLOT69,ox%,oy%:PR
OCEXPLODE2:c%=0:ENDPROC
1630      co%=co%+1
1640      IFco%>12 THEN PLOT69,ox%,oy%:c%=0:
ENDPROC
1650      ENDPROC
1660      REM*****PLAYER 2'S BOMB*****
1670      DEFPROCBOB2
1680      GCOL3,1
1690      IFch2%=1 THENPLOT69,ox2%,oy2% ELSE
ch2%=1
1700      PLOT69,x2%,y2%
1710      ox2%=x2%:oy2%=y2%
1720      y2%=y2%+dy2:x2%=x2%+dx2
1730      IFx2%<-30 THENx2%=1310
1740      IFx2%>1310 THENx2%=-30
1750      IFy2%<75 THENy2%=930
1760      IFy2%>930 THENy2%=75
1770      IF(x2%>X1%-12ANDx2%<X1%+72)AND(y2%
>Y1%-42ANDy2%<Y1%+12) THEN PLOT69,ox2%,o
y2%:PROCEXPLODE2:c2%=0:ENDPROC
1780      co2%=co2%+1
1790      IFco2%>12 THEN PLOT69,ox2%,oy2%:c2
%=0:ENDPROC
1800      ENDPROC
1810      REM****PLAYER 2 HIT****
1820      DEFPROCEXPLODE2

```



```

1830 SOUND&10,-10,6,3
1840 FORT5=1TO20:VDU19,1,RND(16),0,0,0:
NEXT:VDU19,1,0,0,0,0
1850 sc%=sc%+1
1860 VDU4:COLOUR15:COLOUR130:PRINTTAB(6
,1);sc%;
1870 ENDPROC
1880 REM*****PLAYER 1 HIT*****
1890 DEFPROCEXPLODE
1900 SOUND&10,-10,6,3
1910 FORT5=1TO20:VDU19,4,RND(16),0,0,0:
NEXT:VDU19,4,4,0,0,0
1920 sc2%=sc2%+1
1930 VDU4:COLOUR11:COLOUR130:PRINTTAB(1
3,1);sc2%;
1940 ENDPROC
1950 REM*****QUESTIONS*****
1960 DEFPROCQUES
1970 CLS:PRINTTAB(5,10)CHR$131;"PLAYER
1 BI-PLANE OR JET?";:H=GET:IFH<>ASC"J"AND
H<>ASC"B"THEN1970ELSEIFH=ASC"B" THEN T=
.5:S=1:TY=9:FR=0.1:FR1=0.2:ELSET=.25:S=1.
5:TY=0:FR=0.15:FR1=.25
1980 CLS:PRINTTAB(5,10)CHR$129;"PLAYER
2 BI-PLANE OR JET?";:H=GET:IFH<>ASC"J"AND
H<>ASC"B"THEN1980ELSEIFH=ASC"B" THEN T2=
.5:S2=1:TY2=9:FR2=0.2:FR3=0.1:ELSET2=.25
:S2=1.5:TY2=0:FR2=0.25:FR3=.15
1990 CLS:PRINTTAB(7,10)CHR$131;"DO YOU
WANT CLOUDS?";:H=GET:IFH<>ASC"Y" AND H<>
ASC"N" THEN GOTO1990 ELSEIFH=ASC"Y"THENC
=1 ELSE C=0
2000 CLS:PRINTTAB(0,10)CHR$131"How much
ammo for player 1 (max=99)";:INPUTa%:IF
a%<10ORa%>99 THEN PRINTTAB(15,15)CHR$131
;"SILLY":TIME=T:REPEAT:UNTILTIME>T+100:G
OTO2000 ELSE A=a%
2010 CLS:PRINTTAB(0,10)CHR$129;"How muc
h ammo for player 2 (max=99)";:INPUTa2%:
IFa2%<10ORa2%>99 THEN PRINTTAB(15,15)CHR
$129;"SILLY";:TIME=T:REPEAT:UNTILTIME>T+
100:GOTO2010 ELSE A2=a2%
2020 ENDPROC
2030 REM*****DATA FOR DIRECTION***
2040 DATA18,0,13,13,0,18,-13,13,-18,0,-

```

```

13,-13,0,-18,13,-13
  2050 DATA40,0,28,28,0,40,-28,28,-40,0,-
28,-28,0,-40,28,-28
  2060 DATA60,-16,60,0,32,0,0,0,0,-16,0,-
30,32,-30,60,-30
  2070 MODE6:VDU19,0,4,0,0,0:REPORT
  2080 PRINT" at line ";ERL
  2090 END
  2100 DEFPROCINSTRUCT
  2110 VDU26,12
  2120 PRINTCHR$130;CHR$141;SPC13;"Dogfig
ht"CHR$130;CHR$141;SPC13;"Dogfight"
  2130 PRINT'"CHR$131"The game is an air
battle between two"CHR$131"planes - a d
ogfight.'"
  2140 PRINT'CHR$130"The aim is to score
10 hits against"CHR$130"your opponent b
efore you run out of"
  2150 PRINTCHR$130"fuel.'"
  2160 PRINT'CHR$131"You may pilot either
a bi-plane or jet."CHR$131"A jet is fas
ter, but consumes more fuel"CHR$131"and
has a larger turning circle.'"
  2170 PRINT'CHR$130"You may also choose
a cloudy or clear"CHR$130"sky, and the
amount of ammunition.'"
  2180 PRINT'"Player 1(blue)"TAB(10)"Play
er 2(black)"
  2190 PRINT"W = left"TAB(20);"O = left"
"E = right"TAB(20);"P = right"
  2300 PRINT"A = fire"TAB(20)"K = fire"
  2310 PRINT'SPC5"Press any key to contin
ue";
  2320 *FX15
  2330 A=GET
  2340 ENDPROC

```



# EULER'S TOURING KNIGHT

By Alun Hughes

For those of you who dislike the fast action games, here is a computerized version of an old chess problem. The idea is to visit every square on the board, once only, with a knight. Although there is an option to allow the computer to demonstrate the pattern of moves, you should find this problem much harder than it appears first sight.

When you run the program you will be presented with the chessboard and the option of either playing or watching the computer demonstration. Whichever option is chosen, you must then specify the starting position for the tow wound the board. When viewing the demonstration, all you have to do is to press the space bar for each move. When attempting the tour yourself you must enter the co-ordinates of each successive move. In both cases the knight is represented by a K and squares are coloured red if they have already been visited.

## Program Description

**35** Display instructions.

**40** Change mode.

**50** Define the character that will be used as a square on the board

**70-140** Data for the computer's view of the board.

**150-200** Initialize the variables.

**210** Turn off the cursor.

**220** Print the board.

**230-250** Select demonstration or player mode and then act accordingly.

**260-320** PROCBOARD - print the board on the screen.

**330-350** PROCPIECE - print the K at the correct place upon the board.

**360-550** PROCcomputersolution - comprising:

**310-390** Reset the screen.

**400** Get the initial position of the knight.

**430-520** The main loop for moving the knight.  
**560-610** PROCin - enter valid coordinates for the knight.  
**620-810** PROChumansolution comprising:  
**630-680** Reset the screen and set up the knight's initial posit}  
**700** Print the knight on the screen.  
**730** Get the knight's next position.  
**740** Check that the move is valid.  
**760** Check if the square has been previously visited.  
**770-780** Colour in the old square.  
**800** Calculate the player's score.  
**820-900** PROCinstructions a procedure to print instructions.

---

```
10  REM Eulers tour.
20  REM By Alun Hughes
30  REM Adapted by A.Ruddle
34  MODE 6
35  PROCinstructions
40  MODE2
50  VDU23,255,255,129,129,129,129,129,
129,255
60  REM FIRST READ IN DATA
70  DATA 17,16,8,20,10,11,12,22
80  DATA 25,3,0,1,2,7,4,5
90  DATA 26,27,24,29,30,6,28,13
100 DATA 9,19,32,21,18,23,15,14
110 DATA 49,48,40,45,42,31,44,54
120 DATA 50,35,57,33,34,39,36,37
130 DATA 58,59,56,61,62,63,60,38
140 DATA 41,51,52,53,43,55,47,46
150 DIM BOD%(63)
160 SCORE=0:WRONG=0
170 FOR A%=0TO63
180     READ BOD%(A%)
190     NEXT
200 DIM USED%(7,7)
210 VDU23;8202;0;0;0;
220 PROCBOARD:REM PRINT BOARD
230 PRINTTAB(0,30)"Computer demo?":REP
EAT:A$=GET$:UNTILA$="Y"ORA$="N"
240 IFA$="Y" THEN PROCcomputersolution
```

```

:PRINTTAB(0,0)"Press 'R' to restart":REPEAT:A$=GET$:UNTILA$="R":RUN
  250 PROC humansolution:PRINTTAB(0,0)"Press 'R' to restart":REPEAT:A$=GET$:UNTILA$="R":RUN
  260 DEFPROC BOARD
  270 CLS
  280 FOR A%=0 TO 7
  290   COLOUR 131:PRINT TAB(6,8+A%);STRING$(8,CHR$(255));:COLOUR 128:PRINT;" ";A%+1
  300   NEXT A%
  310 PRINT'TAB(6)"12345678"
  320 ENDPROC
  330 DEFPROC PIECE
  340 PRINT TAB(X%+6,Y%+8);"K"
  350 ENDPROC
  360 DEFPROC computersolution
  370 PRINTTAB(0,0)"Computer's solution"
  380 PRINTTAB(0,30)SPC(19);
  390 PRINTTAB(0,19)"Initial"
  400 PROC in
  410 PRINTTAB(0,19)SPC(7)
  420 PRINTTAB(0,30)"SPACE for next move"
"
  430 FORMove%=1 TO 63
  440   PROC PIECE
  450   COLOUR 128
  460   C%=BOD%((Y%*8)+X%)
  470   REPEAT:A$=GET$:UNTILA$=" "
  480   COLOUR 129:COLOUR 7
  490   PRINTTAB(X%+6,Y%+8);CHR$255
  500   X%=C%MOD 8
  510   Y%=C%DIV 8
  520   NEXT
  530 PROC PIECE
  540 PRINTTAB(0,30)"          SOLVED !"
"
  550 ENDPROC
  560 DEFPROC in
  570 REPEAT:PRINTTAB(0,20)SPC(40):INPUTTAB(0,20)"'X' position ",X%:UNTIL X%>0 AND X%<9
  580 REPEAT:PRINTTAB(0,20)SPC(40):INPUTTAB(0,20)"'Y' position ",Y%:UNTIL Y%>0 AND

```

DY%<9

```
590 X%=X%-1:Y%=Y%-1
600 PRINTTAB(0,20)SPC(20)
610 ENDPROC
620 DEFPROC humansolution
630 PRINTTAB(0,30)"Human Solution."
640 PRINTTAB(0,19)"Initial"
650 PROCin
660 PRINTTAB(0,19)SPC(7)
670 PROCPIECE
680 USED%(X%,Y%)=USED%(X%,Y%)+1:CORREC
T=1
690 REPEAT
700     PROCPIECE
710     X=X%:Y=Y%
720     COLOUR2:COLOUR128
730     PROCin
740     IF (NOT(((X%=X+2)OR(X%=X-2))AND(
(Y%=Y-1)OR(Y%=Y+1)))OR(((X%=X+1)OR(X%=X-
1))AND((Y%=Y-2)OR(Y%=Y+2)))) THEN730
750     USED%(X%,Y%)=USED%(X%,Y%)+1
760     IF USED%(X%,Y%)>1 THEN WRONG=WRO
NG+1:SCORE=SCORE-2:SOUND 1,-15,0,5 ELSE
SCORE=SCORE+2:CORRECT=CORRECT+1
770     COLOUR129:COLOUR7
780     PRINTTAB(X+6,Y+8);CHR$255
790     UNTILCORRECT=64
800 PRINTTAB(0,30)"          SOLVED !
":COLOUR7:COLOUR128:PRINTTAB(0,29)"Your
score=";SCORE/128*100" %"
810 ENDPROC
820 DEFPROC instructions
825 CLS:PRINTTAB(8,0)"Euler's Touring
Knight"'
830 PRINT "The idea of this game is t
o visit every square on a chess board wi
th a knight, visiting each square only
once. The knight makes normal chess
moves."
840 PRINT '"You can watch the computer
solve the problem or solve it yourse
lf."'
850 PRINT '"With either option you mus
t enter a starting point in the form
X co-ordinateY co-ordinate."
```

```

860 PRINT '"When attempting the tour y
ou must enter the co-ordinates of each s
uccessive moveThe knight is represented
by a 'k' and previously visited squares
are coloured red.'"
870 PRINT '"Your score is displayed at
the end of the game as a percentage o
f correct/incorrect moves"
880 PRINT:PRINT " PRESS SPACE TO
START";
890 REPEAT :get$=GET$:UNTIL get$=" "
900 ENDPROC

```



# MISSILE DEFENDER

By S. Roden

\*\*\* MESSAGE TO SECTOR A37 STOP ENEMY NUCLEAR MISSILES HEADING YOUR WAY STOP BEST OF LUCK STOP \*\*\*

Just your luck to be in command when the enemy decides to take some pot shots at yam sector! Missile Defender is a version of the classic arcade favourite Missile Command.

The object of the game is to protect your cities from an onslaught of IBMs (Intercontinental Ballistic Missiles, dummy, not computers from America!) The arcade original gave you three different bases from which to launch you missiles but, due to defence budget cuts, you have just one!

You score 25 points for each missile zapped. Every 1000 points you score gives you a new sector of four cities to protect. Each incoming missile is capable of wiping out a city and when all the cities are destroyed, the game is over. Your controls are:

Z = Left  
X = Right  
\* = Up  
? = Down

Happy blasting!

## Program Description

**10** Set up high-score Hall of Fame.

**50** Print the instructions

**60** Define the characters and initialize some variables.

**70-120** Reset some variables

**140** Enter game mode.

**150** Turn off the cursor.

**160** Set target position variables.

**170** If there are no missiles moving, then set up some new missiles; otherwise move missiles.

**180** If joystick has been selected then test; otherwise test the keyboard.

**190** If the game has ended then go to the Hall of Fame procedure and restart the game

**200** If a bonus is due to the player, then go to the bonus procedure.

**210** Return to the game loop

**220-300** PROCSETUP - initialize some variables and define some user-defined characters.

**310** PROCSCREEN - draw a new screen.

**320** Plot some stars on the screen  
**0-380** Draw the missile base.

**330-380** Draw the missile base.

**390-520** PROCJMOVE - test the joystick port and move the missile sights accordingly.

**530-620** PROCFIRE - give the fire sound, display the shot and check for hits. If hit then go to the hit procedure.

**630-640** PROCA - missile destroyed by player.

**650-740** PROCMISSILES - create new missiles, new screen and redraw cities. Also call the missile direction procedure.

**750-810** PROCB - procedure to define missile direction.

**820-870** PROCMOVEMISS - update the position of the missile and print the missiles.

**880-980** PROCC - print the missiles on the screen.

**990-1060** PROCEXP - make explosion sound and draw explosion on the screen.

**1070-1250** PROCFAME - Hall of Fame update and print procedure.

**1280-1340** PROCCITIES - print remaining cities on screen.

**1350-1400** PROCHECK - check for missile striking city.

**1410-1510** PROCMOVE check the keyboard and move the sights accordingly.

**1520-1740** PROCINTRO - print instructions and get choice of keyboard or joystick.

**1750** Subroutine to check if ESCAPE key is pressed.

**1760-1850** PROCBONUS - procedure to give new sector, score bonus and print cities.

**1860-1900** PROCCOLOURS - redefine logical colours.

```

B$(A%)="P.C.W. Soft":C(A%)=500:NEXT
  30  ON ERROR GOTO 1750
  40  MODE7
  50  PROCINTRO
  60  PROCSETUP
  70  FORA%=0TO3:R(A%)=1:NEXT
  80  SCORE=0
  90  VDU23;10,32,0;0;0;
100  CITIES=4:CLS
110  MISSILES=0
120  SOM=24:SOS=32
140  MODE2
150  VDU23;10,32;0;0;0
160  X=650:Y=X
170  IF MISSILES>0THENPROCMOVEMISS ELS
E  PROCMISSILES:COLOUR7:PRINTTAB(0,0);"SC
ORE=";SCORE
  180  IFJ$="Y"THENPROCJMOVE ELSEPROCMOV
E
  190  IF CITIES=0THENMODE6:PROCFAME:PRO
CINTRO:GOTO70
  200  IFBSCORE>=1000THENPROCBONUS:MISSI
LES=0:GOTO140
  210  GOTO170
  220  DEF PROCSETUP
  230  SCREEN=0:BSCORE=0
  240  BONUS=0
  250  SCORE=0
  260  VDU23,224,255,255,255,255,255,255
,255,255
  270  VDU23,226,129,66,36,0,0,36,66,129
  280  VDU23,225,100,101,105,110,100,255
,255,255
  290  CLS
  300  ENDPROC
  310  DEF PROCSCREEN
  320  CLS:FORW%=1TO50:PLOT69,RND(1280),
RND(1024):NEXT
  330  COLOUR2
  340  PRINTTAB(0,31);CHR$224;:FORA%=1TO
19:PRINTCHR$224;:NEXT
  350  COLOUR3
  360  GCOL0,2
  370  PLOT69,540,64:PLOT69,640,100:PLOT
85,740,64
110

```

```

380  ENDPROC
390  DEF PROCJMOVE
400  VDU5
410  GCOL0,0:MOVEX,Y:PRINTCHR$226:IF(A
DVAL(0) AND3)=1THENPROCFIRE
420  XX=ADVAL(1) DIV256
430  YY=ADVAL(2) DIV256
440  X=X+((XX>205)-(XX<55))*SOS
450  Y=Y+((YY<55)-(YY>205))*SOS
460  IFX>1249 X=1
470  IFX<-20 X=1248
480  IFY>1024 Y=1024
490  IFY<150 Y=150
500  GCOL0,7:MOVEX,Y:PRINTCHR$226
510  VDU4
520  ENDPROC
530  DEF PROCFIRE
540  ENVELOPE3,7,2,1,1,1,1,1,121,-10,-
5,-2,10,120
550  SOUND&100,3,2,50
560  GCOL0,3:MOVE650,100:DRAWX+31,Y-13
:GCOL0,0:MOVE650,100:DRAWX+31,Y-13
570  X%=X+31:Y%=Y-13
580  IFMR1=1THENMX=MX1:MY=MY1:MR=MR1:P
ROCA:MR1=MR
590  IFMR2=1THENMX=MX2:MY=MY2:MR=MR2:P
ROCA:MR2=MR
600  IFMR3=1THENMX=MX3:MY=MY3:MR=MR3:P
ROCA:MR3=MR
610  IFMR4=1THENMX=MX4:MY=MY4:MR=MR4:P
ROCA:MR4=MR
615  *FX15
620  ENDPROC
630  DEF PROCA:IFMX<Y%+13ANDMX>Y%-13 A
NDMY<X%+13ANDMY>X%-13 THENPROCXP:SCORE=
SCORE+25:MR=0:BSCORE=BSCORE+25
640  ENDPROC
650  DEF PROCMISSILES
660  IFBONUS>0THENPROCCOLOURS ELSEVDU2
0
670  MISSILES=0:MR1=1:MR2=1:MR3=1:MR4=
1
680  SOM=SOM+1:SOS=SOS+1:PROCSCREEN:PR
OCCITIES
690  IFSOM>31THENSOM=31:SOS=35

```

```

700  PROCB:MX1=MX:MY1=MY:A1=A
710  PROCB:MX2=MX:MY2=MY:A2=A
720  PROCB:MX3=MX:MY3=MY:A3=A
730  PROCB:MX4=MX:MY4=MY:A4=A
740  ENDPROC
750  DEF  PROCB:A=RND(3)-RND(4)
760  MX=1024
770  MY=RND(1280)
780  PLOT69,MY,MX
790  MY=MY-.5
800  MISSILES=MISSILES+1
810  ENDPROC
820  DEF  PROCMOVEMISS
830  IFMR1=1THENMX=MX1:MY=MY1:A=A1:PRO
CC:MX1=MX:MY1=MY:A1=A
840  IFMR2=1THENMX=MX2:MY=MY2:A=A2:PRO
CC:MX2=MX:MY2=MY:A2=A
850  IFMR3=1THENMX=MX3:MY=MY3:A=A3:PRO
CC:MX3=MX:MY3=MY:A3=A
860  IFMR4=1THENMX=MX4:MY=MY4:A=A4:PRO
CC:MX4=MX:MY4=MY:A4=A
870  ENDPROC
880  DEF  PROCC:GCOL0,6:PLOT69,MY,MX
890  MX=MX-SOM
900  MY=MY+A*5
910  DRAWMY,MX
920  IFMY>1249THENMY=0
930  IFMY<0THENMY=1249
940  IFMX>64ANDMX<110THENPROCHECK
950  IFMX<60THENMISSILES=MISSILES-1
970  GCOL0,3
980  ENDPROC
990  DEF  PROCEXP
1000  MR=0
1010  ENVELOPE1,130,80,-4,0,2,40,50,1,-
1,1,0,1,0:ENVELOPE2,2,0,0,0,0,0,0,60,-1,
-10,0,126,60
1020  GCOL0,1:SOUND&13,0,10,255:SOUND&1
10,2,7,255:SOUND&111,1,10,50:Q=1:FORB=1T
O20:PLOT69,MY-10+RND(50),MX-10+RND(50):P
LOT85,MY-10+RND(50),MX-10+RND(50):GCOL0,
Q
1030  IFQ=1THENQ=3ELSEQ=1
1040  NEXT
1050  MISSILES=MISSILES-1

```

```

1060  ENDPROC
1070  DEF PROCFAME
1080  CLS
1090  PRINTCHR$141;CHR$129;"  MISSILE D
EFENDER HALL OF FAME      "
1100  PRINTCHR$141;CHR$129;"  MISSILE D
EFENDER HALL OF FAME      "
1110  *FX15,1
1120  IFC(1)>SCORE THEN 1190
1130  PRINTTAB(0,10);"PLEASE ENTER YOUR
NAME"
1140  INPUT,A$
1150  A%=11:REPEAT:A%=A%-1:IFSCORE>C(A%
) THEN1170
1160      UNTILA%=0
1170  D=A%:IFD>10THEND=10
1180  FORF=2TOD:C(F-1)=C(F):B$(F-1)=B$(
F):NEXT:B$(D)=A$:C(D)=SCORE
1190  CLS:PRINTCHR$141;CHR$129"  MISSIL
E DEFENDER HALL OF FAME      ":PRINTCHR$14
1;CHR$129"  MISSILE DEFENDER HALL OF FAM
E      "
1200  FORA%=10TO1STEP-1:PRINTC(A%);"...
.";B$(A%):PRINT:NEXT
1210  VDU23;10,32,0;0;0;
1220  BONUS=0:BSCORE=0
1230  PRINT'"  PRESS SPACE BAR OR FIRE T
O CONTINUE";
1240  A$=INKEY$(0):IFADVAL(0)=1OR A$="
" THENENDPROC ELSE1240
1280  DEF PROCCITIES
1290  IFR(0)=1THENPRINTTAB(3,29);CHR$22
5
1300  IFR(1)=1THENPRINTTAB(6,29);CHR$22
5
1310  IFR(2)=1THENPRINTTAB(13,29);CHR$2
25
1320  IFR(3)=1THENPRINTTAB(16,29);CHR$2
25
1330  COLOUR3
1340  ENDPROC
1350  DEF PROCHECK
1360  IFMY>196ANDMY<266ANDR(0)=1THENR(0
)=0:PROCEXP:CITIES=CITIES-1:MISSILES=MIS
SILES-1

```

