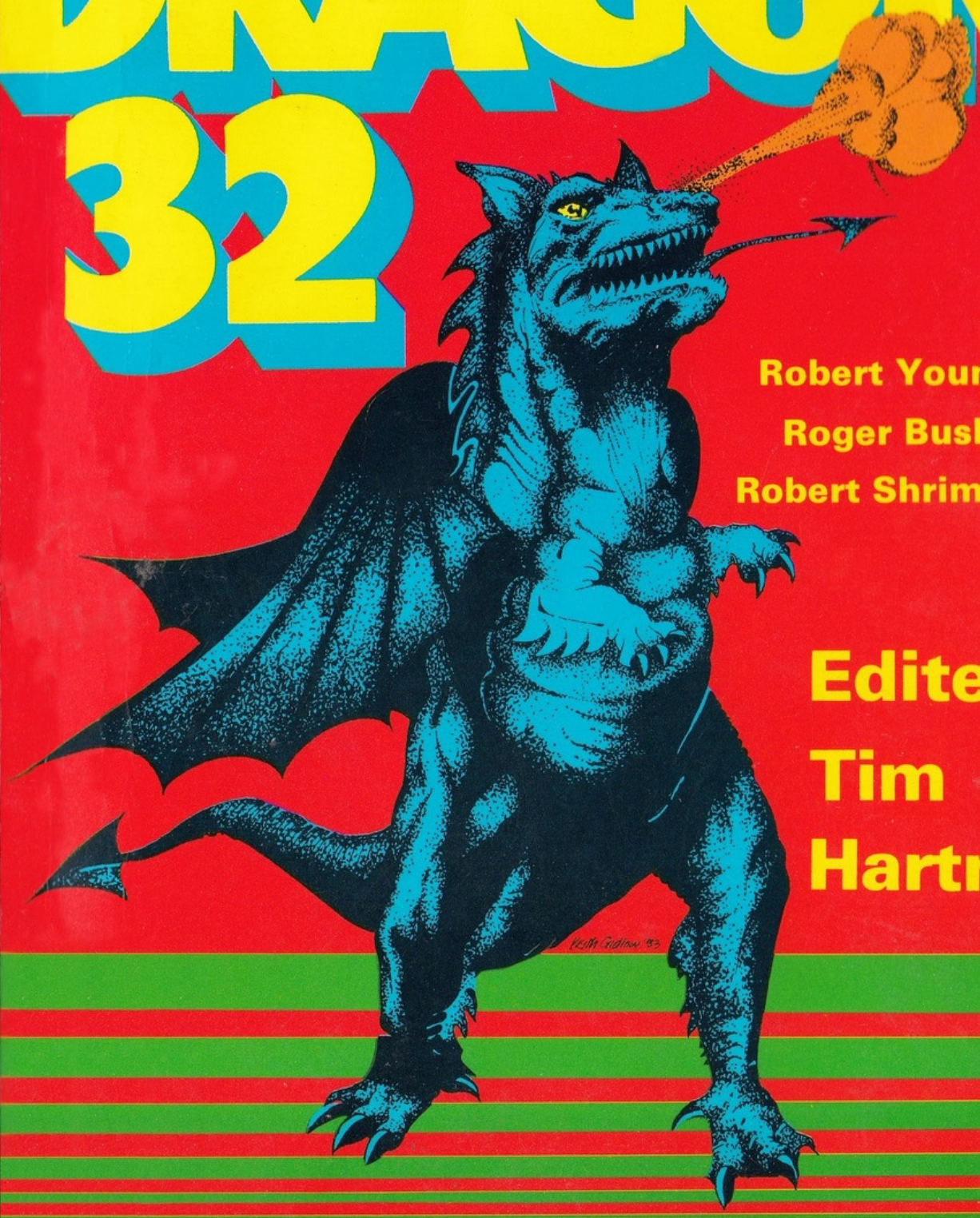
DYNAMIC GAMES FOR YOUR DRAGON 32



Robert Young
Roger Bush
Robert Shrimpton

Edited:
Tim
Hartnell

DYNAMIC GAMES FOR YOUR DRAGON 32

Robert Young, Roger Bush and Robert Shrimpton

Additional programs—Alan Blackman
Illustrations—John Walz

Interface 'Success in the Fast Lane' programming series
Foreword by Tim Hartnell

Dedication:
For Carol and Adam, whose support made it happen

INTERFACE
PUBLICATIONS
44-46 Earls Court Road, London W8 6EJ
Cover design by Keith Gidlow

First published Interface Publications, 1983

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

CONDITIONS OF SALE

Any enquiries regarding the contents of this book should be directed by mail to Interface Publications, 44-46 Earls Court Road, London, W6 6EJ

ISBN 0 907563 26 0

First printing April, 1983

Printed in England by Commercial Colour Press, London E.7.