```

1370 IFMY>388ANDMY<458ANDR(1)=1THENR(1
)=0:PROCEXP:CITIES=CITIES-1:MISSILES=MIS
SILES-1
1380 IFMY>834ANDMY<904ANDR(2)=1THENR(2
)=0:PROCEXP:CITIES=CITIES-1:MISSILES=MIS
SILES-1
1390 IFMY>1027ANDMY<1097ANDR(3)=1THENR
(3)=0:PROCEXP:CITIES=CITIES-1:MISSILES=M
ISSILES-1
1400 MR=0
1410 DEF PROCMOVE
1420 VDU5
1430 GCOL0,0:MOVEX,Y:PRINTCHR$226:IF I
NKEY(-99)THENPROCFIRE
1440 X=X+(INKEY(-98)-INKEY(-67))*SOS
1450 Y=Y+(INKEY(-105)-INKEY(-73))*SOS
1460 IFX>1280 X=1
1470 IFX<-20 X=1279
1480 IFY>1024 Y=1024
1490 IFY<150 Y=150
1500 GCOL0,7:MOVEX,Y:PRINTCHR$226
1510 VDU4:ENDPROC
1520 DEF PROCINTRO
1530 VDU23;10,32,0;0;0;
1540 CLS
1550 PRINTSPC(9);CHR$141;CHR$129;"MISS
ILE DEFENDER"
1560 PRINTSPC(9);CHR$141;CHR$129;"MISS
ILE DEFENDER"
1570 PRINT'" As the defender of the l
ast laser base on earth, your mission is
to destroy the incoming missiles before
they destroy the four cities which you
are defending."'
1580 PRINT" You must destroy the missi
les by aimingthe sights at the head of t
he missiles which arrive in groups of f
our and by firing your laser."
1590 PRINT'" If you are not using joy-
sticks, use the following keys:"
1600 PRINT
1610 PRINTSPC(15);"Z - Left"
1620 PRINTSPC(15);"X - Right"
1630 PRINTSPC(15);"* - Up"
1640 PRINTSPC(15);"? - Down"

```

```

1650 PRINTSPC(11);"SPACE - Fire"
1660 PRINT'CHR$136;" PRESS SPACE-BAR O
R FIRE TO CONTINUE";
1710 TIME=0:REPEATUNTILTIME>20
1720 A%=(ADVAL(0) AND3):IFA%=1THENJ$="
Y":ENDPROC
1730 S$=INKEY$(0):IFS$<>" " THEN 1720
1740 J$="N":ENDPROC
1750 MODE7:IF ERR<>17 REPORT:PRINT" at
line ";ERL:END ELSE END
1760 DEF PROCBONUS
1770 *FX21,5
1780 FORA%=0TO200STEP5:SOUND1,-10,A%,3
:VDU19,6,5,0,0,0:VDU19,2,4,0,0,0:VDU19,3
,1,0,0,0:SOUND1,-10,220-A%,3:VDU20:NEXT
1790 IFBONUS>0 BONUS=0 ELSEBONUS=1
1800 PRINT''''
1810 CLS:PRINT''':FORA%=1TOCITIES:PRI
NTCHR$225;" ";:NEXT
1820 PRINT" X 100 =";CITIES*100:SCORE=
SCORE+CITIES*100:BSCORE=CITIES*100:CITIE
S=4
1830 TIME=0:REPEATUNTILTIME=250
1840 FORA%=0TO3:R(A%)=1:NEXT
1850 ENDPROC
1860 DEF PROCCOLOURS
1870 VDU19,6,5,0,0,0
1880 VDU19,2,4,0,0,0
1890 VDU19,3,1,0,0,0
1900 ENDPROC

```



# BLACK HOLES

By Freda Farrow

Black Holes is not a high-speed action game, but a game of skill and strategy with a little bit of luck thrown in for good measure.

Twenty-five aliens appear at the bottom of the screen, one at a time and in one of five colours. The colour of each is chosen at random and your task is to store these creatures in the black holes. Each black hole is identified by a letter, and you choose them by pressing the appropriate keys on the keyboard.

The first four aliens that appear must be placed in the four corners of the grid. After that they may be placed anywhere on the screen, providing the following rules are obeyed:

- 1 Each alien must be placed next to another one.
- 2 Aliens of the same colour must not be placed next to each other.
- 3 Aliens must be stored before they cross the screen, otherwise the game is lost.

## Program Description

**50** Sets ON ERROR.

**60** Sets first colour duration to one and a half seconds.

**70** Sets second colour duration to one and a half seconds.

**80** Change mode and reprogram display character.

**90** Dimension arrays.

**110** Define characters and strings.

**130** Change mode and turn the cursor off.

**140** Set up the variables for a new game.

**150-220** The main game loop.

**230-250** Reset computer for normal use.

**280-380** PROCINIT - define the characters used in the game.

**400-550** PROCHEADER - print the instructions.

**510-640** PROCSHUFFLE - shuffle the order in which the aliens appear.

**660-760** PROCSETUP - print the black holes onto the screen.  
**720-780** Data for the positions of the black holes on the screen.  
**800-940** PROCPLAY, comprising:  
**800-830** Move the alien and check whether the edge of the screen has been reached.  
**840-810** Determine which hole the alien is to be placed in.  
**890** Check that the first four aliens are placed into the corners.  
**910** Check that the hole is not occupied.  
**960-1020** FROGMAN - move the alien and check that it has not reached the edge of the screen  
**1040-1120** PROCCAUGHT - flash the alien when it reaches the edge of the screen  
**1140-1210** PROCPLACE - print the alien in its hole and check for the end of the game.  
**1230-1420** PROCMATE - check that the alien is correctly placed within the grid.  
**1440-1580** PROCFIN - print the correct reason for the end of the game. Enquire if the player would like to play again.  
**1600-1620** PROCWAIT - delay loop.

---

```

10  REM  ****
20  REM  ***  BLACK  HOLES  ***
30  REM  ***  FRED  PERROW  ***
40  REM  ****
50  ON ERROR GOTO 230
60  *FX9,75
70  *FX10,75
80  MODE1:VDU23;8202;0;0;0;
90  DIMA(25),H(25),V(25),PLACE(25)
100 X=RND(-TIME):*FX11,0
110 PROCINIT:PROCHEADER
120 REPEAT
130     MODE2:VDU23;8202;0;0;0;
140     PROCSETUP:A=1:D=0:SC=0
150     REPEAT
160         COLOURA(A)
170         I=0:*FX15,0
180         PROCPLAY
190         IF A=25 THEN D=5 ELSE A=A+1

```

```

200      UNTILD<>0
210      PROCFIN(D)
220      UNTIL FIN%
230      CLEAR:VDU7:MODE6:*FX12,0
240      *FX9,25
250      *FX10,25
260      END
270      :
280      DEFPROCINIT
290      VDU23,224,&18,&04,&07,&0D,&0F,&06
, &02,&02
300      VDU23,225,&18,&20,&E0,&B0,&F0,&60
, &40,&40
310      VDU23,226,&FF,&FF,&FF,&FF,&FF,&FF
, &FF,&FF
320      VDU23,227,&FF,&FE,&FC,&F8,&F0,&E0
, 0,0
330      VDU23,228,0,0,&07,&0F,&1F,&3F,&7F
, &FF
340      VDU23,229,0,0,&E0,&F0,&F8,&FC,&FE
, &FF
350      VDU23,230,&FF,&7F,&3F,&1F,&0F,&07
, 0,0
360      HOLE$=CHR$230+CHR$227+CHR$11+CHR$
8+CHR$8+CHR$226+CHR$226+CHR$11+CHR$8+CHR
$8+CHR$228+CHR$229+CHR$10+CHR$10
370      MAN$=CHR$224+CHR$225:FIN%=FALSE
380      ENDPROC
390      :
400      DEFPROCHEADER
410      COLOUR0:COLOUR135:CLS
420      PRINTTAB(4,7);"BLACK-HOLES";TAB(8
,9)"BY";TAB(26,7);"FREDA";TAB(26,9)"PERR
OW":VDU5:GCOL0,1
430      Y$=STRING$(37,".")
440      FORX=1TO37:MOVEX*32,(300*SIN(X/6)
)/2+800:PRINTMID$(Y$,X,1):SOUND1,-X/2,50
+X*3,1:NEXT:VDU4:*FX15,0
450      FOR I=22TO38 STEP6:SOUND0,-15,4,1
:PRINTTAB(I,4)HOLE$:PROCWAIT(1):NEXT
460      PROCWAIT(1)
470      FOR I=2TO18 STEP6:SOUND0,-15,4,1:
PRINTTAB(I,12)HOLE$:PROCWAIT(1):NEXT
480      PRINTTAB(0,16)"As the aliens app
ear at the bottom of the screen they mu

```

st be caught and put in the black holes  
. The first four must be placed in the  
corners, that is in holes A,J,P and Y.  
After this you must"

490 PRINT"put them along side each o  
ther, in any direction. The problem is  
that no two adjacent aliens can be the  
same colour."

500 PRINT"Nor can you put one out on  
its own. See how many you can catch, b  
ut don't let one escape as it will rescu  
e the others!"

505 PRINT "To place an alien in a hol  
e press the corresponding key"

510 COLOUR2

520 PRINTTAB(8,30)"PRESS ANY KEY TO S  
TART"

530 S\$=GET\$

540 CLG

550 ENDPROC

560 :

570 DEFPROC SHUFFLE

580 FOR L%=0TO4:FOR J%=1TO5

590 A(L%\*5+J%)=L%+1:NEXT:NEXT

600 FOR L%=1TO100

610 A%=RND(25):B%=RND(25)

620 T%=A(A%):A(A%)=A(B%):A(B%)=T%

630 NEXT

640 ENDPROC

650 :

660 DEFPROC SETUP

670 COLOUR134:CLS:RESTORE

680 FOR I=1 TO 25:READ H(I),V(I)

690 COLOUR0:COLOUR134

700 PRINTTAB(H(I),V(I))HOLE\$

710 COLOUR15:COLOUR128

720 PRINTTAB(H(I),V(I)-1)CHR\$(64+I)

730 PLACE(I)=0

740 NEXT

750 COLOUR134:PROC SHUFFLE

760 ENDPROC

770 DATA 9,3,6,6,9,6,12,6,3,9,6,9,9,9  
,12,9,15,9,0,12,3,12,6,12,9,12,12,12,15,  
12,18,12

780 DATA 3,15,6,15,9,15,12,15,15,15,6

```

,18,9,18,12,18,9,21
790      :
800      DEFPROCPLAY
810      REPEAT
820          REPEAT
830          PROCMAN:IF D<>0 THEN UNTIL TR
UE:UNTIL TRUE:ENDPROC
840      COLOUR7:PRINTTAB(1,22)"WHICH"TAB(
1,24)"BLACK-HOLE - ? "
850      *FX 15,0
860      X=INKEY(150)
870      UNTIL X>64 AND X<90
880      PRINTTAB(14,24);CHR$X'TAB(1,28)SP
C(18)'TAB(1,30)SPC(18):VDU7
890      IF A<5 AND X<>65 AND X<>74 AND X<
>80 AND X<>89 THEN COLOUR 11:PRINTTAB(2,
28)"A,J,P or Y first":SOUND1,-15,150,12:
UNTIL FALSE
900      X=X-64
910      IF PLACE(X)<>0 THEN COLOUR11:PRIN
T TAB(1,28);"OCCUPIED":VDU7
920      UNTIL PLACE(X)=0
930      PROCPLACE:PLACE(X)=A(A)
940      ENDPROC
950      :
960      DEFPROCMAN
970      I=I+1
980      COLOURA(A)
990      SOUND0,-10,4,1
1000     PRINTTAB(I,30);" ";MAN$
1010     IF I>15 THEN PROCCAUGHT
1020     ENDPROC
1030     :
1040     DEFPROCCAUGHT
1050     *FX9,10
1060     *FX10,10
1070     FOR I%=1TO25
1080         COLOURI%:PRINTTAB(17,30)MAN$
1090         SOUND1,-12,I%*10,1
1100         PRINTTAB(0,30)"HA!HA! I'M FREE!"
"
1110         NEXT:SC=A-1:D=1
1120     ENDPROC
1130     :
1140     DEFPROCPLACE

```

```

1150 PRINTTAB(0,30)SPC(19)
1160 COLOUR(A):COLOUR128
1170 PRINTTAB(H(X),V(X)-1)MAN$
1180 IF A>4 THEN PROCIMATE
1190 PROCWAIT(10)
1200 COLOUR134
1210 ENDPROC
1220 :
1230 DEFPROCIMATE
1240 H=H(X)+1:V=V(X):NL=0
1250 X1=64*H+10:Y1=1024-(32*V)
1260 P=POINT(X1,Y1):NL=0
1270 HH=H:VV=V
1280 FORI%=-3TO3STEP3
1290     FORJ%=-3TO3STEP6
1300         H=H+I%:V=V+J%
1310         X1=64*H+10:Y1=1024-(32*V)
1320         IF P=POINT(X1,Y1) THEN SC=A:D
=3 ELSE IF POINT(X1,Y1)<>0 AND POINT(X1,
Y1)<>6 THEN NL=1
1330         H=HH:V=VV:NEXT:NEXT
1340     IF D<>0 ENDPROC
1350     FOR I%=-3TO3STEP6
1360         H=H+I%
1370         X1=64*H+10:Y1=1024-(32*V)
1380         IF P=POINT(X1,Y1) THEN SC=A:D=3:
ENDPROC ELSE IF POINT(X1,Y1)<>0 AND POIN
T(X1,Y1)<>6 THEN NL=1
1390         H=HH:V=VV:NEXT
1400     IF D<>0 ENDPROC
1410     IF NL=0 THEN SC=A:D=4
1420 ENDPROC
1430 :
1440 DEFPROCFIN(D)
1450 IF D=5 THEN SOUND1,-15,150,10 ELS
E SOUND1,-15,25,10
1460 VDU28,0,31,19,22
1470 COLOUR129+D:CLS
1480 COLOUR7:PRINT"GAME OVER";
1490 IF D=5 PRINT"-Well done.""All
the aliens""are safe and sound!"ELSE
IF D=1 PRINT" - OOPS!""One of the al
iens""escaped and rescued"" the
rest!"
1500 IF D=3 PRINT"-You placed""aliens

```

```

    of the same""colour next to each""
other.They'll fight!"ELSE IF D=4 PRINT"
- Aliens""need mates to exist"" and
d sustain life"" Spoil-sport!!"
1510 IF SC>0 THEN PRINT TAB(0,8)"You c
aught ";SC;
1520 IF SC=1 THEN PRINTTAB(14,8)"alien
"; ELSE IF SC>1 PRINT TAB(14,8)"aliens";
1530 PROCWAIT(90):CLS
1540 *FX 15,0
1550 PRINT TAB(3,5)"ANOTHER GAME?"
1560 REPEAT:A$=GET$:UNTIL A$="Y" OR A$
="N"
1570 IF A$="N" THEN FIN%=TRUE
1580 ENDPROC
1590 :
1600 DEFPROCWAIT(S)
1610 W=TIME:REPEATUNTIL TIME-W>S*10
1620 ENDPROC

```

# OMNIOPOLY

By Richard Smith

Omniopoly bears more than a passing resemblance to a well-known board game... become a property millionaire! The computer plays the Bank and displays a very realistic board, while two human opponents compete against each other. The only change to the standard rules is the Auction Rule. In the boardgame a property is auctioned if not purchased by the first player to land on it, but this does not apply with the computer version. All other features of the boardgame are supported, and presentation of this excellent game rivals that of commercial versions. See you in Mayfair!

## Program Description

There are two parts to the Omniopoly program. The first sets up the user-defined graphics and gives instructions for playing the game, saving space in the main program. Program 2 should be saved with the name "OMN".

### Program 1

**70** Change mode.

**80-380** Set up the user-defined graphics.

**390-400** Print the title on the screen.

**410** Define the text window.

**420** Load and run program 2, "OMN".

**410-530** Print the instructions.

### Program 2

**80-90** Initialize variables.

**100-1030** Main body of program, which controls program flow and calls the following procedures.

**1040-1080** PROCGETPROP - gets details of the property on a square.

**1090-1190** various checking procedures.

**1200-1260** PROCINPUT TEXT - general purpose input routine.



**1230 PROCGETHOUSES.**

**1280-1360** Alternative routes if ESCAPE is pressed.

**1310-1400** PROCBOARD - draws the onmiopoly board, using the following two procedures.

**1410** PROCSBAR draws a bar.

**1420** PROCBOX - draws a box.

**1480-1520** PROCSMBOX - fills in a box.

**1530** PROCNAM - prints a specified message on the screen.

**1540-1600** PROCLBOX - prints a specified message on the screen.

**1640-1670** PROCHIGH - general procedure to highlight areas.

**1680-1700** PROCUNHIGH cancels highlight.

**1210** PROCGETDA - read in data.

**1120-1850** PROCTITLEDEED - gets all the required data for a title deed and prints onto the screen using PROCTDERAME

**1860** PROCGETLCOL - reads in required colour from data for property groups.

**1810-1910** PROCDERAME - draws title-deed on screen.

**1920-1960** Procedures to format text.

**1910-2100** Main routine to handle "COMMUNITY CHEST", or "CHANCE".

**2110-2150** Several short text formatting routines.

**2160-2310** PROCAT - special squares, GO, JAIL, and so on.

**2320-2310** Routine to check number of houses on a property. It also allows the sale or purchase of houses.

**2380** PROCSALE - calls other procedures.

**2390-2480** Shows details of square that player is on.

**2490** PROCPG - PASS GO-COLLECT £200.

**2500-2510** PROCMONEY prints player's cash holdings.

**2520-2540** PROCGET - checks whether Y or N key is pressed.

**2550-2590** PROCTYPE - formats the printing of Z\$.

**2600-2610** PROCWAIT - flush keyboard buffer and check for whether a key has been pressed.

**2620** PROCCLS - clears the middle of the board to screen colour.

**2630-2640** Print a warning message if applicable.

**2650-2710** PROCPROP checks the status of a property.

**2120** PROCASK

**2130-2140** PROCGETCOL checks property colour.

**2150-2180** PROCALL - defines positions for counter movement  
**2190-2800** PROCSTNRENT calculates the rent on stations.  
**2810-2820** PROCUTRENT calculates the rent to be charged on utilities.  
**2830-2850** PROCROLL - rolls die and checks for third double throw.  
**2860-3620** Data.

---

## Program 1

```
10 REM OMNIPOLY loader
20 REM Runs on any 32K BBC
30 REM
40 REM (c) Copyright Richard Smith
50 REM Aug 23 1983
60 REM
70 MODE7
80 VDU23,224,7,5,5,7,5,5,5,0
90 VDU23,225,7,5,5,6,5,5,7,0
100 VDU23,226,7,4,4,4,4,4,7,0
110 VDU23,227,6,5,5,5,5,5,6,0
120 VDU23,228,7,4,4,6,4,4,7,0
130 VDU23,229,7,4,4,6,4,4,4,0
140 VDU23,230,7,4,4,4,5,5,7,0
150 VDU23,231,5,5,5,7,5,5,5,0
160 VDU23,232,2,2,2,2,2,2,2,0
170 VDU23,233,1,1,1,1,5,5,7,0
180 VDU23,234,5,5,6,6,6,5,5,0
190 VDU23,235,4,4,4,4,4,4,7,0
200 VDU23,236,5,7,7,7,5,5,5,0
210 VDU23,237,7,5,5,5,5,5,5,0
220 VDU23,238,7,5,5,5,5,5,7,0
230 VDU23,239,7,5,5,7,4,4,4,0
240 VDU23,240,7,5,5,6,5,5,5,0
250 VDU23,241,7,4,4,7,1,1,7,0
260 VDU23,242,7,2,2,2,2,2,2,0
270 VDU23,243,5,5,5,5,5,5,7,0
280 VDU23,245,5,5,5,7,7,7,5,0
290 VDU23,246,5,5,5,2,5,5,5,0
300 VDU23,247,5,5,5,7,2,2,2,0
310 VDU23,248,7,1,1,7,4,4,7,0
320 VDU23,249,3,2,2,7,2,2,7,0
```

```

330 VDU23,250,7,1,1,7,1,1,7,0
340 VDU23,251,5,5,5,7,1,1,1,0
350 VDU23,252,7,4,4,7,5,5,7,0
360 VDU23,253,7,1,1,1,1,1,1,0
370 VDU23,254,7,5,5,7,5,5,7,0
380 VDU23,255,7,5,5,7,1,1,1,0
390 FORI%=0TO1:VDU129,157,131,141:PRIN
TSPC(12)"OMNIOPOLY":NEXT
400 PRINT'"          (c) COPYRIGHT RICHARD S
MITH 1983"
410 PROC_instructions
420 VDU28,10,17,29,15
430 CHAIN"OMN"
440 DEFPROC_instructions
450 PRINT'"  Omniopoly is a board game
for two      players which bears more th
an a passing resemblance to a well known
board game. The computer plays the bank
and displaysa very realistic board, whi
le the two   players compete ";
460 PRINT'"against each other. "
470 PRINT"The only change to the stand
ard rules    is the Auction Rule. In this
game if oneplayer does not buy a proper
ty they landon, the property is not auct
ioned.        The game is very user frien
dly so there is not much more to say ";
480 PRINT"at this stage  except to con
tinue after any new screen display just
press space."
481 PRINT'"Pressing ESCAPE will return
you to the""main menu."
490 COLOUR 12
500 PRINT 'SPC(8);CHR$(132);"PRESS SPA
CE TO CONTINUE";
510 REPEAT:G$=GET$:UNTIL G$=" "
520 CLS:FORI%=0TO1:VDU129,157,131,141:
PRINTSPC(12)"OMNIOPOLY":NEXT
530 ENDPROC

```

## Program 2

```

10 REM OMNIPOLY
20 REM Runs on any 32K BBC

```

```

30 REM
40 REM (c) Copyright Richard Smith
50 REM Aug 31 1983
60 REM
70 REMONERRORGOTO1360
80 DIMN%(8),A$(30),NAM$(9),LO%(40),N$(
(2),OPOS(2),NPOS(2),MO(2),IJ(2),GJ(2),PR
(4),PR2(3),PR$(3),P$(5),H(3),CH%(52,1),C
HP(1),JC(1):NJ=1:DO=0:FORI%=0TO4:P$(I%)=
STR$I%:NEXT:P$(5)="H":A$=STRING$(100," ")
):B$=A$:N$=A$:*FX4,1
90 A=RND(-TIME):FORI%=1TOTIME MOD10:A
=RND(1):NEXT:LE$="ABCDEFGHIJKLMNOPQRSTUVWXYZ
XYZ£346789":OK$=LE$+" Q1250"
100 FORK%=0TO1:FORI%=1TO26
110 J%=RND(26)*2-1:IFCH%(J%,K%)=0C
H%(J%,K%)=I%ELSEGOTO110
120 NEXT:CH%(RND(26)*2,K%)=1:NEXT
130 MODE5:VDU19,1,2;0;19,2,6;0;29,640;
512;23;8202;0;0;0;5:COLOUR129:COLOUR0:RE
STORE3270:FORI%=0TO9:READNAM$(I%):NEXT:P
ROCB0ARD:Y%=RND(2):A$=N$(Y%)+ " STARTS":Y
=320:PROCTEXT
140 ONERRORGOTO1280
150 PROCWAIT
160 X%=Y%:Z%=OPOS(X%):MO(X%)=INTMO(X%)
:PROCAT(Z%):PROC MONEY:PROCWARN:RESTORE36
10:FORI%=1TO8:READA$:MOVE-336,-32*(I%+1)
:A$=STR$I%+" "+A$:PROCPRINT(A%):NEXT:MOV
E-160,-320:PROCPRINT("ENTER CHOICE")
170 PROCWAIT:B=VALA$:IFB<1ORB>8THEN170
180 ONB GOTO190,440,450,480,510,580,77
0,850
190 IFSGNMO(X%)=-1THEN860
200 IFZ%=41THEN900ELSEIJ(X%)=0
210 D1=RND(6):D2=RND(6):IFD1<>D2 Y%=3-
X%:DO=0:GOTO230
220 Y%=X%:DO=DO+1
230 PROCROLL
240 PROCWAIT
250 NPOS(X%)=Z%:PROC NPOS
260 IFA<>18ANDA<>11PROC MONEY
270 IFA=17Z%=41:GOTO240ELSEIFA<>18ANDA
<>11PROC MONEY
280 IFA=10PROCCOMCHEST:GOTO880

```

```

290 IFA=12PROCCHANCE:GOTO880
300 ONERRORGOTO1280
310 IFA=9ORA=15ORA=16ORA=19THEN150
320 IFA=11MO(X%)=MO(X%)-200:PROC MONEY:
GOTO150
330 IFA=18MO(X%)=MO(X%)-100:PROC MONEY:
GOTO150
340 PROCGETDA(Z%):READD,A$,A
350 IFN%THEN380
360 MOVE-336,-96:PROCPRINT("DO YOU WAN
T TO BUY IT"):PROCGET:IFA$="N"THEN160
370 MO(X%)=MO(X%)-A:LO%(Z%)=LO%(Z%)OR1
6*X%:PROCTITLEDEED(Z%):GOTO150
380 IFLO%(Z%)AND16*X%ORLO%(Z%)AND64THE
N150
390 MOVE-64,-96:PROCPRINT("RENT")
400 IFD=13PROCUTRENT:GOTO430ELSEIFD=14
PROCSTNRENT:GOTO430
410 FORI%=0TOK%:READA:NEXT
420 IFK%=0PROCALL(D):IFB A=A*2
430 A$="£"+STR$A:Y=-160:PROCTEXT:MO(3-
X%)=MO(3-X%)+A:MO(X%)=MO(X%)-A:GOTO150
440 PROCHIGH:GOTO160
450 PROCASK:FORA%=1TO40:PROCGETDA(A%):
READA:IFW=0THEN460ELSEIFA<>W THEN470
460 IFLO%(A%)AND(X%*16)PROCPROP(A%,1
):IFA$="T"PROCTITLEDEED(A%):PROCWAIT
470 NEXT:GOTO160
480 PROCASK:FORI%=1TO40:PROCGETDA(I%):
READA:IFA>8ANDA<>13ANDA<>14THEN500
490 IFW=0ORW=A PROCPROP(I%,0)
500 NEXT:GOTO160
510 PROCCLS:PROCGETPROP:PROCCLS:PROCCH
ECK1:IFER THEN150
520 PROCHECK3:IF ER THEN 150
530 PROCHECK2:A$="WHAT IS THE AGREED/
SELLING PRICE":Y=224:PROCTEXT:PROCINPUTN
:LO%(A)=LO%(A)AND207OR16*(3-X%):MO(X%)=M
O(X%)+B:MO(3-X%)=MO(3-X%)-B
540 IF(LO%(A)AND64)=0THEN570
550 PROCCLS:X%=3-X%:A$=N$(X%)+"/DO YOU
WANT TO LIFT/THE MORTGAGE NOW":Y=320:PR
OCTEXT:PROCGET:IFA$="Y"PROCCLS:FL%=0:GOT
O840
560 PROCGETDA(A):READB,A$,B:A$="YOU ST

```

```

ILL HAVE TO PAY/INTEREST OF £"+STR$(B*.0
5):MO(X%)=MO(X%)-B*.05:Y=192:PROCTEXT:PR
OCWAIT
570 PROCTITLEDEED(A):PROCWAIT:GOTO160
580 FORI%=1TO3:H(I%)=0:NEXT:PROCCLS:A$
="DO YOU WANT TO SELL/BUILDINGS":Y=320:P
ROCTEXT:PROCGET:B$=A$
590 PROCCLS:PROCGETCOL(8):PROCCLS:PROC
ALL(A):IFB<>X%A$="YOU DO NOT HAVE THE/EN
TIRE GROUP/OR PART OF THE GROUP/IS MORTG
AGED":Y=320:PROCTEXT:GOTO150ELSEA$="EXIS
TING BUILDINGS/AND PROPERTY NAME":Y=320:
PROCTEXT
600 G%=0:PR(3)=-1:J%=0:FORI%=1TO40:PRO
CGETDA(I%):READW:IFW=A B=LO%(I%)AND7:J%=
J%+1:PR(J%)=B:PR2(J%)=I%:READA$,R%,R%,R%
,R%,R%,R%,R%:PROCSPACES(1):PROCNORTH:
PR$(J%)=N$:MOVEQ%,256-32*J%:PROCPRINT(P$
(PR(J%))+ " "+N$)
610 NEXT:ONERRORGOTO1350
620 IFB$="Y"THEN670ELSEA$="HOUSES COST
£"+STR$R%+" EACH":Y=96:PROCTEXT
630 A$="HOW MANY HOUSES DO/YOU WANT TO
BUILD ON":Y=32:PROCTEXT
640 FORI%=1TOJ%
650 PROCGETHOUSES:H(I%)=B:IFPR(I%)+B
>5MOVE-128,-192:PROCPRINT("TOO MANY"):PR
OCWAIT:GOTO650
660 GOTO710
670 R%=-R%/2:A$="HOUSES ARE WORTH £"
+STR$-R%:Y=96:PROCTEXT
680 A$="HOW MANY HOUSES DO/YOU WANT
REMOVED FROM":Y=32:PROCTEXT
690 FORI%=1TOJ%
700 PROCGETHOUSES:H(I%)=B:IFPR(I%)
-B<0MOVE-128,-192:PROCPRINT("TOO MANY"):
PROCWAIT:GOTO700
710 NEXT
720 L%=5:H%=0:FORI%=1TOJ%:A=PR(I%)+S
GNR%*H(I%):IFA<L%L%=A
730 IFA>H%H%=A
740 NEXT:IFH%-L%>1A$="BUILDINGS AR
E NOT/DISTRIBUTED EVENLY":Y=-192:PROCTEX
T:PROCWAIT:GCOL0,1:PROCHBAR(-320,-64,-25
6):GCOL0,0:ONSGNR%+2GOTO690,0,640

```

```

750      FORI%=1TOJ%:LO%(PR2(I%))=LO%(PR2
(I%))AND248OR(PR(I%)+SGNR%*H(I%)):MO(X%)
=MO(X%)-R%*H(I%):NEXT:ONERRORGOTO1280
760      GOTO160
770      FL%=1:PROCCLS:A$="DO YOU WANT TO
LIFT/A MORTGAGE":Y=320:PROCTEXT:PROCGET
:B$=A$
780      PROCGETPROP:PROCCLS:PROCHECK1:I
FER PROCWAIT:GOTO780
790      IFB$="Y"THEN830
800      PROCHECK2:IF ER PROCWAIT:GOTO78
0
810      PROCHECK3:IF ER PROCWAIT:GOTO78
0
820      LO%(A)=LO%(A)OR64:PROCGETDA(A):R
EADB,A$,B:MO(X%)=MO(X%)+B/2:PROCTITLEDEE
D(A):PROCWAIT:GOTO780
830      PROCHECK4:IF ER PROCWAIT:GOTO78
0
840      LO%(A)=LO%(A)AND191:PROCGETDA(A)
:READB,A$,B:MO(X%)=MO(X%)-.55*B:PROCTITL
EDEED(A):PROCWAIT:IFFL%THEN780ELSEGOTO16
0
850      PROCCLS:A$="ARE YOU SURE YOU/WAN
T TO END/ /YOU WILL LOSE/IF YOU DO":Y=32
0:PROCTEXT:PROCGET:IFA$="N"THEN160
860      PROCCLS:A$=N$(3-X%)+"/ /IS THE W
INNER":Y=320:PROCTEXT:PROCWAIT:MODE7:*FX
4,0
870      END
880      IFMV THEN240ELSEONERRORGOTO1280
890      PROCWAIT:PROCAT(Z%):PROC MONEY:GO
TO150
900      DO=0:IJ(X%)=IJ(X%)+1:GCOL0,1:PRO
CHBAR(352,-64,-356):GCOL0,0:N$="YOU ARE
IN JAIL":PROCCENTRE:MOVEX,-64:PROCPRINT(
N$):RESTORE3620:A=3:B=2:IFGJ(X%)A=5:B=3
910      FORI%=1TOA:READA$:MOVE-320,-32*(
I%+3):PROCPRINT(A$):NEXT:MOVE-160,-32*(I
%+4):PROCPRINT("ENTER CHOICE")
920      PROCWAIT:A=VAL(A$):IFA<1ORA>B TH
EN920
930      ONA GOTO940,960,1010
940      MO(X%)=MO(X%)-50
950      Y%=3-X%:IJ(X%)=0:OPOS(X%)=11:Z%=

```

```

11:GOTO210
  960      D1=RND(6):D2=RND(6):PROCROLL:IFD
1=D2 Y%=X%:DO=1:GOTO990
  970      Y%=3-X%:IFIJ(X%)=3A$="YOU MUST P
AY £50":Y=192:PROCTEXT:MO(X%)=MO(X%)-50:
GOTO1000
  980      GOTO150
  990      Y%=X%
 1000      OPOS(X%)=11:Z%=11+D1+D2:IJ(X%)=0
:GOTO240
 1010      GJ(X%)=GJ(X%)-1:IFJC(0)=X%J%=0EL
SEJ%=1:JC(J%)=0:I%=CHP(J%):IFI%=0I%=52
 1020      CH%(I%,J%)=1
 1030      GOTO950
 1040      DEFPROCGETPROP:PROCGETCOL(10):PR
OCCLS:W=A:IFA=9W=14ELSEIFA=10W=13
 1050      J%=0:FORI%=1TO40:PROCGETDA(I%):R
EADA:IFW=A READ A$:PROCSPPACES(1):PROCNOR
TH:J%=J%+1:N$=STR$J%+" "+N$:MOVE-320,352
-J%*32:PROCPRINT(N%):PR(J%)=I%
 1060      NEXT:MOVE-320,288-32*J%:PROCPR
INT("ENTER CHOICE")
 1070      PROCWAIT:A=VALA$:IFA<1ORA>J%THEN
1070
 1080      A=PR(A):ENDPROC
 1090      DEFPROCCHCK1:ER=0:IF(LO%(A)AND4
8)/16<>X%A$="YOU DO NOT OWN THIS/PROPERTY
Y":Y=320:PROCTEXT:ER=1
 1100      ENDPROC
 1110      DEFPROCCHCK2:ER=0:IFLO%(A)AND64
A$="THIS PROPERTY IS/MORTGAGED":Y=320:PR
OCTEXT:ER=1
 1120      ENDPROC
 1130      DEFPROCCHCK3:ER=0:IFLO%(A)AND7A
$="THERE ARE BUILDINGS/ON THIS SITE":Y=3
20:PROCTEXT:ER=1
 1140      ENDPROC
 1150      DEFPROCCHCK4:ER=0:IF(LO%(A)AND6
4)=0A$="THIS PROPERTY IS/NOT MORTGAGED":
Y=320:PROCTEXT:ER=1
 1160      ENDPROC
 1170      DEFPROCINPUTN:X=-96:Y=128:PROCTY
PE("£"):X=-64
 1180      MOVEX,Y:PROCINPUT:B=VALB$:IFB$<>
STR$B OR B>9999 THENGCOL0,1:MOVEX,Y:PROC

```



```

TYPE(B$):GCOL0,0:GOTO1180
  1190      ENDPROC
  1200      DEFPROCINPUTTEXT:MOVE-160,224:PR
OCINPUT:ENDPROC
  1210      DEFPROCINPUT:B$=" "
  1220      PROCWAIT:IFASCA$<>13THEN1240
  1230      IFB$=" "THEN1220ELSEENDPROC
  1240      IFASCA$=127ANDLENB$>0GCOL0,1:PLO
T0,-32,0:PROCPRINT(RIGHT$(B$,1)):B$=LEFT
$(B$,LENB$-1):GCOL0,0:PLOT0,-32,0:GOTO12
20
  1250      IFINSTR(OK$,A$)=0ORLENB$=10THEN1
220
  1260      B$=B$+A$:PROCPRINT(A$):GOTO1220
  1270      DEFPROCGETHOUSES:GCOL0,1:PROCHBA
R(320,-64,-256):GCOL0,0:N$=PR$(I%):PROCC
ENTRE:MOVEX,-64:PROCPRINT(N$):X=-32:Y=-1
28:GOTO1180
  1280      IFERR=17THEN160
  1290      MODE6:REPORT:PRINT" in line ";ER
L:*FX 4,0
  1300      END
  1310      IFERR<>17THEN1290
  1320      PROCUNHIGH:ONERRORGOTO1280
  1330      GOTO160
  1340      IFERR=17THEN250ELSEGOTO1290
  1350      IFERR=17THEN720ELSEGOTO1290
  1360      IFERR=17RUN ELSEGOTO1290
  1370      DEFPROCBOARD:GCOL0,1:PROCSBAR(51
2,512):GCOL0,2:PROCSBAR(480,480):GCOL0,1
:PROCSBAR(360,360):GCOL0,0:PROCBOX(480,4
80):PROCBOX(360,360)
  1380      FORY=-360TO360STEP80:MOVE-480,Y:
DRAW-360,Y:MOVEY,480:DRAWY,360:MOVE480,Y
:DRAW360,Y:MOVEY,-480:DRAWY,-360:NEXT
  1390      FORI%=1TO40:PROCGETDA(I%):READA:
PROCLOC(I%):IFA<9PROCSMBOX(I%)ELSEPROCNA
M(I%)
  1400      NEXT:GA=0:FORX%=1TO2:MO(X%)=15
00:PROCCLS:A$="WHAT IS THE NAME OF/PLAYE
R "+STR$X%:Y=320:PROCTEXT:PROCINPUTTEXT:
N$(X%)=B$:OPOS(X%)=40:Z%=1:PROCNPOS:NEXT
:GA=1:ENDPROC
  1410      DEFPROCSCBAR(X,Y)MOVEX,Y:MOVEX,-Y
:PLOT85,-X,-Y:MOVE-X,Y:PLOT85,X,Y:ENDPRO

```

```

C
1420      DEFPROCBOX(X,Y)MOVEX,Y:DRAWX,-Y:
DRAW-X,-Y:DRAW-X,Y:DRAWX,Y:ENDPROC
1430      DEFPROCLOC(I%)S=(I%-1)MOD10*80-3
60:ON(I%-1)DIV10+1GOTO1440,1450,1460,147
0
1440      X=-S:Y=-360:ENDPROC
1450      X=-360:Y=S:ENDPROC
1460      X=S:Y=360:ENDPROC
1470      X=360:Y=-S:ENDPROC
1480      DEFPROCSTBOX(I%)ON(I%-1)DIV10+1G
OTO1490,1500,1510,1520
1490      MOVEX+8,Y-4:MOVEX+72,Y-4:PLOT85,
X+72,Y-40:MOVEX+8,Y-40:PLOT85,X+8,Y-4:EN
DPROC
1500      MOVEX-8,Y-4:MOVEX-8,Y-76:PLOT85,
X-40,Y-76:MOVEX-40,Y-4:PLOT85,X+8,Y-4:EN
DPROC
1510      MOVEX-8,Y+4:MOVEX-72,Y+4:PLOT85,
X-72,Y+40:MOVEX-8,Y+40:PLOT85,X-8,Y+4:EN
DPROC
1520      MOVEX+8,Y+4:MOVEX+8,Y+76:PLOT85,
X+40,Y+76:MOVEX+40,Y+4:PLOT85,X+8,Y+4:EN
DPROC
1530      DEFPROCNAM(I%)B$=NAM$(A-9):PROCL
BOX(0):ENDPROC
1540      DEFPROCBOX(M%):IFI%=11THEN1590E
LSEIFI%=31THEN1600ELSEON(I%-1)DIV10+1GOT
O1550,1560,1570,1580
1550      MOVEX+8,Y-16-M%*32:PROCPRINT(B$)
:ENDPROC
1560      MOVEX-48-32*M%,Y-12:PROCPRINT(LE
FT$(B$,1)):MOVEX-48-32*M%,Y-44:PROCPRINT
(RIGHT$(B$,1)):ENDPROC
1570      MOVEX-72,Y+40+32*M%:PROCPRINT(B$)
:ENDPROC
1580      MOVEX+16+32*M%,Y+68:PROCPRINT(LE
FT$(B$,1)):MOVEX+16+32*M%,Y+36:PROCPRINT
(RIGHT$(B$,1)):ENDPROC
1590      MOVEX-72,Y-16-32*M%:PROCPRINT(B$)
:ENDPROC
1600      MOVEX+12,Y+40+32*M%:PROCPRINT(B$)
:ENDPROC
1610      DEFPROCGETCOL(N%)PROCCLS:RESTORE
3280:FORI%=0TON%-1:READC$:MOVE-320,320-I

```

```

%*32:PROCPRINT(STR$I%+" "+C$):NEXT:MOVE-
320,256-I%*32:PROCPRINT("ENTER CHOICE")
1620    PROCWAIT:A=VALA$:IFA=0ANDAS<>"0"
THENA=-1
1630    IFA<0ORA>N%-1THEN1620ELSEA=A+1:P
ROCGETLCL:ENDPROC
1640    DEFPROCHIGH:PROCGETCOL(8):IFC=0O
RC=6C=C+8
1650    ONERRORGOTO1310
1660    VDU19,3,C;0::GCOL0,3:FORI%=1TO40
:PROCGETDA(I%):READB:IFB=A PROCLOC(I%):P
ROCSMBOX(I%)
1670    NEXT:PROCWAIT:PROCUNHIGH:ENDPR
OC
1680    DEFPROCUNHIGH:GCOL0,0:FORI%=1TO4
0:PROCGETDA(I%):READB:IFA=B PROCLOC(I%):
PROCSMBOX(I%)
1690    NEXT:ONERRORGOTO1280
1700    ENDPROC
1710    DEFPROCGETDA(I%)RESTORE(2850+10*
I%):ENDPROC
1720    DEFPROCTITLEDEED(I%)PROCGETDA(I%
):READA:IFA>8ANDAS<>13ANDAS<>14THEN1850
1730    IFLO%(I%)AND64THEN1830
1740    PROCTDFRAME:IFA=13ORA=14THENQ%=Q
%-64:GOTO1800
1750    MOVE-160,272:PROCPRINT("TITLE DE
ED"):PROCSACES(1):PROCNORTH
1760    PROCCENTRE:MOVEX,208:PROCPRINT(N
$):GCOL0,0:FORI%=0TO7:READN%(I%):NEXT:N%
(8)=N%(0)/2:RESTORE3300
1770    X=-288:Y=112:PROCTYPE("RENT"):FO
RI%=1TO8:Y=Y-32:IFI%>6Y=Y-32
1780    READAS$:PROCTYPE(A$):MOVEQ%,Y:P
ROCPRINT("£"+STR$N%(I%)):IFI%=7Y=Y-32:PR
OCTYPE("COST HOTEL £"+STR$N%(7)):Y=Y-32
:PROCTYPE("PLUS 4 HOUSES")
1790    NEXT:GOTO1850
1800    PROCSACES(1):PROCCENTRE:MOVEX,2
40+24*(A-13):PROCPRINT(N$):IFA=14MOVE-11
2,216:PROCPRINT("STATION"):RESTORE3310EL
SERESTORE3320
1810    X=-228:Y=96:FORI%=1TO12:READAS$:I
FAS=" "AS=" "
1820    PROCTYPE(A$):Y=Y-32:NEXT:GOTO1

```

850

```
1830      C=A:PROCCLS:GCOL0,3:READA$,A:PRO
CSPACES(1):VDU19,3,1;0;:PROCSBAR(320,320
):GCOL0,0:PROCBOX(320,320):FORY=-1TO1STE
P2:MOVE-288,288*Y:DRAW288,288*Y:DRAW288,
96*Y:DRAW-288,96*Y:DRAW-288,228*Y:NEXT:M
OVE-144,208:IFC=14N$=N$+" STN"
```

```
1840      PROCPRINT("MORTGAGED"):PROCNORTH
:A$=N$+" / /"+"MORTGAGED FOR £"+STR$(A/2)
:Y=48:PROCTEXT:Y=-132:X=-256:RESTORE3330
:FORI%=1TO4:READN$:PROCPR:NEXT
```

```
1850      ENDPROC
```

```
1860      DEFPROCGETLCOL:LOCALI%:RESTORE32
90:FORI%=1TOA:READC:NEXT:ENDPROC
```

```
1870      DEFPROCTDFRAME:PROCCLS:IFA<9PROC
GETLCOL ELSEC=7
```

```
1880      VDU19,3,C;0;:GCOL0,2-(A>12):PROC
SBAR(320,320):GCOL0,0:PROCBOX(320,320):I
FA<13ANDA<>2GCOL0,3:PROCHBAR(304,304,144
)
```

```
1890      GCOL0,0:PROCHBOX(304,304,144)
```

```
1900      PROCGETDA(I%):READA,A$:COLOUR131
:IFA=1ORA=8COLOUR2:GCOL0,2ELSECOLOUR0:GC
OL0,0
```

```
1910      ENDPROC
```

```
1920      DEFPROCSPACES(I%)B=INSTR(A$,"/",
I%):IFB=0N$=A$ELSEN$=LEFT$(A$,B-1)+" "+M
ID$(A$,B+1)
```

```
1930      A$=N$:ENDPROC
```

```
1940      DEFPROCCENTRE:X=-288+32*((18-LEN
N$)/2):ENDPROC
```

```
1950      DEFPROCNORTH:IFN$="NORTHUMBER LA
ND AVENUE"N$="NORTHUMBERLAND AVENUE"
```

```
1960      ENDPROC
```

```
1970      DEFPROCCHANCE:N$="CHANCE":CH=1:G
OTO1990
```

```
1980      DEFPROCCOMCHEST:N$="COMMUNITY CH
EST":CH=0
```

```
1990      MV=0:PROCWAIT:PROCCLS:VDU19,3,5,
5-2*CH;0;:GCOL0,3:PROCSBAR(320,192):GCOL
0,0:PROCBOX(320,192):Y=128:A$=N$:PROCTEX
T:Y=32
```

```
2000      IFCH%(CHP(CH)+2,CH)A=27:CH%(CHP(
CH)+2,CH)=0:JC(CH)=X%ELSEA=CH%(CHP(CH)+1
,CH):CHP(CH)=CHP(CH)+2:IFCHP(CH)=52CHP(C
```

```

H)=0
2010      RESTORE (3330+10*A):READA$,A1%,A2
%
2020      PROCTEXT:IFA1%MO(X%)=MO(X%)+A1%:
ENDPROC
2030      IFA2%=0THEN2050ELSEMV=1:IFA2%=-3
Z%=OPOS(X%)+3:ENDPROC
2040      Z%=A2%:ENDPROC
2050      IFA=27THEN2090
2060      B%=0:FORI%=1TO40:IF(LO%(I%)AND48
)/16<>X%THEN2080
2070      J%=LO%(I%)AND7:IFJ%<5B%=B%+J%*
40ELSEB%=B%+115
2080      NEXT:A$="TOTAL COST £"+STR$B%:
Y=-224:PROCTEXT:MO(X%)=MO(X%)-B%:ENDPROC
2090      GJ(X%)=GJ(X%)+1:IFJC(CH)=X%
2100      ENDPROC
2110      DEFPROCPR:PROCCENTRE:MOVEX,Y:Y=Y
-32:PROCPRINT(N$):ENDPROC
2120      DEFPROCHBAR(X,Y,Y1)MOVEX,Y:MOVEX
,Y1:PLOT85,-X,Y1:MOVE-X,Y:PLOT85,X,Y:END
PROC
2130      DEFPROCHBOX(X,Y,Y1)MOVEX,Y:DRAWX
,Y1:DRAW-X,Y1:DRAW-X,Y:DRAWX,Y:ENDPROC
2140      DEFPROCTEXT:I%=0
2150      PROCSPACES(I%+1):IFB=0N$=MID$(A$
,I%+1):PROCPR:ENDPROC ELSEN$=MID$(A$,I%+
1,B-I%-1):I%=B:PROCPR:GOTO2150
2160      DEFPROCAT(L%)PROCCLS:K%=0:GCOL0,
2:PROCHBAR(320,320,64):GCOL0,0:PROCHBOX(
320,320,64):Y=256:PROCGETDA(L%):READA:IF
A<9ORA=13ORA=14THEN2270
2170      ONA-8GOTO2180,2190,2200,2210,0,0
,2220,2230,2240,2250,2260
2180      A$="GO":GOTO2300
2190      A$="COMMUNITY/CHEST":GOTO2300
2200      A$="INCOME/TAX/PAY £200":GOTO230
0
2210      A$="CHANCE":GOTO2300
2220      A$="JAIL/JUST/VISITING":GOTO2300
2230      A$="FREE/PARKING":GOTO2300
2240      A$="GO TO/JAIL":GOTO2300
2250      A$="SUPER/TAX/PAY £100":GOTO2300
2260      A$="IN/JAIL":GOTO2300
2270      IFA<9PROCGETLCL:VDU19,3,C;0;:GC

```