Contents:

INTRODUCTION

A good computer program starts with one original idea, which is nurtured and allowed to grow in its own good time.

From the moment the first keys are pressed on the Dragon 32, to the time when the final program starts to roll off the printer, programmers often find they are undergoing a process of discovery. So it was with this book. Many times, the authors found the computer was bringing its own influence to bear, helping to shape programs into their final form. In many cases, they found the Dragon appeared to have as much to do with the creative process as the programmers did, as the screen format and extensive colour and sound facilities *demanded* to be used to the full.

Robert Young, Roger Bush and Robert Shrimpton did not hurry this book. They wanted the programs to unfold, so they would be a true reflection of the capabilities of the Dragon 32. The extensive range of programs in this book suggests they approached the task in the right way. You're sure to have as much fun running and developing the games in this book as the two Roberts and Roger did when writing them.

Tim Hartnell,
Editor,
London, March 1983

Author of *THE PERSONAL COMPUTER GUIDE, THE ZX SPECTRUM EXPLORED* and *49 EXPLOSIVE GAMES FOR THE ZX81*

ADVENTURE:

MAGIC CASTLE
DIAMOND JIM
NUCLEAR SUBMARINE
SNARK ATTACK
TREASURE HUNT
TUNNEL TREK

MAGIC CASTLE

When this game begins you are standing on the doorstep of a magic castle. This castle is the home of an evil wizard who has cast a spell over the surrounding countryside. You must enter the castle and kill the wizard to end the spell. The wizard is very powerful and has protected himself with a bodyguard of goblins.

The castle has no doors, but being a magic castle it will let you into certain rooms if you shout the number of the room you wish to go to. The magic will transport you from outside the castle into one of the rooms. When you get there you will be told which rooms you can move to. You will also be told if the wizard or his goblins are nearby.

Some rooms are magic rooms and when you arrive in them they will move you off somewhere else.

To fight the wizard you are carrying a magic bow and three arrows. You can't be in the same room as the wizard and live, so you must use a magic bow to fire an arrow through the wall to hit the wizard.

When you are in a room and you have been warned that the wizard is nearby, you may choose to shoot the arrow into one of the available rooms. To do so enter the number of the room preceded by a minus sign. For example −32.

You only have three arrows, so use them wisely. However you may be lucky and find other arrows as you wander around the castle.

```
10 REM MAGIC CASTLE
20 GOSUB 960
30 CLS
40 DIMA(30)
50 DIMB(30)
60 G=3
70 FOR Z=0 TO 30
80 A(Z)=0
90 NEXT Z
100 FOR Z=0 TO 10
110 A=RND(30)
120 IF A(A)=1 THEN 110
130 A(A)=1
140 B(Z)=A+11
150 NEXT Z
160 PRINT@ 10, "MAGIC HERE"
170 FOR C=1 TO 1000:NEXT C
180 X=10+2*RND(16)
190 PRINT@ 32, "YOU ARE BEING SENT TO RO
OM ";X
200 FOR QQ=1 TO 1000:NEXT QQ
210 Y=7
220 CLS
230 PRINT@ 98, "YOU ARE NOW IN ROOM *
X
240 A=-1
250 FOR Z=0 TO 15
260 IF B(Z)=X THEN A=INT(Z/5)
270 NEXT Z
280 IF A=0 THEN 160
290 IF A=1 THEN CLS ELSE GOTO 350
300 FOR DD=1 TO 500:NEXT
310 PRINT@ 197, "THERE ARE GOBLINS HERE"
```

```
320 FOR DD=1 TO 500:NEXT
330 PRINT TAB(6) "THEY HAVE KILLED YOU"
340 GOTO 770
350 IF A=2 AND RND(2)=2 THEN PRINT "you
found the wizard":FOR DD=1 TO 1000:NEXT:
GOTO 690
360 IF A=3 AND RND(2) =2 THEN PRINT "YOU
 HAVE FOUND AN ARROW":G=G+1
370 IF A>0 THEN 910
380 PRINT@ 129, "YOU CAN MOVE TO:";X-1;"
*";X+1;"*";X+Y
390 FOR Z=0 TO 2
400 A(Z)=0
410 NEXT Z
420 FOR Z=0 TO 10
430 D=B(Z)-X
440 IF ABS(D)=1 OR D=Y THEN A(INT(Z/5))=

450 NEXT Z
460 D=ABS(D)
470 IF D=2 OR D=6 OR D=8 THEN A(2)=1
480 IF A(0)=1 THEN PRINT TAB(6)"+++ MAGI
C NEARBY +++"
490 IF A(1)=1 THEN PRINT TAB(6) "*** GOB
LINS NEARBY ***"
500 IF A(2)=1 THEN PRINT TAB(5) "*+* WIZ
ARD NEARBY *+*"
510 A=2
520 PRINT "WHICH ROOM DO YOU WISH TO ENT
ER";
530 INPUT M
540 CLS
```

```
550 FOR DD=1 TO 500:NEXT
560 IF M<0 THEN 610
570 IF M < 1 OR M > 45 THEN PRINT "CANNO
T MOVE THERE":GOTO 530
580 X=M
590 Y=-Y
600 GOTO 220
610 IF RND(10)>6 THEN 630
620 IF M= -B(10) THEN 860
630 G=G-1
640 PRINT@ 234,  YOU MISSED"
650 PRINT TAB(7) G;  ARROWS LEFT"
660 FOR DD=1 TO 500:NEXT
670 CLS
680 IF G>0 THEN 230
690 CLS RND(8)
700 X=RND(4)
710 IF X=1 THEN Y$="TOAD"
720 IF X=2 THEN Y$="TREE"
730 IF X=3 THEN Y$="RABBIT"
740 IF X=4 THEN Y$="WORM"
750 PRINT@ 192, TAB(3) "YOU HAVE BEEN TU
RNED INTO A
760 PRINT TAB(14)Y$
770 FOR DD=1 TO 6
780 FOR SS=20 TO 100 STEP 20
790 SOUND SS,1
800 NEXT
810 FOR SS=120 TO 1 STEP -20
820 SOUND SS,1
830 NEXT
840 NEXT
850 END
```

```
860 IF A=2 AND RND(2)=2 GOTO 360
870 CLS RND(8)
880 FOR DD=1 TO 1000:NEXT
890 PRINT@ 192, TAB(4) "YOU HAVE KILLED
THE WIZARD"
900 GOTO 770
910 FOR DD=1 TO 1000:NEXT
920 CLS RND(8)
930 PRINT@ 192, TAB(4) "YOU HAVE NO ARRO
WS LEFT"
940 FOR DD=1 TO 500:NEXT
950 GOTO 770
960 FOR JJ=1 TO 5
970 CLS RND(8)
980 PRINT@ 192, TAB(5) "**** MAGIC CASTL
E ****"
990 FOR SS=200 TO 220 STEP 5
1000 SOUND SS,1:NEXT
1010 PRINT@ 192, TAB(5) "**************
*******"
1020 FOR SS=222 TO 202 STEP -2
1030 SOUND SS,1:NEXT
1040 NEXT JJ
1050 RETURN
```