```

OL0,3:PROCHBAR(312,316,224):GCOL0,0:MOVE
-320,224:DRAW320,224:Y=192:PROCHOUSES
2280    PROCGETDA(L%):READJ%,A$,J%:IFA=1
4A$=A$+" /STATION"
2290    PROCTEXT:A$="£"+STR$J%
2300    PROCTEXT:IFA<9ORA=13ORA=14N%=(LO
%(L%)AND63)/16:PROCSALE ELSEN%=4
2310    ENDPROC
2320    DEFPROCHOUSES:K%=LO%(L%)AND7:IFK
%=0ENDPROC ELSEIFK%=5A$="HOTEL"ELSEA$=ST
R$K%+" HOUSE"
2330    IFK%<5ANDK%>1A$=A$+"S"
2340    IFA=1ORA=8GCOL0,2ELSEGCOL0,0
2350    PROCSPACES(1):PROCENTRE:MOVEX,2
88:PROCPRINT(N$):GCOL0,0:ENDPROC
2360    DEFPROCSALE2:IFN%=0N$="FOR SALE"
ELSEN$="OWNED BY "+N$(N%)
2370    ENDPROC
2380    DEFPROCSALE:PROCSALE2:PROCENTRE
:MOVEX,52:PROCPRINT(N$):ENDPROC
2390    DEFPROCNPOS:PROCCLS:B$=LEFT$(N$(
X%),2):N%=OPOS(X%):G%=Z%:IFG%=41G%=11:NJ
=0:DO=0:IFD1=D2 Y%=3-X%
2400    IFZ%=41ANDN%<32AND N%>11 ST=-1EL
SEST=1
2410    IFN%=G%THEN2460
2420    REPEAT:PG=0:K%=N%:N%=N%+ST:IFN%=
41N%=1:PG=1
2430    GCOL0,2:I%=K%:PROCLOC(I%):PROC
LBOX(X%):GCOL0,0:I%=N%:SOUND1,-7,50,1:PR
OCLOC(I%):PROCLBOX(X%):IFPG ANDGA ANDNJ
PROCPG
2440    *FX15,1
2450    A=INKEY25:UNTILN%=G%
2460    OPOS(X%)=Z%
2470    IFGA PROCAT(Z%)
2480    NJ=1:ENDPROC
2490    DEFPROCPCG:VDU19,3,7;0;:GCOL0,3:P
ROCHBAR(320,320,192):GCOL0,0:PROCHBOX(32
0,320,192):A$="COLLECT £200 SALARY/AS YO
U PASS GO":Y=288:PROCTEXT:MO(X%)=MO(X%)+
200:PROCWAIT:ENDPROC
2500    DEFPROCONEY:N$=N$(X%):PROCENR
E:Y=0:PROCTYPE(N$):N$="£"+STR$(ABS(MO(X%
))) :IFSGNMO(X%)=-1N$="MINUS "+N$

```

```

2510      N$=N$+"      "+STR$GJ(X%):PROCCENTRE
:Y=-32:PROCTYPE(N$):ENDPROC
2520      DEFPROCGET
2530      PROCWAIT:IFA$<>"Y"ANDA$<>"N"THEN
2530
2540      ENDPROC
2550      DEFPROCTYPE(Z$)MOVEX,Y:GOTO2570
2560      DEFPROCPRINT(Z$)
2570      FORE%=1TOLENZ$:PLOT0,-32,0:F%=ASC
(MID$(Z$,E%,1)):IFF%=49F%=73ELSEIFF%=50
F%=90ELSEIFF%=53F%=83ELSEIFF%=48F%=79ELS
EIFF%=81F%=86:VDU23,244,7,5,5,5,7,2,3,0E
LSEIFF%=86VDU23,244,5,5,5,5,5,2,2,0
2580      IFF%=32PRINT"      ";ELSEPRINTCHR$(
223+INSTR(LE$,CHR$(F%)));
2590      NEXT:ENDPROC
2600      DEFPROCWAIT:*FX15,1
2610      SOUND1,-7,125,1:A$=GET$:ENDPROC
2620      DEFPROCCLS:GCOL0,1:PROCSBAR(352,
356):GCOL0,0:ENDPROC
2630      DEFPROCWARN:IFSGN(MO(X%))=-1A$="
YOU ARE IN DEBT/ /YOU MUST PAY THIS/BEFO
RE ROLLING DICE":Y=-96:PROCTEXT:PROCWAIT
:GCOL0,1:PROCHBAR(320,-96,-224):GCOL0,0
2640      ENDPROC
2650      DEFPROCPROP(I%,J%)PROCCLS:READA$
:IFA=14THENA$=A$+"/STATION"
2660      K%=LO%(I%)AND7:IFA>8THEN2690
2670      IFK%=0A$=A$+"/ /NO BUILDINGS"ELS
EIFK%=5A$=A$+"/ /HOTEL"ELSEA$=A$+"/ /"+S
TR$K%+" HOUSE"
2680      IFK%<5ANDK%>1A$=A$+"S"
2690      IFLO%(I%)AND64A$=A$+"/ /MORTGAGE
D"
2700      IFJ%A$=A$+"/ /PRESS T FOR/TITLE
DEED/ELSE ANY OTHER KEY"ELSEN%=(LO%(I%)A
ND48)/16:PROCSALE2:A$=A$+"/ /"+N$
2710      Y=320:PROCTEXT:PROCWAIT:ENDPROC
2720      DEFPROCASK:PROCCLS:A$="DO YOU WA
NT JUST/ONE GROUP LISTED":Y=320:PROCTEXT
:PROCGET:IFA$="N"W=0:ENDPROC
2730      PROCGETCOL(10):W=A:IFW=9W=14ELSE
IFW=10W=13
2740      ENDPROC
2750      DEFPROCALL(J%)B=-1:FORI%=1TO40:P

```

```

ROCGETDA(I%):READC:IFC<>J%THEN2780ELSEB%
=(LO%(I%)AND112)/16
  2760      IFB=-1B=B%ELSEIFB<>B%B=0
  2770      IFB<>B%B=0
  2780      NEXT:ENDPROC
  2790      DEFPROCSTNRENT:J%=-1:FORI%=6TO36
STEP10:IF(LO%(I%)AND112)/16=3-X%J%=J%+1
  2800      NEXT:A=2^J%*25:ENDPROC
  2810      DEFPROCUTRENT:J%=-1:FORI%=13TO29
STEP16:IF(LO%(I%)AND112)/16=3-X%J%=J%+1
  2820      NEXT:A=(4+6*J%)*(D1+D2):ENDPRO
C
  2830      DEFPROCROLL:PROCCLS:A$="YOU HAVE
ROLLED/ /"+STR$D1+" AND "+STR$D2:IFDO=3
A$=A$+"/ /YOU HAVE THROWN/THREE DOUBLES/
/GO TO JAIL":Z%=41:NJ=0:Y%=3-X%:DO=0ELS
EIFIJ(X%)=0Z%=Z%+D1+D2:IFZ%>40Z%=Z%-40
  2840      Y=320:PROCTEXT:ONERRORGOTO1340
  2850      ENDPROC
  2860      DATA9
  2870      DATA1,OLD KENT/ROAD,60,2,10,30,9
0,160,250,50
  2880      DATA10
  2890      DATA1,WHITECHAPEL/ROAD,60,4,20,6
0,180,320,450,50
  2900      DATA11
  2910      DATA14,KINGS CROSS,200
  2920      DATA2,ANGEL/ISLINGTON,100,6,30,9
0,270,400,550,50
  2930      DATA12
  2940      DATA2,EUSTON/ROAD,100,6,30,90,27
0,400,550,50
  2950      DATA2,PENTONVILLE/ROAD,120,8,40,
100,300,450,600,50
  2960      DATA15
  2970      DATA3,PALL MALL,140,10,50,150,45
0,625,750,100
  2980      DATA13,ELECTRIC/COMPANY,150
  2990      DATA3,WHITEHALL,140,10,50,150,45
0,625,750,100
  3000      DATA3,NORTHUMBER/LAND AVENUE,160
,12,60,180,500,700,900,100
  3010      DATA14,MARYLEBONE,200
  3020      DATA4,BOW STREET,180,14,70,200,5
50,750,950,100

```



3030 DATA10  
 3040 DATA4,MARLBOROUGH/STREET,180,14,  
 70,200,550,750,950,100  
 3050 DATA4,VINE STREET,200,16,80,220,  
 600,800,1000,100  
 3060 DATA16  
 3070 DATA5,STRAND,220,18,90,250,700,8  
 75,1050,150  
 3080 DATA12  
 3090 DATA5,FLEET STREET,220,18,90,250  
 ,700,875,1050,150  
 3100 DATA5,TRAFALGAR/SQUARE,240,20,10  
 0,300,750,925,1100,150  
 3110 DATA14,FENCHURCH ST,200  
 3120 DATA6,LEICESTER/SQUARE,260,22,11  
 0,330,800,975,1150,150  
 3130 DATA6,COVENTRY/STREET,260,22,110  
 ,330,800,975,1150,150  
 3140 DATA13,WATER/WORKS,150  
 3150 DATA6,PICCADILLY,280,22,120,360,85  
 0,1025,1200,150  
 3160 DATA17  
 3170 DATA7,REGENT/STREET,300,26,130,390  
 ,900,1100,1275,200  
 3180 DATA7,OXFORD/STREET,300,26,130,390  
 ,900,1100,1275,200  
 3190 DATA10  
 3200 DATA7,BOND STREET,320,28,150,450,1  
 000,1200,1400,200  
 3210 DATA14,LIVERPOOL ST,200  
 3220 DATA12  
 3230 DATA8,PARK LANE,350,35,175,500,110  
 0,1300,1500,200  
 3240 DATA18  
 3250 DATA8,MAYFAIR,400,50,200,600,1400,  
 1700,2000,200  
 3260 DATA19  
 3270 DATAGO,CC,IT,CH,UT,SN,JA,FP,GJ,ST  
 3280 DATABLACK,CYAN,MAGENTA,WHITE,RED,Y  
 ELLow,GREEN,BLUE,STATIONS,UTILITIES  
 3290 DATA0,6,5,7,1,3,2,4,0,0  
 3300 DATASITE ONLY,1 HOUSE,2 HOUSES,3 H  
 OUSES,4 HOUSES,HOTEL,COST HOUSE,MORTGAGE  
 3310 DATA" BRITISH RAILWAYS",,RENT  
 £25,,IF 2 STATIONS £50,,IF 3 STATI

ONS £100,, IF 4 STATIONS £200,,,MORTGAGE  
 £100  
 3320 DATAIF ONE UTILITY,IS OWNED RENT I  
 S 4,TIMES AMOUNT SHOWN,ON DICE,, IF BOTH  
 UTILITIES,ARE OWNED RENT IS,10 TIMES AMO  
 UNT,SHOWN ON DICE,,,MORTGAGE £75  
 3330 DATACARD MUST BE,TURNED THIS SIDE,  
 UP IF PROPERTY,IS MORTGAGED  
 3340 DATASPEEDING FINE £15,-15,0  
 3350 DATADOCTORS FEE/PAY £50,-50,0  
 3360 DATAPAY HOSPITAL £100,-100,0  
 3370 DATAYOU ARE ASSESSED/FOR STREET RE  
 PAIRS/£40 PER HOUSE/£115 PER HOTEL,0,0  
 3380 DATABANK PAYS YOU/DIVIDEND OF £50,  
 50,0  
 3390 DATAFROM SALE OF STOCK/YOU GET £50  
 ,50,0  
 3400 DATAYOU INHERIT £100,100,0  
 3410 DATAINCOME TAX REFUND/COLLECT £20,  
 20,0  
 3420 DATAANNUITY MATURES/COLLECT £100,1  
 00,0  
 3430 DATAYOUR BUILDING LOAN/MATURES/REC  
 EIVE £150,150,0  
 3440 DATAADVANCE TO/TRAFALGAR SQUARE/IF  
 YOU PASS GO/COLLECT £200,0,25  
 3450 DATAADVANCE TO/PALL MALL/IF YOU PA  
 SS GO/COLLECT £200,0,12  
 3460 DATAADVANCE TO MAYFAIR,0,40  
 3470 DATAADVANCE 3 SPACES,0,-3  
 3480 DATAPAY INTEREST ON/A LOAN/£50,-50  
 ,0  
 3490 DATATAKE A TRIP TO/MARYLEBONE STAT  
 ION/AND IF YOU PASS GO/COLLECT £200,0,16  
 3500 DATAADVANCE TO GO,0,1  
 3510 DATADRUNK IN CHARGE/FINE £20,-20,0  
 3520 DATAPAY YOUR INSURANCE/PREMIUM £50  
 ,-50,0  
 3530 DATAPAY SCHOOL FEES/OF £150,-150,0  
 3540 DATAYOUR BUILDING LOAN/MATURES/REC  
 EIVE £150,150,0  
 3550 DATAYOU HAVE WON A/CROSSWORD/COMPE  
 TITION/COLLECT £100,100,0  
 3560 DATABANK ERROR IN YOUR/FAVOUR/COLL  
 ECT £200,200,0

3570 DATA YOU HAVE WON/SECOND PRIZE IN A  
/BEAUTY CONTEST/COLLECT £10,10,0

3580 DATA RECEIVE INTEREST/ON 7PC PREFER  
ENCE/SHARES/£25,25,0

3590 DATA GO TO JAIL/MOVE DIRECTLY TO/JA  
IL/DO NOT PASS GO/DO NOT COLLECT £200,0,  
41

3600 DATA GET OUT OF JAIL/FREE/ /THIS CA  
RD MAY BE/KEPT UNTIL NEEDED,0,0

3610 DATA ROLL DICE,SHOW COLOUR GROUP,LI  
ST OWNED PROPERTY,LIST ALL PROPERTY,SELL  
PROPERTY,BUY OR SELL HOUSES,MORTGAGE PR  
OPERTY,END GAME

3620 DATA 1 PAY £50,2 ROLL DICE AND HOPE  
, " FOR DOUBLES",3 USE A GET OUT OF," J  
AIL FREE CARD"

# MAP QUIZ

By Alan Wagstaff

Map Quiz is a simple geography quiz in which you are asked 15 questions about the location of towns in England, Scotland and Wales. A dot appears on the map and you have to guess which town it is by choosing from the five names that are provided. All you have to do is press the number of the answer that you think is correct, then sit back!

## Program Description

- 10** Set up the arrays used in the game.
- 80** The start of the main loop.
- 30** Change mode and turn off the cursor.
- 40** Call the start procedure.
- 80-80** The game loop.
- 90** Print the final score.
- 100** Return to the main loop.
- 120-290** PROCstart.
- 130** Change colour to green.
- 140** Draw the map.
- 150-190** Print the instructions in a text window.
- 200-210** Wait for a key to be pressed.
- 220-260** Read in the data for the towns and their positions.
- 270** Define the character for the towns.
- 280** Set the score to zero and print it.
- 300-520** PROCquestion.
- 320** Find a town that has not already been used.
- 330** Print the town at the correct place on the map.
- 340** Set all the positions in the ANS% array to zero.
- 350** Pick a number between 1 and 5 for the correct answer.
- 360-390** Pick four other towns at random, ensuring that the same one is not used twice.
- 400** Define the window for the questions.
- 410-440** Print the five possible answers.

**450-410** Wait for a number between 1 and 5 to be pressed.  
**490** Print whether or not the answer was correct.  
**510** Remove the town from the map.  
**540-610** PROCscore - print the player's score.  
**630-650** PROCdelay - delay for a given time.  
**610-730** PROCfmish - print the final score.  
**750-820** PROCmap - draw a map of a given size at a given place on the screen.  
**840-1060** Data for the X coordinates of the map.  
**1080-1290** Data for the Y coordinates of the map.  
**1310-1420** Data for the towns.

---

```

10  DIM X%(260),Y%(260),T$(47),TX%(47
),TY%(47),ASKED%(47),ANS%(5)
20  REPEAT
30      MODE 1:VDU 23;10,32;0;0;0
40      PROCstart
50      FOR Q%=1 TO 15
60          PROCquestion
70          PROCscore
80      NEXT
90      PROCfinish
100     UNTIL FALSE
110     REM *****
120     DEF PROCstart
130     VDU 19,1,2;0;
140     GCOL 0,1:PROCmap(400,4,3.7)
150     COLOUR 2:PRINTTAB(7,0)"MAP"TAB(26
,0)"QUIZ"
160     VDU 28,0,24,12,2
170     PRINT"In this quiz you will be a
sked 15      questions      about towns      in
England,      Scotland and Wales."
180     PRINT'"For each you will be shown
a town and      given a list of 5 possiblea
nswers. All you have to do is pick th
e right one"
190     PRINT"Press any key to start"
200     *FX 15,1
210     G%=GET

```

```

220 RESTORE 1300
230 FOR L%=1 TO 47
240 READ T$(L%),TX%(L%),TY%(L%)
250 ASKED%(L%)=0
260 NEXT
270 VDU 23,224,112,248,248,112,0,0,0,
0
280 SC%=0:Q%=0:PROCscore
290 ENDPROC
300 REM *****
310 DEF PROCquestion
320 REPEAT:N%=RND(47):UNTIL ASKED%(N%)
)=0
330 VDU 5:GCOL 3,3:MOVE TX%(N%),TY%(N
%):VDU 224,4
340 FOR L%=1 TO 5:ANS%(L%)=0:NEXT
350 C%=RND(5):ANS%(C%)=N%:ASKED%(N%)=
1
360 FOR L%=1 TO 5
370 REPEAT:O%=RND(47):UNTIL ANS%(1)
<>O% AND ANS%(2)<>O% AND ANS%(3)<>O% AND
ANS%(4)<>O% AND ANS%(5)<>O%
380 IF L%<>C% ANS%(L%)=O%
390 NEXT
400 VDU 28,23,11,39,2,12
410 PRINT"Which is correct?"
420 FOR L%=1 TO 5
430 PRINTTAB(0,L%+1);L%;") ";T$(ANS
%(L%))
440 NEXT
450 PRINT'"Press a number"
460 *FX 15,1
470 REPEAT:K%=GET:K%=K%-48:UNTIL K%>0
AND K%<6
480 CLS
490 IF K%=C% PRINTTAB(0,5)"** CORRECT
**":SC%=SC%+1 ELSE PRINTTAB(2,4)"** WRO
NG **"' "The right answer"' "was ";T$(N%)
)
500 PROCdelay(200)
510 VDU 5:MOVE TX%(N%),TY%(N%):VDU 22
4,4
520 ENDPROC
530 REM *****
540 DEF PROCscore

```

```

550   VDU 28,0,24,12,2,12
560   PRINT'TAB(4)"You have"
570   PRINTTAB(5)"scored"
580   PRINT'TAB(7);SC%
590   IF Q%<10 PRINTTAB(7)"-" ELSE PRIN
TTAB(7)"--"
600   PRINTTAB(7);Q%
610   ENDPROC
620   REM *****
630   DEF PROCdelay(T%)
640   TIME=0:REPEAT UNTIL TIME>T%
650   ENDPROC
660   REM *****
670   DEF PROCfinish
680   VDU 28,23,11,39,2,12
690   PRINT"Your final score was ";SC%;
" out of 15"
700   PRINT'"Press any key to      play a
gain"
710   *FX 15,1
720   K%=GET
730   ENDPROC
740   REM *****
750   DEF PROCmap(X1%,Y1%,SIZE)
760   RESTORE 830
770   FOR L%=0 TO 260:READ X%(L%):NEXT
780   FOR L%=0 TO 260:READ Y%(L%):NEXT
790   MOVE X1%,Y1%
800   FOR L%=0 TO 260
810       PLOT 1,X%(L%)*SIZE,Y%(L%)*SIZE
820   NEXT:ENDPROC
830   REM *****X CO-ORDINATES*****
840   DATA 5,1,5,2,9,10,6,2,9,7,12,-1,3
850   DATA 10,-1,4,5,2,11,10,6,3,2,5,1
860   DATA 9,-2,-8,-1,-4,-2,-6,-2,-7,7
870   DATA 5,4,5,-1,-3,4,2,4,-2,4,4,2,0
880   DATA -5,-10,-8,-3,-3,-2,4,2,-7,-3
890   DATA -3,-11,0,11,2,6,-6,3,-6,-6,-
4
900   DATA -6,0,-1,-5,-3,-2,-7,-3,-9,-4
910   DATA -6,8,3,3,-6,-3,-5,5,7,6,2,5,
3
920   DATA -1,3,4,-3,-7,-1,-3,-13,-4,-1
,3
930   DATA -4,8,2,-6,-2,5,14,1,-7,-16,-

```

```

5
  940 DATA -5,-5,0,-4,1,-3,3,-4,1,-3
  950 DATA -2,2,0,-2,0,3
  960 DATA -3,0,-2,-1,2,-2,-5,-1,4,-1,5
  970 DATA 5,-5,-2,1,-3,1,-3,-1,6,-1,3,
0
  980 DATA 8,-5,-1,3,4,4,-3,-3,5,0,-8,0
  990 DATA -2,-3,3,4,1,2,2,5,0,2,9,4,10
 1000 DATA -4,-3,-5,-2,-1,3,3,-1,3,2,3,
2
 1010 DATA -3,1,-3,-1,4,-5,-1,4,-4,-2,0
,-3
 1020 DATA -6,-1,-10,-1,-9,5,7,1,-1,-2,
-5
 1030 DATA -10,-2,-7,0,3,-2,1,4,0,5,1,5
 1040 DATA 3,-3,0,6,4,2,2,1,8,0,3,1,4,6
,1
 1050 DATA -5,-7,-1,-7,-8,-8,-1,2,-9,-1
 1060 DATA -5,-5,-2,-7,-6,-1,5,1,2,0,2
 1070 REM *****Y CO-ORDINATES*****
 1080 DATA 3,3,1,4,1,-5,6,7,2,-5,0,3,-1
 1090 DATA 2,3,-3,0,-3,3,-4,4,0,2,-1,5,
3
 1100 DATA 9,-1,1,1,2,0,2,-1,1,1,0,4,3,
0
 1110 DATA 3,-1,3,2,0,4,9,10,7,3,0,-2,1
 1120 DATA 3,4,6,9,0,4,-1,2,1,-4,0,14,3
,3
 1130 DATA 12,0,2,5,9,15,7,0,12,0,4,-3,
3
 1140 DATA 4,0,4,3,-1,0,0,3,7,0,4,7,1,5
,5
 1150 DATA 7,0,-1,2,-4,-4,2,3,1,3,3,0,3
,0
 1160 DATA 15,7,1,-4,1,4,-5,-4,-1,-4,-2
,-3
 1170 DATA -2,1,-1,-5,-2,-3,-1,-6,0,-8,
-2
 1180 DATA 0,-5,-1,-2,0,-4,-2,-2,-3,2,-
8
 1190 DATA -6,-1,-3,-8,-7,-7,0,4,7,10,8
 1200 DATA -7,-7,-4,8,-2,-1,-6,-6,-4,-1
1
 1210 DATA -8,4,-1,-12,0,4,-1,-4,0,6,-3
 1220 DATA 2,6,0,-3,0,-10,0,-2,-7,-5,-3

```



1230 DATA -2,3,1,-1,-3,-3,-1,-5,-1,-3  
 1240 DATA -5,-4,3,-1,-2,2,-1,1,-3,-4  
 1250 DATA -7,-2,2,-6,-5,-6,-4,-3,1,-5  
 1260 DATA -2,0,-3,-2,0,-3,0,2,1,-1,-1  
 1270 DATA -5,0,1,-3,-1,-3,0,2,2,-1,1,7  
 , -1  
 1280 DATA -9,-4,-4,0,2,-1,-3,-1,0,-8,-  
 5  
 1290 DATA -2,-4,-6,-1,-6,0,1,-1,-2,-1  
 1300 REM \*\*\*\*\*TOWN DATA\*\*\*\*\*  
 1310 DATA Inverness,584,888,Aberdeen,6  
 40,860,Fort William,500,752,Perth,560,78  
 4  
 1320 DATA Glasgow,532,704,Edinburgh,57  
 6,716,Newcastle,680,604,Stranraer,480,58  
 0  
 1330 DATA Carlisle,624,576,Middlesbrou  
 gh,696,536,Kendal,616,524,York,700,468  
 1340 DATA Leeds,672,440,Preston,644,44  
 0,Hull,748,448,Liverpool,604,404  
 1350 DATA Manchester,644,412,Sheffield  
 ,664,400,Peterborough,764,292,Lincoln,73  
 6,376  
 1360 DATA Stoke,648,352,Nottingham,700  
 ,352,Shrewsbury,592,312  
 1370 DATA Norwich,852,296,Birmingham,6  
 64,284,Aberystwyth,512,284,Cambridge,792  
 ,264  
 1380 DATA Hereford,588,268,Colchester,  
 832,220,Oxford,696,212,Northampton,716,2  
 68  
 1390 DATA Cardiff,548,196,Carmarthen,4  
 96,236,London,752,168,Bristol,612,172  
 1400 DATA Guildford,728,124,Maidstone,  
 808,140,Dover,872,124,Barnstaple,500,144  
 1410 DATA Salisbury,644,124,Taunton,56  
 0,120,Southampton,688,100,Brighton,764,9  
 2  
 1420 DATA Exeter,520,96,Dorchester,608  
 ,96,Plymouth,476,60,Penzance,372,40



# AIRCRAFT LENDER

By Paul Bailey

This program puts you in the position of a pilot faced with landing an aircraft safely on the airport runway. By using different control keys the plane has to be guided down at the right speed, and at the right time, to ensure safe arrival of all aboard.

A continuous display provides a pilot's-eye view of the approaching runway. In addition, a number of instrument readings give essential details of such things as altitude, speed direction, pitch, roll and distance from runway. Keys may be used to manoeuvre the aeroplane's speed, roll and pitch, the pilot's efforts being reflected in the instrument readings. Realistic sound effects accompany the simulation, which will soon discover whether you can say with honesty "I'll take good care of you!"

## Flying tips

The distance reading indicates the distance you are from the end of the runway. As the runway is 2000m long you must touch down when this reads 2000m or less. At touchdown, yaw speed must be between 45m/s and 55m/s. Your pitch and roll must be less than 3 degrees and since the runway is 100m wide, the deviation readout must be within +/- 150m. After touchdown, you can use the rudder to correct the direction of travel, and your brakes to slow down to zero.

Your instruments give the following information:

**SPEED** airspeed in metres per second (m/s).

**BEAR** bearing is the direction in which yaw aircraft pointing: 0 = straight ahead, positive values to the right, negative to the left (in degrees).

**PITCH** negative values when the aircraft's nose is pointing down, positive when up.

**ROLL** roll causes the banking to change, which alters the aircraft's bearing; positive values clockwise, negative clockwise.

ALT height above ground in metres.

DIST distance from the far end of the runway in metres.

DEV sideways deviation from the centre of the runway.

## Controls

F = Roll to left

G = Maintain roll

H = Roll to right

M = Nose up

K = Maintain pitch

O = Nose down

0-9 = Throttle control

## Program Description

**10-250** Main program which calls a number of procedures to read user inputs, perform calculations and change the screen display. Landings, crash or otherwise, produce different endings to the simulation.

**260-390** Procedures to help with screen displays.

**400-870** Initialization procedure. Asks user for wind speed and direction, sets a variety of constants to their starting values, outputs display heading.

**880-920** Procedure to calculate aircraft velocity in X, Y directions.

**930-1030** Procedure to interpret in-flight control keys

**1040-1100** Procedure to interpret control keys during landing.

**1110-1390** Procedures to calculate relative position of aircraft and effect its movement

**1400-1670** Procedure to update screen display - produces new picture of runway and new values for instrument readings.

**1680-1920** Procedures to check for, and report on crash landings.

**1930-1960** Procedure to report a safe landing - most unlikely to needed!

**1970-2070** Procedures to display information on final position of aircraft and to produce variable length delay.

**2080-2470** Procedure to display operating instructions.

**2480-2590** Procedure, and data, for safe landing fanfare!

```

10  REM AIRCRAFT LANDER
20  MODE 7
30  PROCscreen
40  PRINTTAB(0,10)"Do you require ins
tructions?";
50  REPEAT IN$=GET$:UNTIL IN$="Y" OR
IN$="N"
60  IF IN$="Y" PROCins
70  MODE 4
80  PROCsetup
90  REPEAT
100  PROCkeys
110  PROCupdate_pos
120  PROCupdate_screen
130  PROCkeys
140  UNTIL PZ<=0
150  PROCcrash_check
160  IF LF=1 THEN PROCcrash:PROCend:EN
D
170  PRINTTAB(16,4)"TOUCHDOWN";
180  PROCland_keys
190  REPEAT
200  PROCmove
210  PROCupdate_screen
220  UNTIL LF=2 OR LF=3
230  IF LF=2 PROCoff_runway:PROCend
240  IF LF=3 PROCgood_land
250  END
260  DEFPROCnext_page
270  PRINT TAB(0,20)"Press any key to
continue";
280  *FX 15,0
290  KEY=GET
300  CLS
310  ENDPROC
320  DEFPROCscreen
330  PRINT TAB(3,0)CHR$132;CHR$157;TAB
(36,0);CHR$156
340  FOR Y%=1 TO 2
350  PRINT TAB(3,Y%)CHR$132;CHR$157;
CHR$134;CHR$141;"AIRCRAFT LANDING SIMULA
TOR ";CHR$156
360  NEXT
370  PRINT TAB(3,3)CHR$132;CHR$157;TAB
(36,3);CHR$156

```

```

380   VDU28,0,24,39,4
390   ENDPROC
400   DEFPROCsetup
410   DIM Q(11,3),R(11,2),TS(9)
420   C1=PI/180:C2=180/PI
430   C5=.5
440   LF=0:REM LANDING FLAG
450   PZ=3000:REM ALTITUDE
460   PY=-2E4:REM DISTANCE FROM RUNWAY
470   PX=0:REM LATERAL DEVIATION
480   RL=0:REM ROLL
490   PT=0:REM PITCH
500   AS=150:REM AIRSPEED
510   AD=0:REM AIR DIRECTION
520   PROCcalcvel
530   CLS
540   PRINT"Please select wind speed an
d direction. A direction of 0 means that
the wind is blowing straight at you and
this gives the easiest landing since y
ou do not need to alter roll and bear
ing. The"
550   PRINT"direction should be between
-90 (from the left) and +90 (from the
right). A wind speed of 5 is a light
breeze, a speed of 30 a gale."
560   REPEAT
570     PRINTTAB(0,16)STRING$(39," ")
580     INPUTTAB(0,16)"WIND SPEED (M/S)
",X0
590     UNTIL X0>0 AND X0<35
600   REPEAT
610     PRINTTAB(0,18)STRING$(39," ")
620     INPUTTAB(0,18)"WIND DIRECTION (
DEG)",X1
630     UNTIL X0>=-90 AND X0<=90
640     WY=-X0*COS(X1*C1)
650     WX=-X0*SIN(X1*C1)
660     VZ=0
670     GZ=VZ:GY=VY+WY:GX=VX+WX
680     TC=5:REM THROTTLE CONTROL
690     BC=0:REM BANKING CONTROL
700     NC=0:REM NOSE UP/DOWN CONTROL
710     RC=0:REM RUDDER CONTROL
720     FD=0:REM BRAKES

```

```

730   FOR I=0 TO 9:READ TS(I):NEXT
740   DATA 0,25,50,75,100,150,200,250,2
75,300
750   YT=20:REM YAW TOLERANCE
760   RT=3:REM ROLL TOLERANCE
770   TP=5:REM PITCH TOLERANCE
780   LR=2000:REM RUNWAY LENGTH
790   WR=50:REM RUNWAY WIDTH
800   HD=3E4:REM DISTANCE TO HORIZON
810   CLS
820   VDU 23,224,255,255,255,255,25
5,255,255
830   VDU 23;10,32;0;0;0;
840   PRINT"SPEED BEAR PITCH ROLL ALT
DIST    DEV"
850   PRINTTAB(0,4)STRING$(39,CHR$224)
860   PRINTTAB(0,31)STRING$(39,CHR$224)
;
870   ENDPROC
880   DEFPROCcalcvel
890   RA=AD*C1
900   VX=AS*SIN(RA)
910   VY=AS*COS(RA)
920   ENDPROC
930   DEFPROCkeys
940   X$=INKEY$(0)
950   IF ASC(X$)>47 AND ASC(X$)<58 THEN
TC=VAL(X$)
960   IF X$="F" BC=-1
970   IF X$="G" BC=0
980   IF X$="H" BC=1
990   IF X$="M" NC=1
1000  IF X$="K" NC=0
1010  IF X$="O" NC=-1
1020  *FX 15 1
1030  ENDPROC
1040  DEFPROCland_keys
1050  X$=INKEY$(0)
1060  IF X$="R" RC=1
1070  IF X$="E" RC=-1
1080  IF X$="C" RC=0
1090  IF X$="B" FD=2
1100  ENDPROC
1110  DEFPROCupdate_pos
1120  PZ=PZ+GZ

```

```

1130 PY=PY+GY
1140 PX=PX+GX
1150 PT=PT+NC
1160 RL=RL+BC
1170 AS=AS+2*SGN(TS(TC)-AS)
1180 AD=AD+RL
1190 VZ=1*(TC-5)+AS*SIN(PT*C1)
1200 PROCcalcvel
1210 GZ=VZ
1220 GY=VY+WY
1230 GX=VX+WX
1240 GD=-ATN(VX/VY)*C2
1250 SOUND 0,-10,7,15
1260 IF AS<155 SOUND 1,0,AS+100,15 ELS
E SOUND 1,0,255,15
1270 ENDPROC
1280 DEFPROCmove
1290 PT=0:RL=0:PZ=0
1300 PROCland_keys
1310 PX=PX+VX
1320 PY=PY+VY
1330 IF ABS(PX)>WR OR PY>0 LF=2
1340 AD=AD+AD*(SGN(AD)*SGN(RC))
1350 AS=AS-FD:IF AS<=0 THEN LF=3
1360 SOUND 0,-10,6,10
1370 IF AS<155 SOUND 1,0,AS+100,15 ELS
E SOUND 1,0,255,15
1380 PROCcalcvel
1390 ENDPROC
1400 DEFPROCupdate_screen
1410 HT=-PZ-5:LD=-PX-WR:RD=-PX+WR
1420 HD=HT*7:IF HD<-PY THEN HD=-PY
1430 IF HD<3000 THEN HD=3000
1440 FOR I=0 TO 4
1450 YD=-PY-I*500
1460 Q(I,1)=LD:Q(I+5,1)=RD
1470 Q(I,2)=YD:Q(I+5,2)=YD
1480 Q(I,3)=HT:Q(I+5,3)=HT
1490 NEXT
1500 Q(10,1)=-PX:Q(10,2)=-PY:Q(10,3)=H
T
1510 Q(11,1)=-PX:Q(11,2)=-PY-LR:Q(11,3
)=HT
1520 A1=64/WR:A2=64.5:A3=15.5:A4=31
1530 FOR I=0 TO 11

```



```

1540      A0=(HD-Q(I,2))/HD
1550      R(I,1)=INT(Q(I,1)*A0*A1+A2)
1560      R(I,2)=INT(A3+A0*A4)-3
1570      NEXT
1580      VDU28,0,30,39,5,12,26
1590      MOVE R(4,1)*10,1024-(R(4,2)*21.7)
1600      DRAW R(0,1)*10,1024-(R(0,2)*21.7)
1610      DRAW R(5,1)*10,1024-(R(5,2)*21.7)
1620      DRAW R(9,1)*10,1024-(R(9,2)*21.7)
1630      DRAW R(4,1)*10,1024-(R(4,2)*21.7)
1640      MOVE R(10,1)*10,1024-(R(10,2)*21.
7)
1650      PLOT 21,R(11,1)*10,1024-(R(11,2)*
21.7)
1660      PRINT TAB(0,0);TAB(0);INT(AS+C5);
TAB(6);INT(AD+C5);TAB(11);INT(PT+C5);TAB
(17);INT(RL+C5);TAB(22);INT(PZ+C5);TAB(2
8);INT(ABS(PY)+C5);TAB(35);INT(PX+C5);"
";
1670      ENDPROC
1680      DEFPROCcrash_check
1690      IF ABS(RL)>RT OR PT>TP OR PT<0 OR
ABS(AD)>YT OR AS<45 OR AS>55 THEN LF=1:
PRINTTAB(12,4)"CRASH ON LANDING";
1700      IF ABS(PX)>WR OR PY>0 OR PY<-LR T
HEN LF=1:PRINTTAB(15,5)"OFF RUNWAY";
1710      IF LF=1 PROCdelay(500)
1720      ENDPROC
1730      DEFPROCcrash
1740      CLS
1750      VDU19,1,8;0;
1760      PRINTTAB(15,15)"FATAL CRASH"
1770      *FX 15,1
1780      LOADADVAL=GET
1790      CLS
1800      V/19,1,7;0;
1810      PRINT"THE FLIGHT RECORDER HAS BEE
N RECOVERED"
1820      PRINT"THE FINAL FLIGHT DETAILS WE
RE AS FOLLOWS"
1830      ENDPROC
1840      DEFPROCoff_runway
1850      CLS
1860      IF PY>0 THEN PRINT"YOU HAVE GONE
OFF THE END OF THE RUNWAY.TRY USING YOUR

```

```

BRAKES NEXT TIME."
1870 IF ABS(PX)>WR THEN PRINT"YOU ARE
OFF THE SIDE OF THE RUNWAY.      ISN'T IT
WIDE ENOUGH FOR YOU?"
1880 IF AS<=10 THEN PRINT"LUCKILY THER
E IS ONLT MINOR DAMAGE AT      THIS SPEED."
1890 IF AS>10 AND AS<=25 THEN PRINT"SE
RIOUS DAMAGE TO YOUR AIRCRAFT, BUT YOUSU
RVIVE TO SCARE YOUR PASSENGERS TO      DE
ATH AGAIN."
1900 IF AS>25 THEN PRINT'"DISASTER - F
ATAL CRASH"
1910 PRINT'"DETAILS OF THE FLIGHT WHEN
YOU LEFT THE RUNWAY:-"
1920 ENDPROC
1930 DEFPROCgood_land
1940 PRINT TAB(0,4)"CONGRATULATIONS CA
PTAIN - A SAFE LANDING"
1950 PROCTune
1960 ENDPROC
1970 DEFPROCend
1980 PRINT"SPEED = ";AS;" M/S"
1990 PRINT"DIRECTION = ";AD;" DEG"
2000 PRINT"DISTANCE FROM END OF RUNWAY
= ";ABS(PY);" M"
2010 PRINT"DISTANCE FROM CENTRE OF RUN
WAY = ";ABS(PX);" M"
2020 PRINT"PITCH = ";PT;" DEG"
2030 PRINT"ROLL = ";RL;" DEG"
2040 ENDPROC
2050 DEFPROCdelay(T)
2060 TIME=0:REPEATUNTILTIME=T
2070 ENDPROC
2080 DEFPROCins
2090 CLS
2100 PRINT'"It is dark and raining. Yo
u are      approaching the airport wi
th 300      passengers on board and yo
ur co-pilot  has just become unconsciou
s, so you mustland the aircraft alone. T
o be honest"
2110 PRINT"he was not doing a great jo
b before he  passed out."
2120 PRINT'"You are in level flight at
an altitude  of 3000 M but only 20 km f

```

rom the far end of the runway. Your aircraft is pointing in the correct direction but"

2130 PRINT"any crosswind will cause a deviation from this path."

2140 PROCnext\_page

2150 PRINT"You must land with an airspeed of about 50 M/S with no appreciable roll or pitchRoll must be less than 3 degrees and pitch between 0 and 5 degrees. Since the runway is 2000 M long, the distance"

2160 PRINT"indicator must show less than this""figure on touchdown. Similarly the""runway is 100 M wide so that if the""deviation is greater than 50 M on""touchdown, you will miss the runway."

2170 PRINT""After touchdown you must use the rudder to correct the direction of travel and the brakes to stop before you reach the end of the runway (this is indicated by"

2180 PRINT"a distance reading of 00. If you fail to stop or correct your direction (which probably will not be 0 when landing in a crosswind) you will crash."

2190 PROCnext\_page

2200 PRINT"You can see the runway lights ahead and in the distance the lights of the city you hope to avoid."

2210 PRINT""Your instruments give the following:"

2220 PRINT"SPEED airspeed in M/S"

2230 PRINT"BEAR bearing - direction in which aircraft is pointing. 0 is""straight ahead positive values to the right."

2240 PRINT"PITCH positive values when aircraft's nose is up."

2250 PRINT"ROLL positive values clockwise. Roll causes banking which changes the aircraft's bearing."

2260 PRINT"ALT height above ground in M."

```

2270 PRINT"DIST distance from the far
end of the runway in M."
2280 PRINT"DEV sideways deviation fr
om the centre of the runway."
2290 PROCnext_page
2300 PRINT"In flight you have the foll
owing controls:"
2310 PRINT'"F roll to left"
2320 PRINT"G maintain roll at this le
vel"
2330 PRINT"H roll to right"
2340 PRINT"M nose up (increase pitch)
"
2350 PRINT"K maintain pitch at this l
evel"
2360 PRINT"O nose down (decrease pitc
h)"
2370 PRINT'"0-9 throttle control"
2380 PRINT" Each setting has a term
inal value of speed which is reached
eventually. e.g. 5 - 150 M/S, 2 - 5
0 M/S. Landing is usually achi
eved on setting 2."
2390 PROCnext_page
2400 PRINT'"After touchdown only the f
ollowing have effect:"
2410 PRINT'"E rudder left"
2420 PRINT"R rudder right"
2430 PRINT"B apply brakes"
2440 PRINT'"N.B. All controls have a d
elayed effect on a large aircraft. In ge
neral they have effect on the second
display after pressing the key."
2450 PRINT'"GOOD LUCK"
2460 PROCnext_page
2470 ENDPROC
2480 DEFPROCTune
2490 RESTORE 2560
2500 FOR notes%=1 TO 44
2510 READ P%,D%
2520 SOUND 1,-15,P%,D%*1.5
2530 SOUND 1,0,0,1
2540 NEXT
2550 ENDPROC
2560 DATA 33,2,41,2,49,2,53,4,33,5,53,

```

2, 49, 2, 53, 2, 61, 4, 41, 5  
2570 DATA 41, 2, 49, 2, 53, 2, 69, 2, 61, 2, 61,  
2, 53, 2, 53, 2, 49, 2, 41, 2, 49, 2, 33, 4  
2580 DATA 33, 2, 41, 2, 49, 2, 53, 4, 33, 5, 53,  
2, 49, 2, 53, 2, 61, 4, 41, 5  
2590 DATA 41, 2, 49, 2, 53, 2, 69, 2, 61, 2, 61,  
2, 53, 2, 53, 2, 49, 2, 41, 2, 49, 2, 53, 4

# MICRO-MOUSE

By Michael Ashwin

Your micro controls a mouse that has to find its way out of a maze that you create.

To begin with you draw a maze on the screen using the cursor keys to control the direction and SHIFT to leave a trail. You can direct your maze anywhere you like but it must start in the top left corner and finish at the bottom right corner of the screen. When your meandering masterpiece is finished it is then the computer's turn. It has to direct the mouse, again- from the top left corner to the bottom right corner, in as few moves as possible. Can you create a labyrinth from which the mouse will never escape?

## Program Description

**100-150** Main program calling the procedures listed below.

**160-400** PROCMOVE - procedure for moving mouse. The logic for change of direction is contained in lines 240-290.

**410-500** PROCSETVAR - sets up variables and user-defined characters.

**510-610** PROCMAZE - main routine to draw maze in response to cursor key directions.

**620-700** PROC PROMPT - routine to display a prompt and wait for a response.

**710-820** PROC COMPUTER - displays maze as the computer sees it. Prints contents of array Z%(B%,A%).

**830-890** PROC DISMAZE - displays maze.

**900-1030** PROC INIT prints instructions.

**1040-1130** PROC DEMO - demonstration mode where a maze is created randomly.

---

```
10  GOTO 30
20  ON ERROR RUN
30  MODE 7
```

```

40  PROCINIT
50  PROCSETVAR
60  MODEL
70  PROCDIS:IF A$="2" THEN PROCDEMO:G
OTO90
80  PROCMAZE
90  PROC_PROMPT
100 IF A=32 THEN PROCCOMPUTER
110 PROCDISMAZE
120 PROCMOVE
130 IF INKEY$(400)=" " THEN RUN
140 RUN
150 END
160 DEFPROCMOVE
170 C1%=C%+1:X%=2:Y%=2:X1%=X%:Y1%=Y%:
count=0:A%=1:IF C1%=5 THEN ENDPROC
180 IF A%=3 THEN Z%(Y%,X%)=2:T%=0 ELSE
E T%=10
190 IF X1%=31 AND Y1%=26 THEN ENDPROC
200 TIME=0:REPEATUNTILTIME>T%
210 PRINTTAB(X1%+4,Y1%+1) ". "
220 GOES=GOES+1:PRINTTAB(10,29) "Number
of goes ";GOES
230 X1%=X%:Y1%=Y%:PRINTTAB(X%+4,Y%+1)
;CHR$(C%)
240 IF Z%(Y%,X%+1)=1 OR Z%(Y%+1,X%)=1
OR Z%(Y%-1,X%)=1 OR Z%(Y%,X%-1)=1 THEN
A%=1
250 IF Z%(Y%+1,X%)=A% THEN Y%=Y%+1:Z%
(Y%,X%)=3:A%=1:count=0:C%=241:GOTO180
260 IF Z%(Y%,X%+1)=A% THEN X%=X%+1:Z%
(Y%,X%)=3:A%=1:count=0:C%=242:GOTO190
270 IF Z%(Y%-1,X%)=A% THEN Y%=Y%-1:Z%
(Y%,X%)=3:A%=1:count=0:C%=243:GOTO190
280 IF Z%(Y%,X%-1)=A% THEN X%=X%-1:Z%
(Y%,X%)=3:A%=1:count=0:C%=244:GOTO190
290 IF X%=X1% AND Y%=Y1% THEN count=c
ount+1:A%=3:Z%(Y%,X%)=2:IF count>10 THEN
PRINTTAB(X%+4,Y%+1) " ":GOTO180 ELSE 190
300 X%=X1%:Y%=Y1%:GOTO190
310 DEFPROCDIS:CLS
320 VDU23;8202;0;0;0;
330 COLOUR1
340 PRINT'SPC(15)"Micro-Mouse"
350 PRINTSPC(5);STRING$(32,CHR$(240))

```

```

360   FOR A=1 TO 25:PRINTSPC(5);CHR$240
;SPC(30);CHR$240:NEXTA
370   PRINTSPC(5);STRING$(32,CHR$240)
380   COLOUR2
390   FOR A=1 TO 25:PRINTTAB(6,(A+2));S
TRING$(30,CHR$240):NEXTA
400   ENDPROC
410   DEFPROCSETVAR
420   DIM Z%(27,32):C1%=0
430   VDU23,240,255,255,255,255,255
,255,255
440   VDU23,241,60,60,60,60,255,126,60,
24
450   VDU23,242,24,28,254,255,255,254,2
8,24
460   VDU23,243,24,62,126,255,60,60,60,
60
470   VDU23,244,24,56,127,255,255,127,5
6,24
480   FOR A=1 TO 32:Z%(1,A)=0:Z%(27,A)=
0:NEXTA
490   FOR A=1 TO 27:Z%(A,1)=0:Z%(A,32)=
0:NEXTA
500   ENDPROC
510   DEFPROCMAZE
520   X=2:Y=2:Z=0
530   X1=X:Y1=Y
540   IF INKEY(-58) AND Y>2 THEN Y=Y-1
550   IF INKEY(-42) AND Y<26 THEN Y=Y+1
560   IF INKEY(-122) AND X<31 THEN X=X+
1
570   IF INKEY(-26) AND X>2 THEN X=X-1
580   IF Z=1 THEN Z%(Y,X)=1:PRINTTAB(X+
4,Y+1)"X"TAB(X1+4,Y1+1)" " ELSE COLOUR3:
PRINTTAB(X+4,Y+1)"X":COLOUR2:PRINTTAB(X1
+4,Y1+1);CHR$240:Z%(Y,X)=0
590   IF INKEY(-1) THEN Z=1 ELSE Z=0
600   IF INKEY(-74) THEN ENDPROC
610   GOTO530
620   DEFPROC_PROMPT
630   COLOUR130:COLOUR4
640   PRINTTAB(8,7)"Press space to see
how the"
650   PRINTTAB(9,8)"computer sees the m
aze"

```