DIAMOND JIM

Diamond Jim is searching through the tunnels of an ancient temple looking for a fabulous, long-lost diamond of enormous size.

All he has with him to help him in the hunt is a strange invention called a 'diamond detector'. This sensitive instrument can detect the presence of a large diamond and point the way toward it. Unfortunately the read-out of the diamond detector takes a bit of practice to interpret.

Each time the game is run the maze of tunnels inside the temple will be different and the diamond will be hidden in a different location. You will be given a brief look at a map of the maze at the beginning of the game.

The computer will display the move number, the diamond detector's readout and the moving instructions on the screen. It will also tell you what lies in the four directions it is possible to travel. For example:- NORTH: WALL
SOUTH: OPEN
EAST: OPEN
WEST: WALL

You may then enter the direction you wish to travel. You may also ask the computer for help if you get lost. The computer will then show you the map of the maze again. Asking for help will cost you a penalty of 15 moves!

Lines 370 to 460 print the maze on the screen. Diamond Jim is shown as an asterisk. Lines 280 to 320 act on your input and move you around the array set up in lines 470 to 640.

```
10 REM DIAMOND JIM
20 CLS
30 GOSUB 970
40 GOSUB 470
50 GOSUB 370
60 M=M+1
70 CLS:PRINT TAB(7);"MOVE NUMBER";M
80 PRINT "NORTH: ";
90 IF A(D+1,E)=S THEN PRINT "OPEN"
100 IF A(D+1,E)=X THEN PRINT "WALL"
110 PRINT "SOUTH: ";
120 IF A(D-1,E)=S THEN PRINT "OPEN"
130 IF A(D-1,E)=X THEN PRINT "WALL"
140 PRINT "EAST: ";
150 IF A(D,E+1)=S THEN PRINT "OPEN"
160 IF A(D,E+1)=X THEN PRINT "WALL"
170 PRINT "WEST ";
180 IF A(D,E-1)=S THEN PRINT "OPEN"
190 IF A(D,E-1)=X THEN PRINT "WALL"
200 PRINT "DIAMOND INDICATOR":PRINT TAB(
6);"READS";100*(ABS(Z-D)+ABS(Y-E))+Y-E
210 PRINT "DIRECTION?"
220 PRINT "(N)ORTH:(S)OUTH:(E)AST:(W)EST
:    (H)ELP"
230 INPUT A$:IF A$="" THEN230
240 IF A$="N" AND A(D+1,E)=X THEN 230
250 IF A$="S" AND A(D-1,E)=X THEN 230
260 IF A$="E" AND A(D,E+1)=X THEN230
270 IF A$="W" AND A(D,E-1)=X THEN 230
280 IF A$="H" THEN GOSUB 370
290 IF A$="N" THEN D=D+1
300 IF A$="S" THEN D=D-1
310 IF A$="E" THEN E=E+1
```

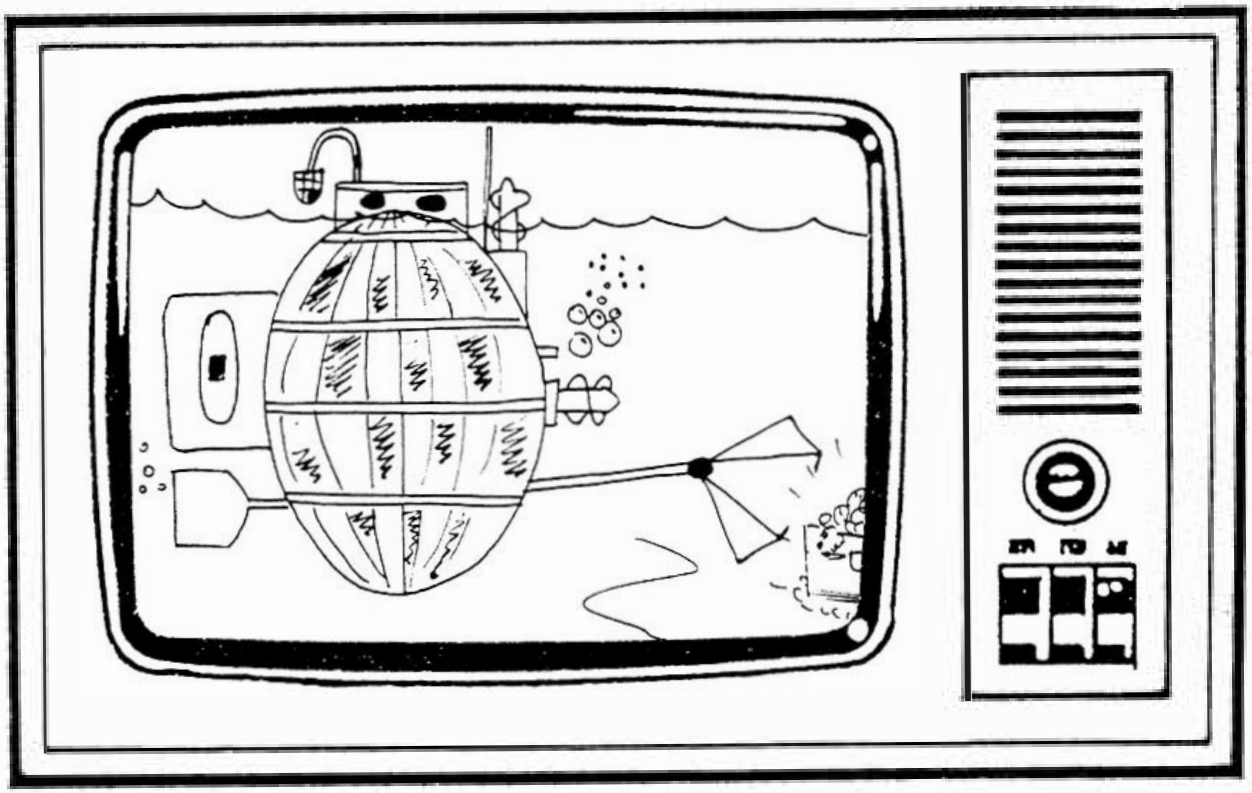
```
320 IF A$="W" THEN E=E-1
330 IF Z=D AND Y=E THEN 350
340 GOTO 60
350 FOR DD=1 TO 1500:NEXT
360 GOTO 750
370 CLS
380 FOR B=15 TO 1 STEP-1:FOR C=1 TO 15
390 IF A(B,C)=X THEN PRINT TAB(6) CHR$(1
91);
400 IF B=D AND C=E THEN PRINT "*";:GOTO
420
410 IF A(B,C)=S THEN PRINT CHR$(175);
420 NEXT:PRINT:NEXT
430 M=M+15
440 FOR J=1 TO 2000:NEXT J
450 CLS:A(D,E)=S
460 RETURN
470 DIMA(15,15)
480 B=INT(RND(0)*3)+1
490 Z=14:Y=14
500 IF B=2 THEN Y=2
510 IF B=3 THEN Z=2
520 X=1:S=2
530 FOR B=1 TO 15:FOR C=1 TO 15
540 A(B,C)=X:IF RND(0)>.9 THEN A(B,C)=S
550 IF C<2 OR C>14 OR B<2 OR B>14 THEN A
(B,C)=X
560 NEXT:NEXT
570 D=2:E=2
580 FOR F=1 TO 68
590 READ B:READ C
600 A(B,C)=S
610 NEXT
```

```
620 M=-15
630 SOUND 200,1
640 RETURN
650 DATA 2,2,2,3,2,4,2,5,2,6,2,7
660 DATA 3,7,4,7,5,7,5,6,5,5,5,4,5,3,6,3
670 DATA 7,3,7,4,7,5,7,6,7,7,7,8,7,9,9,8
680 DATA 9,9,10,8,10,7,10,6,10,5,10,4,8,
8
690 DATA 10,3,11,3,12,3,13,3,14,3,14,2,7
,10
700 DATA 6,10,5,10,4,10,3,10,2,10,2,11,2
,12
710 DATA 2,13,2,14,6,11,6,12,6,13,6,14,7
,12