```

660 COLOUR3:PRINTTAB(20,10)"OR"
670 COLOUR1:PRINTTAB(10,12)"Press any
key to begin"
680 IF INKEY$(0)="" THEN A=GET ELSE 6
80
690 COLOUR128:COLOUR3
700 ENDPROC
710 DEFPROC COMPUTER
720 PRINTTAB(15,29)"0 = Wall"
730 PRINTTAB(15,30)"1 = Passage"
740 COLOUR2
750 FOR A%=1 TO 32:FOR B%=1 TO 27
760     PRINTTAB(A%+4,B%+1);Z%(B%,A%)
770     NEXTB%,A%
780     PRINTTAB(0,29);STRING$(75," ")
790     PRINTTAB(11,29)"Press space to
begin"
800     REPEATUNTILGET=32
810     PRINTTAB(11,29)SPC(20)
820     ENDPROC
830     DEFPROC DISMAZE
840     FOR A%=1 TO 32:FOR B%=1 TO 27
850         IF A%=1 OR A%=32 OR B%=1 OR
B%=27 THEN COLOUR1 ELSE IF Z%(B%,A%)=1
THEN COLOUR3 ELSE COLOUR2
860         PRINTTAB(A%+4,B%+1);CHR$240
870         NEXTB%,A%
880         COLOUR131:COLOUR4
890         ENDPROC
900         DEFPROC INIT
910         FORA=1 TO 2:PRINTCHR$134;CHR$
157;CHR$132;CHR$141;SPC(10);"Micro-Mouse
":NEXTA
920         PRINT'"This program illustrat
es the wanderings of a mouse around a ma
ze that you have created."
930         PRINT'"The size and shape of
maze is up to you,however, the more diff
icult the maze thelonger it takes the mo
use to escape."
940         PRINT'"You draw the maze usin
g the cursor control keys. With the
se you control a flashing cursor that w
ill leave a path if the SHIFT key is pr
essed at the same time."

```

```

950
960      PRINT'" Press the space bar
to continue"
970      REPEATUNTILGET=32
980      CLS:FORA=1 TO 2:PRINTCHR$134;
CHR$157;CHR$132;CHR$141;SPC(10);"Micro-M
ouse":NEXTA
990      PRINT'"You may make a trail w
herever you wish, but it must begin at t
he top left and end at the bottom righ
t of the screen."
1000     PRINT'"When you have finished
the maze, press RETURN."
1010     PRINT'"Try to devise a maze t
hat you think will fool the micromou
se."
1020     PRINT'"PRESS 1) to start
                        2) for demonstr
ation maze"
1030     A$=GET$:IF A$="1" OR A$="2" T
HEN ENDPROC ELSE GOTO1030
1040     DEFPROCDEMO:PRINTTAB(9,5)"PLE
ASE WAIT A FEW SECONDS":FORA=2 TO 31:Z%(
2,A)=1:Z%(26,A)=1:NEXTA:B=RND(29+2):FORA
=2 TO 26:Z%(A,B)=1:NEXTA:FORA=1 TO 20
1050     X=RND(29)+2:Y=RND(24)+2:D=R
ND(2):L=RND(20)
1060     REPEAT:L=L-1
1070     Z%(Y,X)=1
1080     IF D=1 AND Y<25 THEN Y=Y+
1:IF Y=26 THEN L=0
1090     IF D=2 AND Y<30 THEN X=X+
1:IF X=31 THEN L=0
1100
1110     UNTIL L=0
1120     NEXTA
1130     ENDPROC

```

# TRANK

By W.G.T. Walker

Trank is an original zapping game involving Herman, the triangular tank, and a wandering gaily coloured target. The aim is to shoot as many targets as possible, without running into them, the walls or the obstacles. Herman can be controlled by either the keyboard or two joysticks, and the method of control is a little different to most games. If you are using joysticks the left stick controls his left caterpillar track and the right stick controls his right caterpillar track, so if both sticks are fully forward he will go forward at top speed, and if they are both back he will go in reverse at top speed. If one stick is forward and one back he will turn. Using the keyboard is similar:

A - Left track forward  
Z - Left track backwards  
\* - Right track forward  
? - Right track backwards  
SPACE - Fire

## Program Description

**20** Change mode and turn off the cursor.  
**30-70** Print the instructions and see if the keyboard or the joysticks are to be used.  
**80-120** Draw a pattern.  
**110-140** Set up the explosion arrays.  
**150-110** Define the characters for the shot and the target.  
**190** Define the envelope for the shot.  
**210** Wait until a key or the joystick fire button is pressed.  
**220** Start of the main game loop.  
**230** Change mode and turn off the cursor.  
**250** Draw the landscape.  
**260** Reset some variables.  
**270** Draw Herman.

**280-290** Reset some variables.

**300** The game loop. Play the game until Herman hits something.

**310** Blow up Herman.

**320-380** Display the player's score.

**390** Wait for a key or the fire button to be pressed.

**400** Return to a new game.

**420-410** PROCTANK - check the keyboard or the joysticks to update Herman's position.

**490** PROC CONV - check to see if Herman has hit anything.

**510** PROCTK - draw Herman on the screen.

**530-640** PROC DIE - make an explosion.

**660-800** PROCTERRAIN - draw a border around the screen and put obstacles on it.

**820-840** PROC SEED - Find a random position.

**860-910** PROC MISSILE - check the fire button and if a missile has been fired move it one position and check for a hit

**930-980** PROC FIRE - set up the variables for a missile.

**1000** PROC DOBJ print the target object on the screen.

**1020-1010** PROC INITOBJ - set up the variables for the object

**1090-1200** PROC OBJ -- move the object and see if it has hit anything.

**1220-1230** PROC JOYINST - print the instructions for using joysticks.

**1250-1260** PROC KEYINST print the instructions for using the keyboard.

---

```
10 REM TRANK BY Bill Walker (C) W G T
Walker 1983
20 MODE 1:VDU23;10,32;0;0;0
30 PRINTTAB(17)"TRANK":COLOUR2:PRINT'
TAB(12)"by Bill Walker."
40 PRINT'"Steer Herman the triangular
tank around"' the range, avoiding the
obstacles and"'shooting the wandering
gaily coloured"' target"
50 COLOUR1:PRINTTAB(3,12)"Do you want
to use the keyboard or"TAB(3,14)"joysti
cks (J/K) ?":COLOUR2:REPEAT:K$=GET$:UNTI
```

```

LK$="J"ORK$="K"
60 IFK$="J"PROCJOYINST ELSEPROCKEYINS
T
70 PRINTTAB(3,12)SPC(35)TAB(3,14)SPC(
18):VDU5
80 VDU29,640;512;
90 DIM S% 359, C% 359
100 VDU29,640;512;;GCOL3,1
110 FORI%=0TO359STEP5:I=RAD(I%):S%?I%=
SIN(I)*127+128:C%?I%=COS(I)*127+128
120 MOVE0,0:DRAW 3*(S%?I%-128),(C%?I
%-128):NEXT
130 DIMEX%(15),EY%(15),DEX%(15),DEY%(1
5)
140 FORI%=0TO15:DEX%(I%)=5*RND(19)-50:
DEY%(I%)=5*RND(19)-50:NEXT
150 VDU23,254,0,0,&18,&3C,&3C,&18,0,0
160 VDU23,243,&E7,&FF,&FF,&7E,&7E,&FF,
&FF,&E7
170 VDU23,252,&80,&2E,&78,&4C,&0C,&5C,
&62,&00
180 DIMQ%(4):Q%(0)=1:Q%(1)=3:Q%(2)=5:Q
%(3)=6:Q%(4)=7:Q%=0
190 ENVELOPE1,1,6,0,-6,1,1,1,5,-2,0,-1
,30,10
200 *FX 15,1
210 GCOL0,3:MOVE-350,16:PRINT"PRESS AN
Y KEY TO START":REPEAT:UNTILINKEY(0)>0OR
ADVAL(0)AND3
220 REPEAT
230 MODE 1:VDU23;10,32;0;0;0
240 COLOUR129
250 PROCTERRAIN
260 XP=620:YP=512:VL=0:VR=0
270 S=0:C=1:PROC CONV:PROCTK
280 DIR=0
290 K%=0:M%=0:O%=0:SCORE%=0:TIME=0
300 REPEAT:PROCTANK:PROCMISSILE:PROC
OBJ:VDU4,30:PRINTSPC18;SCORE%;" ";:UNT
ILK%
310 PROCTK:PROCDIE
320 MODE1:COLOUR3:VDU19,1,13;0;23;10
,32;0;0;0
330 PRINT''TAB(15)"YOU SCORED"
340 PRINT'TAB(12)"THE GRAND TOTAL"

```

```

350     COLOUR1:PRINT'TAB(10) SCORE%
360     COLOUR3:PRINT'TAB(16) "AND  TOOK"
370     COLOUR1:PRINT'TAB(10) TIME  DIV100
;:COLOUR3:PRINT"  seconds"
380     PRINT' ' '      PRESS ANY KEY FOR AN
OTHER GO...":*FX15,1
390     REPEAT:UNTILINKEY(0)>0ORADVAL(0)
AND3
400     UNTIL FALSE
410
420     DEFPROCTANK:IFK$="J"VL=ADVAL(2)/10
24-32:VR=ADVAL(4)/1024-32
430     IFK$="K"ANDABS(VL)<=32  VL=VL+(1*IN
KEY-98)-(1*INKEY-66)
440     IFK$="K"ANDABS(VR)<=32  VR=VR+(1*IN
KEY-105)-(1*INKEY-73)
450     IFVL>32VL=32ELSEIFVL<-32VL=-32
460     IFVR>32VR=32ELSEIFVR<-32VR=-32
470     SOUND&12,1,ABS(VL),9:SOUND&13,1,AB
S(VR),9:D=(VL+VR)DIV4:DIR=(DIR+(VL-VR)/4
+360)MOD360:S=(S%?DIR-128)/127:C=(C%?DIR
-128)/127:PROCTK:XP=XP+D*S:YP=YP+D*C:PRO
CCONV:PROCTK:ENDPROC
480
490     DEFPROCCONV:VDU29,XP;YP;:T%=-20*(S
+C):U%=20*(S-C):V%=30*S:W%=30*C:K%=POINT
(T%,U%)ORPOINT(V%,W%)ORPOINT(-U%,T%):END
PROC
500
510     DEFPROCTK:VDU29,XP;YP;25,4,T%;U%;2
5,4,V%;W%;25,86,-U%;T%;:ENDPROC
520
530     DEFPROC DIE
540     GCOL3,2:VDU19,2,7;0;:FORJ%=0TO14
550         SOUND&10,J%-15,6,8
560         IFJ%=9THENVDU19,2,1;0;
570         IFJ%=5THENVDU19,2,3;0;
580         FORI%=0TO15
590             PLOT69,EX%(I%),EY%(I%)
600             EX%(I%)=EX%(I%)+DEX%(I%):EY%(I
%)=EY%(I%)+DEY%(I%)
610             PLOT69,EX%(I%),EY%(I%)
620             NEXT:NEXT
630     FORI%=0TO15:PLOT69,EX%(I%),EY%(I%)
:EX%(I%)=0:EY%(I%)=0:NEXT

```

```

640  ENDPROC
650
660  DEFPROCTERRAIN
670  VDU19,1,2;0;:GCOL0,129:CLG
680  VDU24,30;30;1250;950;
690  GCOL0,128:CLG
700  VDU26
710  GCOL0,1
720  VDU29,640;512;
730  FORI%=0TO8
740      J1%=RND(2):IFJ1%=2THENJ1%=-1
750      J2%=RND(2):IFJ2%=2THENJ2%=-1
760      PROCSEED:MOVEJX%,JY%
770      PROCSEED:MOVEJX%,JY%
780      PROCSEED:PLOT85,JX%,JY%
790      NEXT
800  ENDPROC
810
820  DEFPROCSEED
830  REPEAT:JX%=RND(400)*J1%:JY%=RND(40
0)*J2%:UNTILJX%*JX%+JY%*JY%>60000
840  ENDPROC
850
860  DEFPROCMISSILE:IFM%=0AND(ADVAL(0)A
ND3 OR INKEY-99)PROCFIRE:ENDPROC
870  IFM%=0ENDPROC
880  VDU29,MX%;MY%;25,4,-16;16;18,3,3,5
,254:M%=M%-1:SOUND&11,-9,M%,6:IFM%=0ENDP
ROC
890  MX%=MX%+MDX%:MY%=MY%+MDY%:P%=POINT
(0,0):IFP%=0VDU29,MX%;MY%;25,4,-16;16;25
4:ENDPROC
900  IFP%=2THEN0%=0:SCORE%=SCORE%+1:PRO
CDOBJ:GCOL3,1:MOVE-16,16:VDU5,252:PROCDI
E
910  M%=0:ENDPROC
920
930  DEFPROCFIRE
940  MX%=XP+S*60:MDX%=S*32
950  MY%=YP+C*60:MDY%=C*32
960  M%=20
970  VDU29,MX%;MY%;:MOVE-16,16:GCOL3,3:
VDU5,254
980  ENDPROC
990

```

```

1000 DEFPROCDOBJ:VDU29,BX%;BY%;25,4,-16
;16;18,3,2,5,243:ENDPROC
1010
1020 DEFPROCINITOBJ
1030 VDU29,0;0;;REPEAT:BX%=RND(1170)+60
:BY%=RND(100)+60:UNTILPOINT(BX%,BY%)=0
1040 REPEAT:MBX%=RND(31)-16:MBY%=RND(31)
)-16:UNTILABS(MBX%*MBY%)>31
1050 O%=1
1060 PROCDOBJ
1070 ENDPROC
1080
1090 DEFPROC OBJ:VDU19,2,Q%(Q%);0;;Q%=(Q
%+1)MOD5:IF O%=0PROCINITOBJ
1100 PROCDOBJ:BX%=BX%+MBX%:BY%=BY%+MBY%
:VDU29,0;0;;IFPOINT(BX%,BY%)=0PROCDOBJ:E
NDPROC
1110 SOUND0,-4,0,1
1120 F%=1:L%=40
1130 REPEAT
1140     IFBX%<30ORBX%>1250MBX%=-MBX%:F%=
0
1150     IFBY%<30ORBY%>950MBY%=-MBY%:F%=0
1160     IFF%MBX%=RND(16)*2-16:MBY%=RND(1
6)*2-16
1170     L%=L%-1:IFL%<0PROCINITOBJ:PROCDO
BJ
1180     P%=POINT(BX%+MBX%,BY%+MBY%):UNTI
LP%=0ORP%=3
1190     BX%=BX%+MBX%:BY%=BY%+MBY%:PROCDOBJ
1200 ENDPROC
1210
1220 DEFPROCJOYINST:PRINTTAB(2,21)"The
left and right joysticks control"'''TAB(6)
)"Herman's caterpillar tracks:"'''Up=For
wards, Centre=Stop, Down=Backwards"
1230 PRINT"Hence to move forwards push
both sticks"'''TAB(6)"up, to turn right p
ull the"'''TAB(6)"right-hand stick down,
etc.";;ENDPROC
1240
1250 DEFPROCKEYINST:PRINTTAB(2,21)"The
following keys control Herman:-"'''TAB(6)
"A and Z Control the speed of"'''TAB(5)"h
is left caterpillar track."'''TAB(6)"* an

```



```
d ? Control the speed of'''TAB(5)"his ri
ght caterpillar track."
  1260 PRINT'TAB(9)"The SPACE BAR fires."
;:ENDPROC
```



# FOUR IN A LINE

By James Byme

This is a game for two players where the aim is to drop counters into a grid. The winner is the first one to get... you guessed it, four in a row.

So, this is a game of strategy and forethought, where you have to attempt to block your opponent's moves as well as plan ahead for your own.

In order to drop a counter you simply type the number of the column in which you wish it to go.

Although very simple, the graphics are extremely effective and give an impression of smooth movement as the counters fall into the grid.

## Program Description

**90** Set up the arrays for the players' names, their colours and the contents of the board.

**100** The start of the program loop,

**110** Change mode and go to procedure for instructions.

**120** Change mode.

**130** Set up the characters and variables.

**140** Draw the board.

**150** The start of the game loop.

**160** Add one to the number of goes.

**110** Work out which player's turn it is.

**180** Find out which column the player wants to drop the counter in, and work out how far it should be dropped.

**190** Drop the counter into the column.

**200** Put the number of the player into the board array at the counter's position.

**210** Check to see if the game has been won.

**220** If the game has not ended return to the game loop

**230-240** Check if a player has won, if there is a draw and if another game is wanted.

```
10 REM*****
20 REM**
```

```

30  REM**          FOUR IN A LINE          **
40  REM**          By James Byrne          **
50  REM**          March 1983              **
60  REM**                                          **
70  REM*****
80  ON ERROR PROCerror
90  DIM NAME$(2),COL%(2),BOARD%(8,7)
100 REPEAT
110     MODE 1:PROCinstructions
120     MODE 2
130     PROCsetup
140     PROCscreen
150     REPEAT
160         GO%=GO%+1
170         TURN%=GO% MOD 2+1
180         COLUMN%=FNinput(TURN%):DROP%=
FNdepth(COLUMN%)
190         PROCdropcounter(COLUMN%,DROP%
)
200         BOARD%(COLUMN%,DROP%)=TURN%
210         PROCcheckwin(COLUMN%,DROP%,TU
RN%)
220         UNTIL finished OR GO%=43
230         IF finished PROCwon(TURN%) ELSE
PROCdraw
240         UNTIL FNanothergame=FALSE
250     PROCend
260     END
270     REM*****
280     DEF PROCinstructions
290     CLS:VDU 23;8202;0;0;0;
310     COLOUR130:COLOUR1:PRINT TAB(13,1)
"FOUR IN A LINE":COLOUR3:COLOUR128
320     PRINT" " 'Four in a line' is a ga
me for two     players. The players take
it in turns    to drop counters in a grid
. The first    player to get four counter
s in a line    is the winner."
330     PRINT" " When the computer asks '
Which column?'you should type in the num
ber of the     column that you want to dr
op your        counter in and then press
RETURN."
360     COLOUR2:PRINT'"Player one is";:C
OLOUR1:PRINT" red."

```

```

370 COLOUR2:PRINT"Player two is";:COL
OUR3:PRINT" white."
380 COLOUR1:PRINT'" Please type in yo
ur name player one"
390 REPEAT
400 INPUTTAB(1,19)NAME$(1)
410 IF LEN(NAME$(1))>10 THEN PRINTT
AB(1,19)"PLEASE SHORTEN THAT";SPC(20):PR
OCwait(200):PRINTTAB(1,19);SPC(20)
420 UNTIL LEN(NAME$(1))<11
430 COLOUR3:PRINT'" Please type in yo
ur name player two"
440 REPEAT
450 INPUTTAB(1,22)NAME$(2)
460 IF LEN(NAME$(2))>10 THEN PRINTT
AB(1,22)"PLEASE SHORTEN THAT";SPC(20):PR
OCwait(200):PRINTTAB(1,22);SPC(20)
470 UNTIL LEN(NAME$(2))<11
480 PRINT'" Press the SPACE BAR to
begin game";
490 REPEAT UNTIL GET=32
500 ENDPROC
510 REM*****
520 DEF PROCsetup
530 VDU 23,224,0,7,15,31,31,63,63,63
540 VDU 23,225,0,224,240,248,248,252,
252,252
550 VDU 23,226,63,63,63,31,31,15,7,0
560 VDU 23,227,252,252,252,248,248,24
0,224,0
570 VDU 23,228,7,8,16,0,32,0,0,0
580 VDU 23,229,224,16,8,0,4,0,0,0
590 VDU 23,230,0,0,32,0,16,8,7,0
600 VDU 23,231,0,0,4,0,8,16,224,0
610 CIRCLE1$=CHR$224+CHR$225+CHR$10+C
HR$8+CHR$8+CHR$226+CHR$227
620 CIRCLE2$=CHR$228+CHR$229+CHR$10+C
HR$8+CHR$8+CHR$230+CHR$231
630 VDU 19,3,4,0,0,0,19,5,4,0,0,0
640 VDU 23;8202;0;0;0;
650 GO%=1:COL%(1)=1:COL%(2)=7:finishe
d=FALSE
660 FOR X%=1 TO 8
670 FOR Y%=1 TO 7
680 BOARD%(X%,Y%)=0

```

```

690         NEXT
700     NEXT
710 ENDPROC
720 REM*****
730 DEF PROCscreen
740     GCOL 0,132
750     VDU 24,80;128;1200;896;19,4,0,0,0
,0
760     CLG
770     GCOL 0,0
780     VDU 5
790     FOR ROW%=864 TO 160 STEP -128
800         FOR COLUMN%=96 TO 1056 STEP 160
810             MOVE COLUMN%,ROW%:PRINT CIRCL
E1$
820         NEXT
830     NEXT
840     VDU 26,28,0,31,19,29
850     X%=128:GCOL0,4
860     FOR NUMBER%=1 TO 7
870         MOVE X%,960:PRINT ;NUMBER%
880         X%=X%+160
890     NEXT
900     VDU 19,4,4,0,0,0
910 ENDPROC
920 REM*****
930 DEF FNinput (TURN%):VDU 4,12:COLOU
R COL%(TURN%):*FX 15,1
940     PRINT TAB(0,29)"WHICH COLUMN
"
950     PRINT TAB(0,30)NAME$(TURN%);:INPU
T COLUMN%
960     IF COLUMN%<1 OR COLUMN%>7 OR COLU
MN%<>INT(COLUMN%) THEN CLS:PRINT TAB(0,2
9)"    NO SUCH COLUMN";CHR$7:PROCwait(200
):=FNinput (TURN%)
970     IF FNdepth(COLUMN%)>6 THEN PRINTT
AB(0,29)" COLUMN ALREADY FULL";CHR$7:PRO
Cwait(200):=FNinput (TURN%)
980     VDU 5:=COLUMN%
990     REM*****
1000    DEF PROCdropcounter (COLUMN%,DROP%
)
1010        X%=160*(COLUMN%-1)+96:Y%=1024
1020        drop%=DROP%*128+96

```

```

1030  GCOL 3,COL%(TURN%)
1040  MOVE X%,Y%:PRINT CIRCLE1$
1050  REPEAT
1060      Y%=Y%-4
1070      MOVE X%,Y%:PRINT CIRCLE2$
1080      UNTIL Y%=drop%
1090  ENDPROC
1100  REM*****
1110  DEF PROCwait(time)
1120      TIME=0
1130      REPEAT UNTIL TIME=time
1140  ENDPROC
1150  REM*****
1160  DEF FNdepth(COLUMN%)
1170      counter%=7
1180      REPEAT
1190          counter%=counter%-1
1200          UNTIL BOARD%(COLUMN%,counter%)<
>0 OR counter%=0
1210          =counter%+1
1220      REM*****
1230  DEF PROCcheckwin(X%,Y%,TURN%)
1240      YY%=Y%:XX%=X%
1250      REPEAT
1260          YY%=YY%-1
1270          UNTIL BOARD%(X%,YY%)<>TURN%
1280      count%=0
1290      REPEAT
1300          YY%=YY%+1:count%=count%+1
1310          UNTIL BOARD%(X%,YY%)<>TURN% OR
count%=5
1320      IF count%=5 THEN finished=TRUE:EN
DPROC
1330      REPEAT
1340          XX%=XX%-1
1350          UNTIL BOARD%(XX%,Y%)<>TURN%
1360      count%=0
1370      REPEAT
1380          XX%=XX%+1:count%=count%+1
1390          UNTIL BOARD%(XX%,Y%)<>TURN% OR
count%=5
1400      IF count%=5 THEN finished=TRUE:EN
DPROC
1410      XX%=X%:YY%=Y%
1420      REPEAT

```



```

1430      XX%=XX%-1:YY%=YY%-1
1440      UNTIL BOARD%(XX%,YY%)<>TURN%
1450      count%=0
1460      REPEAT
1470      XX%=XX%+1:YY%=YY%+1:count%=count%+1
1480      UNTIL BOARD%(XX%,YY%)<>TURN% OR count%=5
1490      IF count%=5 THEN finished=TRUE:ENDPROC
1500      XX%=X%:YY%=Y%
1510      REPEAT
1520      XX%=XX%+1:YY%=YY%-1
1530      UNTIL BOARD%(XX%,YY%)<>TURN%
1540      count%=0
1550      REPEAT
1560      XX%=XX%-1:YY%=YY%+1:count%=count%+1
1570      UNTIL BOARD%(XX%,YY%)<>TURN% OR count%=5
1580      IF count%=5 THEN finished=TRUE:ENDPROC
1590      ENDPROC
1600      REM*****
1610      DEF PROCwon(TURN%)
1620      VDU 4:CLS
1630      PRINTCHR$7;"WELL DONE ";NAME$(TURN%)
1640      ENDPROC
1650      REM*****
1660      DEF PROCdraw
1670      VDU 4:CLS
1680      PRINTTAB(5,31);CHR$7;"IT'S A DRAW";PROCwait(500)
1690      ENDPROC
1700      REM*****
1710      DEF FNanothergame
1720      CLS:PRINTTAB(5)"DO YOU WANT" 'TAB(4)"ANOTHER GAME?";
1730      REPEAT
1740      A$=GET$
1750      UNTIL A$="Y" OR A$="y" OR A$="N" OR A$="n"
1760      IF A$="Y" OR A$="y" THEN =TRUE ELSE =FALSE

```

```

1770 REM*****
1780 DEF PROCend
1790 VDU 4,26,16:PRINTTAB(6,16)"GOODBY
E"
1800 ENDPROC
1810 REM*****
1820 DEF PROCerror
1830 IF ERR<>17 THEN VDU 4:CLS:REPORT:
PRINT" at line ";ERL:END
1840 PROCend:END
1850 ENDPROC
1860 REM*****

```

# CITY SMASHER

This is a shoot and bomb game where you are flying an aircraft over an enemy city. The plan is to hit the skyscrapers with your bombs, but these are tall tower blocks so you are also in danger of bashing into them as your plane slowly loses height during the game.

You can move the plane down to get closer for better bombing aim and you have a laser gun mounted at the front of the plane for forward firing, in case you get too low and have to blast the top off a building to avoid it. Unfortunately this will only fire four times before it runs out of laser power. Your controls are:

SPACE = Drop bombs

Z = Move down

A = Fire laser

## Program Description

**40-230** Change mode and print instructions.

**240-260** Wait for the S key to be pressed.

**210** Change mode.

**280** Define the envelope.

**290** Turn off the cursor.

**300-360** Define the characters.

**310-380** Set up some variables.

**390** Print the city.

**400** The start of the game loop.

**410** Print the plane.

**420** Print the score.

**430-440** Move the plane.

**450** Make a short delay.

**460-490** Test the keys and act accordingly.

**500** If a bomb is in the air then move it.

**510** See if the plane has landed.

**540-550** If the plane has landed, increase the size of the buildings

and add one to the level number, then go back to the game loop.

**560-590** Print the score and wait for the S key to be pressed, then start the game.

**600-680** PROCBUILD draw the city

**690-750** PROCLAND make the landing noise, increase the score and make a short pause.

**760-850** PROCCRASH - crash the plane.

**860-920** PROBOMB - destroy part of a building.

**930-1010** PROCHIT - bomb hits building.

**1020-1090** PROCFIRE fire the laser

---

```
10 REM      CITY SMASHER
20 REM      (C)N.BELL 1984
30 *TV255
40 MODE7
50 PRINTCHR$141;CHR$131;"          C
ITY SMASHER"
60 PRINTCHR$141;CHR$130;"          C
ITY SMASHER"
90 PRINT:PRINTCHR$129;"You are the pi
lot of a bomber plane"'CHR$129"flying ov
er an enemy city. The idea of"'CHR$129"t
he game is to survive as long as"'CHR$12
9"possible without crashing."
100 PRINT
110 PRINTCHR$134;"To do this you must
bomb the buildings"
120 PRINTCHR$134;"which stand in your
way. If you totally";
130 PRINTCHR$134;"destroy the city, it
is replaced by a"
140 PRINTCHR$134;"bigger one! as well
as bombs, you are"
150 PRINTCHR$134;"equipped with a forw
ard firing laser,"
160 PRINTCHR$134;"but this can only be
used four times"
170 PRINTCHR$134;"in each game. There
is also a control"
180 PRINTCHR$134;"to move you down."
```

```

190 PRINT:PRINTCHR$129;TAB(7)"SPACE =
drop bomb"
200 PRINTCHR$129;TAB(7)"      A = fire l
aser"
210 PRINTCHR$129;TAB(7)"      Z = move d
own"
220 PRINT:PRINTCHR$141;CHR$130;"      PRE
SS KEY S TO START GAME"
230 PRINTCHR$141;CHR$131;"      PRESS KEY
S TO START GAME"
240 *FX15,1
250 IFINKEY$(0)="S"THENGOTO270
260 GOTO250
270 MODE2
280 ENVELOPE1,1,5,0,0,25,0,0,0,0,0,0,0
,0
290 VDU23;8202;0;0;0;0;0;0;0
300 VDU23,224,255,153,153,255,255,153,
153,255
310 VDU23,225,0,0,0,0,195,195,195,0
320 VDU23,226,255,255,255,255,255,255,
255,255
330 VDU23,227,128,192,224,255,243,115,
63,31
340 VDU23,228,0,0,0,240,60,63,255,254
350 VDU23,229,60,24,24,60,60,60,60,24
360 VDU23,230,72,2,164,1,72,36,0,136
370 SCORE=0:SIZE=15:MISL=4:LEV=1
380 A=0:B=0:FLAG=0:BOMBFLAG=0:CLS
390 PROCBUILD
400 REPEAT
410      COLOUR7:PRINTTAB(A,B)CHR$32;CHR$
227;CHR$228
420      PRINTTAB(1,30)"SCORE=";SCORE:PRI
NTTAB(17,30)"L";LEV
430      A=A+1
440      IFA=20THENA=0:B=B+1
450      FORM=1TO40:NEXTM
460      Z$=INKEY$(0)
470      IFZ$=" "ANDBOMBFLAG=0ANDB<28THEN
BOMBFLAG=1:X=A:Y=B
480      IFZ$="A"ANDB>0ANDMISL>0THENPROCF
IRE
490      IFZ$="Z"ANDB<27THENPRINTTAB(A,B)
"      B=B+1

```

```

500     IFBOMBFLAG=1 THEN PROC BOMB
510     IFB=28 AND A=18 THEN PROC LAND
520     IF POINT ( (A+2) * 64 + 3, 1024 - (B+1) * 32
+16) = 6 THEN PROC CRASH
530     UNTIL FLAG=1
540     IF A=18 THEN SIZE=SIZE-2: IF SIZE<1 THEN
SIZE=1
550     IF A=18 THEN LEV=LEV+1: GOTO 380
560     COLOUR5: PRINT TAB(1,10) "        SCORE="
; SCORE; TAB(0,15) "    PRESS KEY 'S'  FOR"; TA
B(0,17) "        ANOTHER GAME"
570     *FX15,1
580     IF INKEY$(0) = "S" THEN GOTO 270
590     GOTO 580
600     DEF PROC BUILD
610     FOR F=2 TO 19 STEP 2
620         FOR G=29 TO SIZE+RND(6) STEP -1
630             COLOUR6: PRINT TAB(F,G) CHR$224
640             NEXT G
650             COLOUR12: PRINT TAB(F,G) CHR$225
660             NEXT F
670     FOR H=1 TO 19: COLOUR2: PRINT TAB(H,29) C
HR$226: NEXT H
680     END PROC
690     DEF PROC LAND
700     COLOUR5: PRINT TAB(2,15) "YOU HAVE LA
NDED! "
710     SOUND1,1,40,44
720     FOR R=1 TO 3500: NEXT R
730     LET SCORE=SCORE+1000
740     FLAG=1
750     END PROC
760     DEF PROC CRASH
770     SOUND0,-15,110,55: PRINT TAB(A,B) CHR
$230: PRINT TAB(A+1,B) CHR$230
780     FOR Q=1 TO 10
790         VDU19,6,1,0,0,0: VDU19,12,1,0,0,0
: VDU19,7,1,0,0,0: VDU19,2,1,0,0,0
800         FOR R=1 TO 200: NEXT R
810         VDU19,6,7,0,0,0: VDU19,12,7,0,0,0
: VDU19,7,7,0,0,0: VDU19,2,7,0,0,0
820         FOR R=1 TO 200: NEXT R
830         NEXT Q
840     CLS: FLAG=1
850     END PROC

```

```

860 DEFPROC BOMB
870 COLOUR5:PRINTTAB(X,Y)CHR$32:PRINTTAB(X,Y+1)CHR$229
880 IF POINT(X*64+4,1024-(Y+3)*32+7)=6 THEN PROC HIT
890 IF Y=27 THEN PRINTTAB(X,Y+1)CHR$32:BO
MBFLAG=0
900 Y=Y+1
910 SOUND1,-10,(32-Y)*5,1
920 ENDPROC
930 DEFPROC HIT
940 COLOUR1
950 SOUND0,-15,100,5
960 PRINTTAB(X,Y+1)CHR$32:PRINTTAB(X,Y
+2)CHR$230:PRINTTAB(X,Y+3)CHR$230:PRINTTAB(X,Y+4)CHR$230
970 VDU19,1,3,0,0,0:FORS=1TO50:NEXTS:VDU19,1,1,0,0,0:FORS=1TO50:NEXTS
980 PRINTTAB(X,Y+2)CHR$32:PRINTTAB(X,Y+3)CHR$32:PRINTTAB(X,Y+4)CHR$32
990 COLOUR2:PRINTTAB(X,29)CHR$226
1000 BOMBFLAG=0:SCORE=SCORE+100
1010 ENDPROC
1020 DEFPROC FIRE
1030 SOUND2,1,100,10
1040 MOVE(A+2)*64,1024-(B*32)-25
1050 GCOL0,1
1060 DRAW1280,1024-(B*32)-25:FOR F=1TO100:NEXT F
1070 PRINTTAB(0,B-1);"
      "
1080 MISL=MISL-1
1090 ENDPROC

```

# BABY MONTY

Monty the mole's baby brother is trapped in a series of deep, dark underground caverns winch are slowly filling with water. Luckily he has found an old boat and is rowing like mad - he has to, to avoid the barrels which are mysteriously tumbling down on top of him.

The walls are also a source of danger, as they will sink the boat if it bumps into them. He will also be crushed to death if trapped try the rising tide against the ceiling of the caves.

However, you can help the miserable mole by steering the boat safely through the maze of caves. Strangely enough you can also use the flood to float the boat to the surface at the top of the screen faster. Yaw controls for this subterranean mission are:

Z = Left

X = Right

SHIFT = Raise water level

Can you save the mole from a watery grave and get him back to his mummy?

## Program Description

**20** Change mode.

**30** Turn off the cursor.

**40** Define the characters.

**50** The start of the program loop.

**60** Print the instructions.

**70** The start of the game loop.

**80** Draw the screen.

**90** The start of the life loop.

**100** If the SHIFT key is being pressed move baby Monty up.

**110** Move the barrels.

**120** Test the left and fight keys and act accordingly.

**130** Raise water level every three-fifths of a second.

**140** Move the barrels.



**150** Unless a life has been lost or a level completed, return to the life loop.

**160** Go to the life lost or level completed procedure.

**170** If the player still has some lives left then return to the game loop.

**180** Return to the program loop.

**210-570** PROCsetup - define the characters, envelope and ban arrays. Set the high score to zero.

**590-850** PROCinstructions - comprising:

**600-780** Print the instructions.

**790-830** Get the number of the starting level.

**840** Set the score to zero and the number of moles to 3.

**810-1060** PROCscreen comprising:

**880** Clear the screen.

**890-920** Print a red border around the game area.

**930-940** Print holes at the top and bottom.

**950-980** Print the barriers with gaps at random positions.

**990** Set the variables for the mole's initial position.

**1000-1030** Set the variables for the number of barrels, their speed and position.

**1040** Print the score and number of lines.

**1050** Print the mole and set some variables.

**1080-1190** PROCscore print the score, high score level and number of lives.

**1210-1250** PROCmoie - print the mole.

**1210-1310** PROCup - comprising:

**1290** Remove the mole from the screen.

**1300** Move the mole up one pixel,

**1310** Print the mole on the screen.

**1320** If any barrels are level with the mole remove them.

**1330-1340** Increase the water level.

**1350** Make the moving noise.

**1360** Replace any barrels previously removed.

**1390-1430** PROCieft - move the mole to the left.

**1450-1490** PROCright - move the mole to the right.

**1510-1590** PROCbarrel - move one of the barrels after a set firm has elapsed since the last move.

**1610-1700** FNhit - check wound the mole to see if he has hit anything.

**1720-1780** PROCdead - make the death noise and reduce the number of moles lives.

**1800-1850** PROCnext-level - make the next level noise, increase the score and add one to the level number.

---

```
10  ON ERROR MODE 1:IF ERR<>17 REPORT
:PRINT" at line ";ERL:END ELSE END
20  MODE 1
30  VDU 23;10,32;0;0;0
40  PROCsetup
50  REPEAT
60    PROCinstructions
70    REPEAT
80      PROCscreen
90      REPEAT
100        IF INKEY-1 PROCup
110        PROCbarrel
120        IF INKEY-98 PROCleft ELSE I
F INKEY-67 PROCright
130        IF TIME>UTIME%+120 PROCup:U
TIME%=TIME
140        PROCbarrel
150        UNTIL FNhit(MX%,MY%) OR MY%
>=920
160        IF MY%>=920 PROCnext_level EL
SE PROCdead
170        UNTIL MO%=0
180        UNTIL FALSE
190      END
200    REM =====
210    DEF PROCsetup
220    VDU 19,0,4;0;19,1,5;0;19,2,0;0;19
,3,6;0;
230    VDU 23,224,243,0,63,0,243,0,63,0
240    VDU 23,225,0,3,15,15,15,0,0,0
250    VDU 23,226,0,240,252,0,0,0,0,0
260    VDU 23,227,0,0,255,63,63,0,0,0
270    VDU 23,228,12,0,0,207,255,0,0,0
280    VDU 23,229,0,0,0,255,255,0,0,0
290    VDU 23,230,3,3,0,252,207,48,0,0
```

```

300  VDU 23,231,0,255,0,0,0,0,0,0
310  VDU 23,232,0,243,207,0,0,0,0,0
320  VDU 23,233,0,255,255,0,0,0,0,0
330  VDU 23,234,0,0,0,0,48,0,252,255
340  VDU 23,235,243,12,48,48,0,0,0,0
350  VDU 23,236,0,15,63,60,60,0,0,0
360  VDU 23,237,0,192,240,0,0,0,0,0
370  VDU 23,238,0,0,255,0,63,0,0,0
380  VDU 23,239,48,0,243,0,255,0,0,0
390  VDU 23,240,0,0,255,0,255,0,0,0
400  VDU 23,241,0,0,0,3,3,0,0,0
410  VDU 23,242,12,12,0,240,60,192,0,0
420  VDU 23,243,0,255,0,63,0,0,0,0
430  VDU 23,244,0,243,0,243,0,0,0,0
440  VDU 23,245,0,255,0,255,0,0,0,0
450  VDU 23,246,0,0,0,0,0,0,3,3
460  VDU 23,247,0,0,0,0,192,0,240,255
470  VDU 23,248,3,0,0,0,0,0,0,0
480  VDU 23,249,204,12,12,12,0,0,0,0
490  VDU 23,250,126,255,255,255,255,25
5,126,0
500  VDU 23,251,0,24,24,24,24,24,0,0
510  VDU 23,252,0,6,12,24,48,96,0,0
520  VDU 23,253,0,0,0,126,0,0,0,0
530  VDU 23,254,0,96,48,24,12,6,0,0
540  ENVELOPE 1,1,-26,-36,-45,255,255,
255,0,0,0,0,0,0
550  DIM BC%(7),BX%(7),BY%(7)
560  HS%=0
570  ENDPROC
580  REM =====
590  DEF PROCinstructions
600  VDU 4:CLS
610  PRINT'"BABY MONTY, MONTY MOLE'S L
ITLLE BROTHER"
620  PRINT'TAB(11)"BY ANTONY CROWTHER"
630  PRINT'TAB(7)"BBC VERSION BY JAMES
BYRNE"
640  PRINT'"I'll start by putting you
in the"
650  PRINT"picture, you are taking par
t in a new"
660  PRINT"adventure with baby mole."
670  PRINT'"Baby mole has fallen from
his shelf"

```

```

680 PRINT"(Gremlin graphics software
house), into"
690 PRINT"a mega-dark cavern. His onl
y means of"
700 PRINT"escape is to jump into an o
ld rowing"
710 PRINT"boat. By staying afloat on
the vessel"
720 PRINT"he can guide himself throug
h the caves"
730 PRINT"above."
740 PRINT'"But beware of the tumbling
barrels"
750 PRINT"which have escaped from an
old arcade"
760 PRINT"game left on the side to ro
t."
770 PRINT'"Keys are:"
780 PRINTTAB(10);"Z - left"TAB(10);"
X - right"TAB(6);"SHIFT - raise water l
evel"
790 REPEAT
800     *FX 15,1
810     PRINTTAB(0,28)SPC(40)
820     INPUTTAB(0,28)"INPUT LEVEL (1-1
5)" ,L%
830     UNTIL L%>0 AND L%<16
840     CLS:SC%=0:MO%=3
850     ENDPROC
860     REM =====
870     DEF PROCscreen
880     GCOL0,0:VDU24,31;128;956;927;:CLG
:COLOUR129:VDU4
890     PRINTTAB(0,4)SPC(31);TAB(0,28)SPC
(31)
900     FOR Y%=5 TO 27
910         PRINTTAB(0,Y%) " ";TAB(30,Y%) " "
920     NEXT
930     COLOUR 131:PRINTTAB(13,28) "      "
940     COLOUR 128:PRINTTAB(13,4) "      "
950     FOR Y%=8 TO 20 STEP 4
960         COLOUR1:COLOUR130:PRINTTAB(1,Y%
)STRING$(29,CHR$224)
970         COLOUR128:PRINTTAB(RND(22)+1,Y%
)SPC(6)

```

```

980      NEXT
990      MX%=440:MY%=176:MC%=0
1000     FORJ%=1TO3:BC%(J%)=0:NEXT
1010     NB%=INT((L%+1)/2):IF L%/2=INT(L%/
2)BS%=16 ELSE BS%=8
1020     IF L%=15 NB%=7:BS%=16
1030     FORJ%=1TONB%:BY%(J%)=1400-L%*25+J
%*48:BX%(J%)=RND(924)+31:NEXT
1040     PROCscore:VDU5
1050     PROCmole:TIME=0:J%=0:BTIME%=0:UTI
ME%=0
1060     ENDPROC
1070     REM =====
1080     DEF PROCscore
1090     COLOUR3
1100     PRINTTAB(32,5)"SCORE"
1110     PRINTTAB(34,6);SC%
1120     IF SC%>HS% HS%=SC%
1130     PRINTTAB(31,8)"HI-SCORE"
1140     PRINTTAB(34,9);HS%
1150     PRINTTAB(32,11)"LEVEL"
1160     PRINTTAB(34,12);L%
1170     PRINTTAB(32,14)"MOLES"
1180     PRINTTAB(34,15);MO%
1190     ENDPROC
1200     REM =====
1210     DEF PROCmole
1220     MOVE MX%,MY%
1230     IF MC%=0 VDU 18,3,1,225,226,10,8,
8,227,228,229,18,3,3,11,8,8,230,10,8,8,2
31,232,233,18,3,2,11,8,8,234,10,8,235:EN
DPROC
1240     VDU 18,3,1,236,237,10,8,8,238,239
,240,18,3,3,11,8,8,8,241,242,10,8,8,243,
244,245,18,3,2,11,8,8,8,246,247,10,8,8,2
48,249
1250     ENDPROC
1260     REM =====
1270     DEF PROCup
1280     PROCbarrel
1290     PROCmole
1300     MY%=MY%+4:MC%=1-MC%
1310     PROCmole
1320     FORZ%=1TONB%:IF BY%(Z%)<MY%-12 AN
D BY%(Z%)>MY%-56 MOVE BX%(Z%),BY%(Z%):VD

```

```

U18,3,3,250,8,18,3,1,250+BC%(Z%):NEXT EL
SE NEXT
1330   GCOL 0,3
1340   PLOT 77,MX%,MY%-52
1350   SOUND 0,-10,5,1
1360   FORZ%=1TONB%:IF BY%(Z%)<MY%-12 AN
D BY%(Z%)>MY%-56 MOVE BX%(Z%),BY%(Z%):VD
U18,3,3,250,8,18,3,1,250+BC%(Z%):NEXT EL
SE NEXT
1370   ENDPROC
1380   REM =====
1390   DEF PROCleft
1400   PROCmole
1410   MX%=MX%-16:MC%=1-MC%
1420   PROCmole
1430   ENDPROC
1440   REM =====
1450   DEF PROCright
1460   PROCmole
1470   MX%=MX%+16:MC%=1-MC%
1480   PROCmole
1490   ENDPROC
1500   REM =====
1510   DEF PROCbarrel
1520   IF TIME<BTIME%+15-L% ENDPROC
1530   J%=J% MOD NB%+1
1540   MOVE BX%(J%),BY%(J%):VDU18,3,3,25
0,8,18,3,1,250+BC%(J%)
1550   BC%(J%)=BC%(J%) MOD 4+1
1560   BY%(J%)=BY%(J%)-BS%
1570   MOVE BX%(J%),BY%(J%):VDU18,3,3,25
0,8,18,3,1,250+BC%(J%)
1580   IF BY%(J%)<96 BY%(J%)=1400-L%*25+
J%*48:BX%(J%)=RND(924)+31
1590   BTIME%=TIME:ENDPROC
1600   REM =====
1610   DEF FNhit(X%,Y%)
1620   H%=FALSE
1630   IF POINT(X%,Y%)<>0 =TRUE
1640   IF POINT(X%,Y%-8)<>0 =TRUE
1650   IF POINT(X%,Y%-32)<>0 =TRUE
1660   IF POINT(X%,Y%-48)<>0 =TRUE
1670   IF POINT(X%+64,Y%+4)<>0 =TRUE
1680   IF POINT(X%+96,Y%-32)<>0 =TRUE
1690   IF POINT(X%+96,Y%-48)<>0 =TRUE

```

```

1700  =FALSE
1710  REM =====
1720  DEF PROCdead
1730  MO%=MO%-1
1740  SOUND 0,-15,4,50
1750  TIME=0
1760  REPEAT UNTIL TIME>300
1780  ENDPROC
1790  REM =====
1800  DEF PROCnext_level
1810  SOUND 1,1,255,156
1820  TIME=0:REPEATUNTILTIME>800
1830  L%=L%+1:IF L%>15 L%=15
1840  SC%=SC%+L%*100
1850  ENDPROC

```

# FLAPPA CATCHA

By Alan Rowley

You are Sam. Together with your girlfriend, Ethel, you have been trying to steal the ill-gotten gold of an evil old witch, mad Mabel. You've both been nabbed by Mabel and Ethel is held in a cage ready for supper - Mabel's!

To provide sport to help her appetite, Mabel has trapped you between two walls and is launching exploding demon-birds at you. You do, however, have a hat with magical properties which will neutralize the birds if you can catch them in it before they land.

For every six birds caught Mabel will give you a bag of gold. You need three bags to win the game and save Ethel.

Every time you miss a bird, Mabel's pet zombie will gradually tunnel under a wall and eventually get you! If you are hit by a bird you will be instantly vaporized. You have only three lives! Be careful, because if Mabel thinks you're doing too well you'll have to dodge a few lightning bolts as well!

Your controls for moving about to catch the birds are:

Z = Left

X = Right

## Program Description

**10-160** Display Instructions.

**170** Change mode

**180-500** Install user-defined graphics

**510-540** Define sound envelopes.

**560** Define logical colour.

**570-590** Set initial variable values.

**600-630** Draw game screen

**640** Write text at graphics cursor.

**650** Set delay before auto-repeat to one hundredth of a second.

**660** Set auto repeat period to one hundredth of a second.



**670** Print Sam, Ethel, Mabel, the zombie and the gold on the screen.

**680-710** Main loop for game.

**730** Routine to deal with errors.

**740-770** End of game routine.

**780-790** Reset computer for normal use.

**810-860** PROC-keys - acts as a time-delay by either moving Sam if the appropriate key is pressed or by calling PROC-wait.

**870-900** PROC-ground - draw a green base on a background previously redefined as cyan by the use of VDU19.

**910-940** PROC-wall - draws a wall at the position and colour given by the parameters.

**950-910** PROC-sam prints Sam at this initial position.

**980-1000** PROC-ethel - prints Ethel.

**1010-1030** PROC-witch prints Mabel.

**1040-1060** PROC-zomb prints zombie in initial position; animation by flashing colours.

**1010-1160** PROC-samright moves Sam right and animate legs.

**1110-1260** PROC-sam1left - moves Sam left and animates legs.

**1270-1490** PROC-birds animates birds along a random parabola.

**1500-1540** PROC-ch checks the number of birds caught.

**1550-1688** PROC-mz moves zombie down.

**1610-1610** PROC-mz2 moves zombie fight.

**1680-1740** PROC-mz3 - moves zombie up.

**1750-1840** PROC-dead - mb out Sam if he is killed and check if he has any fives left.

**1850-1810** Proc-wait - time-delay according to the value given.

**1880-1930** PROC-f1ash produces a colour flash when Sam catches a bird.

**1940-2030** PROC-exp - prints an explosion at the co-ordinates and colour given.

**2040-2110** PROC-cage draws a cage in the colour given.

**2120-2000** PROC-plinth - draws a platform for Mabel to stand on.

**2160-2200** PROC-gold - prints the bags of gold in their original position.

**2210-2240** PROC-rubwitch - rub out Mabel.

**2250-2360** PROC-takegold - checks to see which bag of gold must be taken next and reprints it in its new position.

**2310-2500** PROC-zap - levitates Mabel according to Sam's position ready for lightning bolt.

**2510-2580** PROC-bolt prints a lightning bolt and explosion.

**2590-2630** PROC-help prints Ethel's cry for help  
**2640-2610** PROC-heart prints a heart shape.

**2680-2120** PROC-win - terminates game with congratulatory message.

---