720 DATA 14,12,8,12,8,14,9,12,9,13,9,14,
10,12
730 DATA 11,9,11,10,11,11,11,12,12,9,13,
9,13,10
740 DATA 13,11,13,12,13,13,13,14,14,14
750 CLS:FOR CC=1 TO 250 STEP 10
760 SOUND CC,1
770 NEXT
780 FOR CC=10 TO 220 STEP 10
790 SOUND CC,1
800 NEXT
810 FOR CC=40 TO 180 STEP 10
820 SOUND CC,1
830 NEXT
840 FOR CC=160 TO 20 STEP -10
850 SOUND CC,1
860 NEXT
870 FOR CC=240 TO 20 STEP -10
880 SOUND CC,1
```

```
890 NEXT
900 FOR CC=250 TO 1 STEP -10
910 SOUND CC,1
920 NEXT
930 CLS RND(8)
940 PRINT@ 192, TAB(6) "*** YOU FOUND IT
 ***"
950 PRINT TAB(6) "YOU TOOK ";M;" MOVES"
960 END
970 PCLS
980 PMODE 0,1
990 SCREEN 1,1
1000 LINE (120,8)-(12,84),PSET
1010 LINE (12,84)-(120,184),PSET
1020 LINE (120,184)-(232,84),PSET
1030 LINE(232,84)-(120,8),PSET
1040 PAINT (128,92),5,5
1050 FOR DD=1 TO 1000:NEXT
1060 CLS RND(8)
1070 CC=0
1080 PRINT@ 192, TAB(4) "**************
********"
1090 SOUND 200,1
1100 FOR DD=1 TO 75:NEXT
1110 PRINT@ 192, TAB(4) "****  DIAMOND J
IM  ****"
1120 SOUND 150,1
1130 CC=CC+1
1140 FOR DD=1 TO 200:NEXT
1150 IF CC=10 THEN RETURN
1160 GOTO 1080
```

NUCLEAR SUBMARINE

You are the captain of a nuclear submarine on patrol in the Atlantic Ocean. Somewhere in the waters around you is an enemy submarine. You must find it and use your torpedoes to put it out of action. You must get close to the enemy for your torpedoes to be effective, but do *not* get too close. An underwater collision between two nuclear submarines could produce a spectacular explosion.



Firing your torpedoes or being hit by the enemy submarine uses up your fuel supply. You only have a limited amount available, so use it carefully.

The game ends when you run out of fuel, or when either your submarine or the enemy one is sunk.

Your computer will tell you the co-ordinates of your position, and the approximate position of the enemy submarine. You will receive constantly updated reports of your position, your energy level and you'll be told when your fuel is running low, and when you are under attack.

The GOSUB in line 20 controls the sound and graphics at the start of the game. The GOSUB in line 40 sets up the variables in the game. The GOSUB in line 50 triggers the print-out containing your status on each turn.

Line 750 checks to see if you have collided with the enemy submarine. Lines 770 to 840 check your position in relation to the enemy's position, and provides information about his location. (To demonstrate just how much fun it can be to continue to work on a program long after it appears to be completed, we took NUCLEAR SUBMARINE and twisted it almost out of recognition, to produce the program SNARK ATTACK, which follows NUCLEAR SUBMARINE.)

```
10 REM NUCLEAR SUBMARINE
20 GOSUB 1000
30 CLS
40 GOSUB 860
50 GOSUB 670
60 IF L<0 THEN 470
70 PRINT "WHAT IS YOUR ORDER, CAPTAIN"
80 L=L-0.25
90 PRINT "N,S,E,W,T(ORPEDO),F(ORWARD)":P
RINT"B(ACK)";
100 INPUT Z$
110 IF Z$="T" GOSUB 310
120 IF Z$="N" THEN X=X-1
130 IF Z$="S" THENX=X+1
```

```
140 IF Z$="E" THEN Y=Y+1
150 IF Z$="W" THEN Y=Y-1
160 IF Z$="F" THEN Z=Z-1
170 IF Z$="B" THEN Z=Z+1
180 GOSUB 560
190 A=A+(RND(3)-1)-(RND(3)-1)
200 IF A<1 OR A>10 THEN 190
210 B=B+(RND(3)-1)-(RND(3)-1)
220 IF B<1 OR B>10 THEN 210
230 C=C+(RND(3)-1)-(RND(3)-1)

250 IF C<1 OR C>10 THEN 210
260 GOTO 50
270 L=L-0.75
280 IF ABS(A-X)>3 OR ABS(B-Y)>3 OR ABS(C
-Z)>3 THEN PRINT "OUT OF RANGE ***"
290 FOR J=1 TO 1000
300 NEXT J
310 IF ABS(A-X)>3 OR ABS(B-Y)>3 OR ABS(C
-Z)>3 THEN RETURN
320 PRINT "ORDER TO FIRE UNDERSTOOD CAPT
AIN"
330 FOR J=1 TO 1000
340 NEXT J
350 IF RND(0)>0.7 THEN400
360 PRINT ">>>>>missed"
370 FOR J=1 TO 1000
380 NEXT J
390 GOTO 420
400 PRINT "YOU DID IT, SIR"
410 T=T+1
420 RETURN
430  CLS:PRINT@ 170,"end of game"
```