```
10 MODE7:PRINTTAB(10,1);CHR$130;CHR$1
41;"FLAPPA CATCHA"
20 PRINTTAB(10,2);CHR$130;CHR$141;"FL
APPA CATCHA"
30 PRINT
40 PRINT'"You are Sam. Together with
your girl-"'"friend, Ethel, you have bee
n trying to""steal the ill-gotten gold
of an evil""old witch, mad Mabel. You'v
e both been""nabbed by Mabel and Ethel
is held in a""cage ready for supper."
50 PRINT"To provide sport to help her
appetite,"'"Mabel has trapped you betwe
en two walls""and is launching explodin
g demon-birds"
60 PRINT"at you. You do have, however
, a hat""with magical properties which
will""neutralise the birds if you can c
atch""them in it before they land."
70 PRINT"For every 6 birds caught Mab
el will""give you a bag of gold. You ne
ed 3""bags to win the game and save Eth
el."
80 PRINTTAB(1,23);CHR$129;CHR$157;CHR
$130;" Press <SPACE> to continue...";CHR
$135;CHR$156;
90 REPEATUNTILGET=32
100 CLS:PRINT'
110 PRINT"Every time you miss a bird,
Mabel's pet""zombie will gradually tunn
el under""a wall and eventually get you
! If you""are hit by a bird you will be
instantly""vaporised. You only have 3
lives!"
```

```
120 PRINT"Be careful, because if Mabel  
thinks""you're doing too well you'll h  
ave to""dodge a few lightning bolts as  
well..!"
```

```
130 PRINTTAB(5,14);CHR$129;"Use";CHR$1  
31;"Z";CHR$129;"to move";CHR$131;"LEFT"
```

```
140 PRINTTAB(9,16);CHR$131;"X";CHR$129  
;"to move";CHR$131;"RIGHT"
```

```
150 PRINTTAB(2,20);CHR$129;CHR$157;CHR  
$131;" Press <SPACE> to play...";CHR$135  
;CHR$156;
```

```
160 REPEATUNTILGET=32
```

```
170 MODE2
```

```
180 VDU23;10,32,0;0;0;
```

```
190 VDU23,224,56,68,198,0,0,0,0,0
```

```
200 VDU23,225,56,124,254,254,0,0,0,0
```

```
210 VDU23,226,0,0,0,240,240,96,0,0
```

```
220 VDU23,227,174,232,174,8,14,0,0,0
```

```
230 VDU23,228,24,24,255,24,24,60,36,0
```

```
240 VDU23,229,3,1,0,128,192,0,192,7
```

```
250 VDU23,230,192,128,0,1,3,0,3,224
```

```
260 VDU23,231,0,0,0,0,40,232,136,12
```

```
270 VDU23,232,0,0,0,0,44,36,38,48
```

```
280 VDU23,233,0,0,0,0,40,46,34,96
```

```
290 VDU23,234,3,30,12,4,0,220,254,126
```

```
300 VDU23,235,30,28,28,62,127,0,0,60
```

```
310 VDU23,236,0,0,16,24,8,0,0,0
```

```
320 VDU23,237,0,0,0,0,0,20,20,0
```

```
330 VDU23,238,60,126,255,255,255,1  
26,60
```

```
340 VDU23,239,153,149,149,153,145,209,  
16,1
```

```
350 VDU23,240,56,68,0,0,0,0,0,0
```

```
360 VDU23,241,0,56,56,16,0,68,131,128
```

```
370 VDU23,242,0,0,0,0,124,56,56,56
```

```
380 VDU23,243,56,56,108,108,0,0,0,0
```

```
390 VDU23,244,64,0,0,0,40,40,40,108
```

```
400 VDU23,245,0,0,0,0,104,72,200,24
```

```
410 VDU23,246,255,255,255,255,255,255,  
255,255
```

```
420 VDU23,247,0,20,62,28,62,20,0,0
```

```
430 VDU23,248,20,0,85,0,85,0,20,0
```

```
440 VDU23,249,0,0,34,0,0,0,0,0
```

```
450 VDU23,250,0,0,0,34,0,0,0,0
```

```
460 VDU23,251,0,0,0,20,8,20,0,0
```

```

470 VDU23,252,0,0,28,16,20,28,0,0
480 VDU23,253,28,62,99,111,107,99,62,2
8
490 VDU23,254,0,0,0,0,0,34,54,28
500 VDU23,255,108,254,254,254,124,56,1
6,0
510 ENVELOPE1,1,20,-15,-15,90,90,90,12
7,-100,-1,-7,126,126
520 ENVELOPE2,1,0,0,0,0,0,0,127,-1,-2,
-2,126,126
530 ENVELOPE3,1,0,0,0,6,3,3,127,20,-15
,-2,80,50
540 ENVELOPE4,3,-10,0,-10,6,3,3,0,-127
,0,-120,126,126
550 ON ERROR GOTO730
560 VDU19,0,6;0;19,5,0;0;
570 X%=600:Y%=365:X2%=1000:Y2%=416:W%=
100:Z%=335:X3%=1160:Y3%=365
580 GX%=848:GY%=365:Gold%=0:Q%=0
590 SC%=0:G%=4:Miss%=0:Dead%=FALSE:Liv
es%=3
600 PROC_ground:PROC_wall(800,5):PROC_
wall(200,5)
610 PROC_cage(5):PROC_plinth(5)
620 PRINTTAB(0,1)"GOLD";TAB(13,1)"LIVE
S"
630 PRINTTAB(19,1);Lives%
640 VDU5
650 *FX11,1
660 *FX12,1
670 PROC_sam:PROC_ethel:PROC_witch:PRO
C_zomb:PROC_gold
680 REPEAT
690 IF SC%>2 AND RND(5)>3 PROC_zap
700 PROC_help:PROC_birds
710 UNTIL Dead%=TRUE
720 IF Dead%=TRUE GOTO740
730 VDU4:MODE7:PRINT':REPORT:PRINT" a
t line ";ERL:GOTO780
740 VDU19,5,1;0;:SOUND1,4,80,25:SOUND1
,4,150,5:VDU4:COLOUR1:PRINTTAB(3,8)"PRES
S SPACE BAR"" TO PLAY AGAIN"
750 PRINTTAB(5,12)"OR E TO END"
760 ON INSTR(" E",GET$) GOTO170,770 EL
SE 760

```

```

770 MODE7:PRINTTAB(14,16);CHR$131;"THE
END"
780 *FX12,0
790 *FX15,1
800 END
810 DEFPROC_keys
820 *FX15,1
830 IF INKEY(-67) PROC_samright:ENDPROC
C
840 IF INKEY(-98) PROC_samleft:ENDPROC
850 PROC_wait(40/G%)
860 ENDPROC
870 DEFPROC_ground
880 GCOL0,2:MOVE0,0:MOVE0,300:PLOT85,1
300,300
890 MOVE1300,0:PLOT85,0,0
900 ENDPROC
910 DEFPROC_wall(XW%,CL%)
920 GCOL0,CL%:MOVEXW%,304:MOVEXW%,500:
PLOT85,XW%+50,500
930 MOVEXW%+50,304:PLOT85,XW%,304
940 ENDPROC
950 DEFPROC_sam
960 MOVEX%,Y%:VDU18,0,5,240,18,0,7,8,2
41,18,0,1,8,242,18,0,4,8,10,243,18,0,7,8
,244,18,0,5,11,226
970 ENDPROC
980 DEFPROC_ethel
990 MOVEX3%,Y3%:VDU18,0,3,224,18,0,7,8
,241,18,0,1,8,242,18,0,3,8,10,225,18,0,7
,8,244
1000 ENDPROC
1010 DEFPROC_witch
1020 MOVEX2%,Y2%:VDU18,0,4,234,18,0,7,8
,236,18,0,4,8,10,235,18,0,7,8,237
1030 ENDPROC
1040 DEFPROC_zomb
1050 MOVEW%,Z%:VDU18,0,1,228,8,18,0,9,2
29,8,18,0,14,230
1060 ENDPROC
1070 DEFPROC_samright
1080 IF POINT(X%+108,Y%)=5 ENDPROC
1090 VDU18,0,6,25,4,X%;Y%;246,8,10,246,
11,226
1100 X%=X%+8

```

```

1110 MOVEX%,Y%:VDU18,0,5,240,18,0,7,8,2
41,18,0,1,8,242,18,0,4,8,10,243,18,0,7,8
,231,18,0,5,11,226
1120 VDU18,0,6,25,4,X%;Y%;246,8,10,246,
11,226
1130 X%=X%+8
1140 MOVEX%,Y%:VDU18,0,5,240,18,0,7,8,2
41,18,0,1,8,242,18,0,4,8,10,243,18,0,7,8
,232,18,0,5,11,226
1150 IF X%>W%-64 AND X%<W%+64 AND Z%>29
0 PROC_dead
1160 ENDPROC
1170 DEFPROC_samleft
1180 IF X%<280 ENDPROC
1190 VDU18,0,6,25,4,X%;Y%;246,8,10,246,
11,226
1200 X%=X%-8
1210 MOVEX%,Y%:VDU18,0,5,240,18,0,7,8,2
41,18,0,1,8,242,18,0,4,8,10,243,18,0,7,8
,233,18,0,5,11,226
1220 VDU18,0,6,25,4,X%;Y%;246,8,10,246,
11,226
1230 X%=X%-8
1240 MOVEX%,Y%:VDU18,0,5,240,18,0,7,8,2
41,18,0,1,8,242,18,0,4,8,10,243,18,0,7,8
,245,18,0,5,11,226
1250 IF X%<W%+64 AND Z%>290 PROC_dead
1260 ENDPROC
1270 DEFPROC_birds
1280 B%=RND(100)+200:C%=1000-B%
1290 D%=82
1300 j%=80
1310 SOUND1,3,80,2:SOUND1,3,120,2:SOUND
1,3,100,2
1320 REPEAT
1330 IF Dead%=TRUE ENDPROC
1340 D%=D%-G%
1350 U%=C%+B%*SIN(RAD(D%))-RND(G%*12)
1360 L%=365+300*COS(RAD(D%))+RND(G%*5
)
1370 MOVEU%,L%:VDU18,0,9,249,8,18,0,1
4,250,8,18,0,5,251
1380 SOUND1,-5,j%,1:j%=j%-50
1390 PROC_keys
1400 VDU18,0,6,25,4,U%;L%;249,8,250,8

```

```

,251
1410 UNTIL D%=-98 OR U%<X%+72 AND U%>
X%+20 AND L%<375 AND L%>360
1420 IF U%<X%+72 AND U%>X%+20 AND L%<37
5 AND L%>360 PROC_flash:SC%=SC%+1:PROC_c
h:ENDPROC
1430 PROC_exp(U%,L%,3)
1440 IF U%<X%+40 AND U%>X%-40 AND L%<36
0 THEN PROC_dead
1450 Miss%=Miss%+1
1460 IF Miss%<3 PROC_mz:ENDPROC
1470 IF Miss%>2 AND Miss%<7 PROC_mz2:EN
DPROC
1480 IF Miss%>6 AND Miss%<9 PROC_mz3:EN
DPROC
1490 IF Miss%>8 PROC_mz2:ENDPROC
1500 DEFPROC_ch
1510 IF SC%=6 PROC_takegold:G%=5:ENDPRO
C
1520 IF SC%=12 PROC_takegold:G%=6:ENDPR
OC
1530 IF SC%=18 PROC_takegold:PROC_cage(
6):PROC_ethel:PROC_wall(800,6):PROC_plin
th(6):PROC_rubwitch:PROC_heart:PROC_win:
ENDPROC
1540 ENDPROC
1550 DEFPROC_mz
1560 SOUND1,1,80,5
1570 VDU18,0,6,25,4,W%;Z%;246
1580 Z%=Z%-32
1590 MOVEW%,Z%:VDU18,0,6,246,8,18,0,1,2
28,8,18,0,9,229,8,18,0,14,230
1600 ENDPROC
1610 DEFPROC_mz2
1620 SOUND1,1,80,5
1630 VDU18,0,6,25,4,W%;Z%;246
1640 W%=W%+48
1650 MOVEW%,Z%:VDU18,0,6,246,8,18,0,1,2
28,8,18,0,9,229,8,18,0,14,230
1660 IF W%>X%-72 AND Z%>300 PROC_dead:P
ROC_zomb
1670 ENDPROC
1680 DEFPROC_mz3
1690 SOUND1,1,80,5
1700 VDU18,0,6,25,4,W%;Z%;246

```

```

1710 Z%=Z%+32
1720 MOVEW%,Z%:VDU18,0,6,246,8,18,0,1,2
28,8,18,0,9,229,8,18,0,14,230
1730 IF W%<X%+64 AND W%>X%-64 AND Z%>Y%
-64 PROC_dead:PROC_zomb
1740 ENDPROC
1750 DEFPROC_dead
1760 VDU18,0,6,25,4,X%;Y%;246,8,10,246,
11,226
1770 VDU19,0,1;0;19,6,1;0;:PROC_wait(20
):VDU19,0,6;0;19,6,6;0;
1780 FOR P%=100 TO 0 STEP-10:SOUND1,-10
,P%,3:NEXT
1790 Lives%=Lives%-1
1800 IF Lives%>0 X%=600:Y%=365:PROC_wai
t(50):PROC_sam
1810 IF Lives%<1 Dead%=TRUE
1820 IF Lives%<0 Lives%=0
1830 VDU4:PRINTTAB(19,1);Lives%:VDU5
1840 ENDPROC
1850 DEFPROC_wait(t%)
1860 TIME=0:REPEATUNTILTIME=t%
1870 ENDPROC
1880 DEFPROC_flash
1890 SOUND1,-15,200,1
1900 VDU19,5,3;0;
1910 PROC_wait(10)
1920 VDU19,5,0;0;
1930 ENDPROC
1940 DEFPROC_exp(x%,y%,cl%)
1950 SOUND0,-15,100,2
1960 VDU18,0,cl%,25,4,x%;y%;247
1970 PROC_wait(5)
1980 VDU18,0,6,25,4,x%;y%;247
1990 PROC_wait(5)
2000 VDU18,0,cl%,25,4,x%;y%;248
2010 PROC_wait(5)
2020 VDU18,0,6,25,4,x%;y%;248
2030 ENDPROC
2040 DEFPROC_cage(ca%)
2050 GCOL0,ca%
2060 FOR I%=305 TO 405 STEP 20
2070     MOVE1140,I%:DRAW1240,I%
2080     FOR J%=1140 TO 1240 STEP 25
2090         MOVEJ%,305:DRAWJ%,405

```



```

2100      NEXT,
2110      ENDPROC
2120      DEFPROC_plinth(cp%)
2130      GCOL0,cp%
2140      MOVE1000,304:MOVE1060,304:PLOT85
,1060,352:MOVE1000,352:PLOT85,1000,304
2150      ENDPROC
2160      DEFPROC_gold
2170      MOVEGX%,GY%:VDU18,0,3,254,18,0,3
,8,10,253,18,0,1,8,252
2180      MOVEGX%+48,GY%:VDU18,0,3,254,18,
0,3,8,10,253,18,0,1,8,252
2190      MOVEGX%+96,GY%:VDU18,0,3,254,18,
0,3,8,10,253,18,0,1,8,252
2200      ENDPROC
2210      DEFPROC_rubwitch
2220      VDU18,0,6,25,4,X2%;Y2%;246,8,10,
246
2230      PROC_exp(X2%,Y2%,7)
2240      ENDPROC
2250      DEFPROC_takegold
2260      IF Gold%=0 VDU18,0,6,25,4,GX%;GY
%;254,8,10,253,8,252
2270      IF Gold%=1 VDU18,0,6,25,4,GX%+48
;GY%;254,8,10,253,8,252
2280      IF Gold%=2 VDU18,0,6,25,4,GX%+96
;GY%;254,8,10,253,8,252
2290      GX%=300:GY%=1020
2300      IF Gold%=0 MOVEGX%,GY%:VDU18,0,3
,254,18,0,3,8,10,253,18,0,1,8,252
2310      IF Gold%=1 MOVEGX%+64,GY%:VDU18,
0,3,254,18,0,3,8,10,253,18,0,1,8,252
2320      IF Gold%=2 MOVEGX%+128,GY%:VDU18
,0,3,254,18,0,3,8,10,253,18,0,1,8,252
2330      FOR S%=200 TO 232 STEP8:SOUND1,4
,S%,1:NEXT
2340      GX%=848:GY%=365
2350      Gold%=Gold%+1
2360      ENDPROC
2370      DEFPROC_zap
2380      REPEAT
2390      PROC_keys
2400      VDU18,0,6,25,4,X2%;Y2%;246,8,1
0,246
2410      Y2%=Y2%+16:X2%=X2%-Q%

```

```

2420      PROC_witch
2430      Q%=Q%+1:SOUND1,3,Y2%-400,1
2440      UNTIL X2%<X%+RND(300) OR Y2%>8
00
2450      PROC_bolt
2460      PROC_exp(BE%,340,7)
2470      VDU18,0,6,25,4,X2%;Y2%;246,8,10,
246
2480      IF BE%<X%+48 AND BE%>X%-48 PROC_
dead
2490      PROC_zomb:PROC_wall(200,5):PROC_
wall(800,5):X2%=1000:Y2%=416:Q%=0:PROC_w
itch
2500      ENDPROC
2510      DEFPROC_bolt
2520      BE%=X%-150+RND(300)
2530      VDU18,0,7
2540      PLOT4,BE%,340:PLOT5,X2%-16,Y2%-1
6
2550      PROC_wait(10)
2560      VDU18,0,6
2570      PLOT4,BE%,340:PLOT5,X2%-16,Y2%-1
6
2580      ENDPROC
2590      DEFPROC_help
2600      VDU18,0,1,25,4,1140;432;227,239
2610      PROC_wait(60)
2620      VDU18,0,6,25,4,1140;432;227,239
2630      ENDPROC
2640      DEFPROC_heart
2650      VDU18,0,9,25,4,1190;400;255
2670      ENDPROC
2680      DEFPROC_win
2690      VDU4:COLOUR9:PRINTTAB(6,10)"WELL
DONE!"
2700      PROC_wait(500)
2710      Dead%=TRUE
2720      ENDPROC

```



# LOADING PROGRAMS FROM TAPE

If you have the tape version of the programs in this book, here are the instructions you need to get the games loaded and running.

- 1 Connect the cassette lead from the BBC to the sockets on your cassette player. If your BBC has the disk operating system type \*TAPE and see the note below.
- 2 Place the cassette into your cassette player and rewind to the beginning if necessary.
- 3 Adjust the volume on your cassette player to around 7 or 8 on a scale of 1 to 10, or as advised in your BBC manual.
- 4 Type `LOAD""` on your BBC.
- 5 Press `RETURN` on your BBC. The word "Searching" should appear.
- 6 Press `PLAY` on your cassette player.

After a few seconds (certainly less than a minute) the word "Loading" should appear followed by the name of a program. If this is the name of the program you wish to load, then:

- 7 Wait until your micro emits a beep, and then type `RUN` and press `RETURN`.  
However, if this was not the program you wished to load, then:
- 8 Press `ESCAPE` and repeat steps 4 to 7, as above. After waiting a while you will see name of the next program appear on the screen.

If the word "Loading" is not above the program name, if a question mark appears on the screen, or if the words "Rewind tape" come up, the program has failed to load. Rewind the tape, adjust the volume, check the leads, and start again.

## NOTES

The programs on this tape have a pause of a few seconds between them. A good short cut when looking for a specific program is simply to unplug the recorder from the computer and

listen to the cassette playing. Search for the short breaks; the noise immediately after the break is the command that makes the computer produce the program name.

Alternatively, type \*CAT in step 4, and the names of all the programs on the tape will appear, in the right order. If you have a counter on your tape recorder you can then note at what number each program starts, and catalogue your cassette.

## **Disk Users**

Some of the programs are too long to work automatically on the BBC microcomputers fitted with disk systems. To avoid mysterious problems, it is best to type PAGE=&E00 before attempting to load a program.

These programs are only tested for a BBC model B microcomputer with operating system 1.0, or later, and with the BASIC II interpreter.

# CENTURY SOFTWARE VOUCHER

All the programs in this book are available on cassette from the publishers.

Please cut out this voucher (do not copy the voucher) and send it together with a cheque/postal order for £4.95 made payable to Tiptree Book Services Ltd. to:

Tiptree Book Services Ltd.  
Tiptree  
Colchester  
Essex CO5 0SR

## VOUCHER



Please send me one cassette of the programs in PCW GAMES COLLECTIONS FOR THE BBC MICRO, at the special price of £4.95 including postage and packing.

NAME .....  
ADDRESS .....  
.....  
.....

*Please allow 28 days for delivery.*

## Also available from Century Communications:

VERY BASIC BASIC: COMMODORE 64

VERY BASIC BASIC: ELECTRON

VERY BASIC BASIC: SPECTRUM

Derek Ellershaw and Peter Schofield

MICROGUIDE FOR THE BBC MICRO

MICROGUIDE FOR THE COMMODORE 64

MICROGUIDE FOR THE DRAGON

MICROGUIDE FOR THE SPECTRUM

Peter Morse (Ed)

DISCOVER YOUR COMMODORE 64

DISCOVER YOUR ELECTRON

Philip Williams

DISCOVER YOUR SPECTRUM

Mark Ramshaw

THE MICRO ENQUIRER: THE SPECTRUM

THE MICRO ENQUIRER THE BBC MICRO

THE MICRO ENQUIRER: THE ATARI

THE MICRO ENQUIRER: THE COMMODORE 64

Benjamin Woolley and Christopher Bidmead

THE REALLY EASY GUIDE TO HOME COMPUTING:

ZX SPECTRUM

THE REALLY EASY GUIDE TO HOME COMPUTING:

COMMODORE 64

Sue Beasley and Ruth Clark

MSX: AN INTRODUCTION

Jonathan Pearce and Graham Bland

LOGO PROGRAMMING

Anne Moller

In association with Personal Computer World

BEST OF PCW: SOFTWARE FOR THE COMMODORE 64

BEST OF PCW: SOFTWARE FOR THE SPECTRUM

BEST OF PCW: SOFTWARE FOR THE BBC MICRO

BEST OF PCW: SOFTWARE FOR THE DRAGON

BEST OF PCW: SOFTWARE FOR THE ELECTRON

BEST OF PCW: SOFTWARE FOR THE ATARI

25 NEW PROGRAMS FOR THE SPECTRUM

E A Jackson

THE COMMODORE 64 HANDBOOK

THE ADVANCED COMMODORE 64 HANDBOOK

THE ATARI XL HANDBOOK

THE ORIC ATMOS HANDBOOK

Peter Lupton and Frazer Robinson

THE ATARI XL GRAPHICS HANDBOOK

Richard Hawes

THE MICRONET HANDBOOK

Barbara Hickford

EDUCATIONAL PROGRAMS FOR THE DRAGON 32

EDUCATIONAL PROGRAMS FOR THE SPECTRUM

35 EDUCATIONAL PROGRAMS FOR THE BBC MICRO

EDUCATIONAL PROGRAMS FOR THE ELECTRON

Ian Hurray