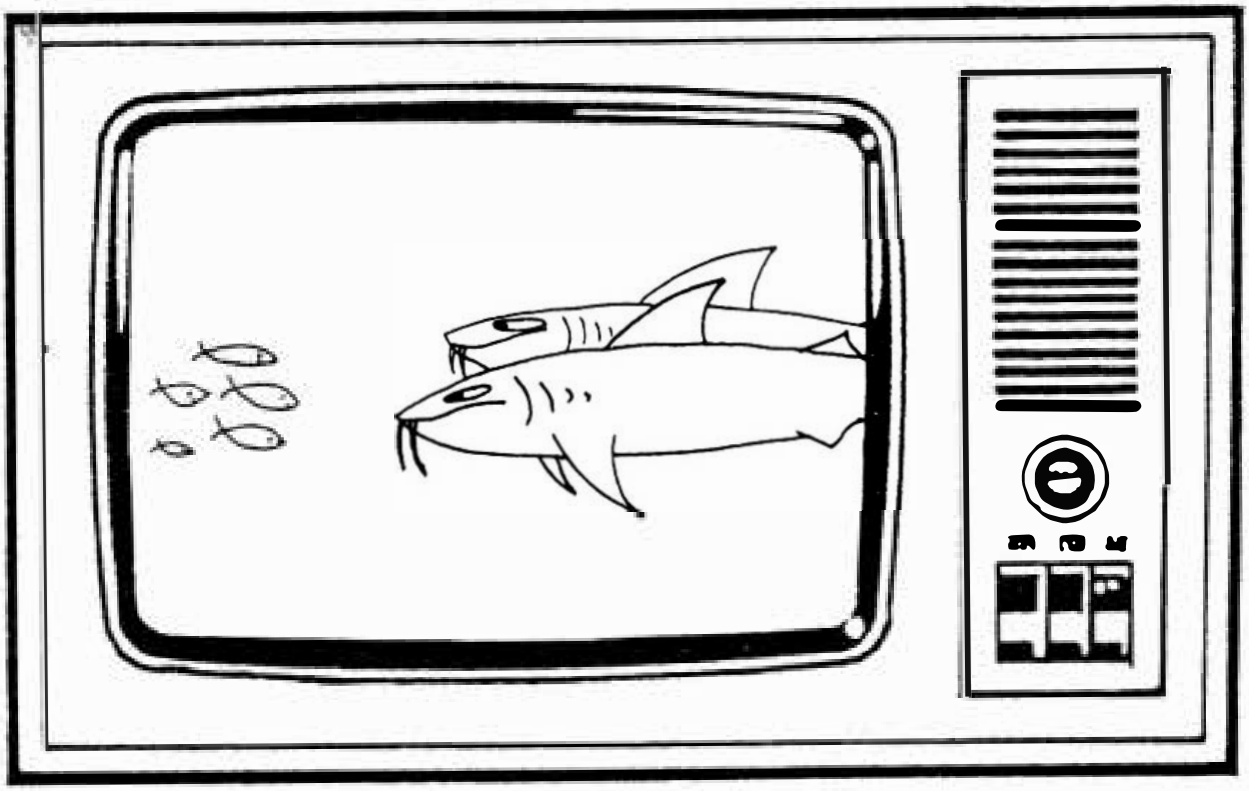
```
440 IF TI=0 THEN PRINT@ 196, "you stayed
 under too long"
450 IF L>0 THEN PRINT TAB(5) "you have b
een defeated"
460 IF L=0 THEN PRINT@ 232, "all energy
used"
470 END
480 PRINT "you have collided with the"
490 PRINT "enemy submarine"
500 END
510 IF ABS(A-X)>3 OR ABS(B-Y)>3 OR ABS(C
-Z)>3 THEN RETURN
520 IF RND(0)>0.75 THEN RETURN
530 PRINT "warning***enemy is firing at
us"
540 FOR J=1 TO 1000
550 NEXT J
560 IF RND(0)>0.7 THEN 640
570 PRINT "the enemy has hit us, sir"
580 L=L-7
590 IF L<0 THEN470
600 FOR J=1 TO 1000
610 NEXT J
620 RETURN
630 PRINT "the enemy missed"
640 FOR J=1 TO 1000
650 NEXT J
660 RETURN
670 CLS
680 PRINT "ENERGY REMAINING:";L;"ERGS"
690 TI=TI-1
700 IF TI=0 THEN 430
710 PRINT "TIME ";TI
```

```
720 IF L<3 THEN PRINT "ENERGY LEVEL IS L
OW"
730 PRINT "TALLY : ";T
740 PRINT "YOU ARE AT ";X;", ";Y;", ";Z
750 IF A=X AND B=Y AND C=Z THEN 480
760 PRINT@192,"THE ENEMY IS ";
770 IFA<>X OR B<>Y THEN PRINT "TO THE ";
780 IF A<X THEN PRINT "NORTH";
790 IF A>X THEN PRINT "SOUTH";
800 IF B>Y THEN PRINT "EAST";
810 IF B<Y THENPRINT "WEST";
820 IF C=Z THEN PRINT@224," OF YOU"
830 IF C>Z THEN PRINT@224," BEHIND YOU"
840 IF C<Z THEN PRINT@224," IN FRONT OF
YOU"
850 RETURN
860 DIM Z$(1)
870 L=25+RND(30)
880 T=0
890 TI=35
900 A=RND(10)
910 B=RND(10)
920 C=RND(10)
930 X=RND(10)
940 Y=RND(10)
950 Z=RND(10)
960 RETURN
1000 CC=0
1010 CLS0
1020 SS=RND(25)+200
1030 TT=RND(2)
1040 SOUND SS,TT:CC=CC+1
```

```
1050 IF CC=10 GOTO 1070
1060 GOTO 1020
1070 PCLS
1080 PMODE 1,1
1090 SCREEN 1,0
1100 PAINT (0,0),2
1110 PAINT (0,0),1
1120 PAINT (0,0),4
1130 PAINT (0,0),3
1140 CLS RND(8)
1150 FOR SS=150 TO 250 STEP 5
1160 SOUND SS,1
1170 NEXT SS
1180 PRINT@ 192, TAB(4) "xxx NUCLEAR SUB
MARINE xxx"
1190 FOR T=100 TO 5 STEP -5
1200 SS=RND(25)+150
1210 SOUND SS,1
1220 FOR OD=1 TO T:NEXT:NEXT
1230 FOR CC=1 TO 1000:NEXT CC:RETURN
```

SNARK ATTACK

This program poses a daunting task. As a SCUBA diver you must enter the depths of the Pacific Ocean and hunt the fierce Great White Snark. This very rare creature is a native to the waters and coastline of Australia. The Snark is the result of an highly unlikely mating between a white pointer



shark and the Australian tiger snake. This chance encounter has resulted in a ferocious creature which is at home both on the land and in the ocean. However it rarely ventures far from the water.

With the aid of your computer you can hunt the Snark in its favourite lurking place, the ocean's deeps.

The computer will keep track of your oxygen supply, the amount of time you have spent underwater and it will provide you with clues to the hideout of the Snark.

Lines 990 to 1060 run the simple cartoon at the start of the game. Lines 150 to 210 uses your input (Z$) to move you around the ocean. Line 890 gives your oxygen for the start of the game, while lines 120, 310 and 600 decreases your oxygen supply. Line 100 tests L to see if you have any oxygen remaining, and lines 920 to 970 determine the starting positions for both the Snark and yourself.

```
10 REM SNARK ATTACK
20 GOSUB 990
30 CLS(3)
40 PRINT@ 192, TAB(6) "*#* SNARK ATTACK
*#*
50 FOR DD=1 TO 20
60 SOUND RND(50)+175,1
70 NEXT DD
80 GOSUB 880
90 GOSUB 690
100 IF L<0 THEN 460
110 PRINT"WHICH DIRECTION"
120 L=L-0.25
130 PRINT "N,S,E,W,G(SPEARGUN),F(ORWARD)
":PRINT"B(ACK)";
140 INPUT Z$
150 IF Z$="G" GOSUB 310
160 IF Z$="N" THEN X=X-1
170 IF Z$="S" THENX=X+1
180 IF Z$="E" THEN Y=Y+1
190 IF Z$="W" THEN Y=Y-1
200 IF Z$="F" THEN Z=Z-1
210 IF Z$="B" THEN Z=Z+1
```

```
220 GOSUB 540
230 A=A+(RND(3)-1)-(RND(3)-1)
240 IF A<1 OR A>10 THEN 230
250 B=B+(RND(3)-1)-(RND(3)-1)
260 IF B<1 OR B>10 THEN 250
270 C=C+(RND(3)-1)-(RND(3)-1)
290 IF C<1 OR C>10 THEN 250
300 GOTO 90
310 L=L-0.75
320 IF ABS(A-X)>3 OR ABS(B-Y)>3 OR ABS(C
-Z)>3 THEN PRINT TAB(6) "*** TOO FAR AWA
Y ***"
330 FOR J=1 TO 1000
340 NEXT J
350 IF ABS(A-X)>3 OR ABS(B-Y)>3 OR ABS(C
-Z)>3 THEN RETURN
360 PRINT"SPEARGUN ";:IF RND(3)>1THEN PR
INT"FIRED" ELSE PRINT"MISSFIRED":GOTO410
370 FOR J=1 TO 1000
380 NEXT J
390 IF RND(0)>0.7 THEN440
400 PRINT ">>>>>missed"
410 FOR J=1 TO 1000
420 NEXT J
430 GOTO 460
440 GOTO 1110
450 T=T+1
460 RETURN
470 CLS:PRINT@202,"end of game"
480 IF L>0 THEN PRINT@135,"you have been
 eaten"
```

```
490 IF L=0 THEN PRINT@327,"all oxygen us
ed"
500 GOTO1100
510 CLS:PRINT@ 192, TAB(2) "you have col
lided with the"
520 PRINT@ 224, TAB(2) "snark and you ha
ve become its   lunch"
530 GOTO 1100
540 IF ABS(A-X)>3 OR ABS(B-Y)>3 OR ABS(C
-Z)>3 THEN RETURN
550 IF RND(0)>0.75 THEN RETURN
560 PRINT "warning***the snark is attack
ing you":SOUND230,1:SOUND240,1:SOUND250,
5
570 FOR J=1 TO 1000
580 NEXT J
590 IF RND(0)>0.7 THEN 650
600 L=L-15
610 IF L<0 THEN470
620 FOR J=1 TO 1000
630 NEXT J
640 RETURN
650 PRINT "the snark missed"
660 FOR J=1 TO 1000
670 NEXT J
680 RETURN
690 CLS
700 PRINT "OXYGEN  REMAINING:";L
710 TI=TI-1
720 IF TI=0 THEN 460
730 PRINT "TIME : ";TI
740 IF L<3 THEN PRINT "OXYGEN LEVEL IS L
OW"
```

```
750 PRINT "DEAD SNARKS :";T
760 PRINT "YOU ARE AT ";X;", ";Y;", ";Z
770 IF A=X AND B=Y AND C=Z THEN 560
780 PRINT@192,"THE SNARK IS ";
790 IFA<>X OR B<>Y THEN PRINT "TO THE:-
";
800 IF A<X THEN PRINT "NORTH";
810 IF A>X THEN PRINT "SOUTH";
820 IF B>Y THEN PRINT "EAST";
830 IF B<Y THENPRINT "WEST";
840 IF C=Z THEN PRINT@224," OF YOU"
850 IF C>Z THEN PRINT@224," BEHIND YOU"
860 IF C<Z THEN PRINT@224," IN FRONT OF
YOU"
870 RETURN
880 DIM Z$(1)
890 L=25+INT(RND(0)*30)
900 T=0
910 TI=35
920 A=RND(10)
930 B=RND(10)
940 C=RND(10)
950 X=RND(10)
960 Y=RND(10)
970 Z=RND(10)
980 RETURN
990 CLS 6
1000 FOR X=1 TO 63:FOR Y=23 TO 31
1010 SET(X,Y,3)
1020 NEXT Y:NEXT X
1030 FOR CC=0 TO 62 STEP 2
1040 SET(CC,20,5):SOUND 2,2:FOR DD=1 TO
5:NEXT:SOUND 1,2:SET(CC,20,6)
```

```
1050 NEXT CC
1060 RETURN
1070 CC=0
1080 SS=RND(50)+100:SOUNDSS,1
1090 CC=CC+1:IFCC=20THENENDELSE GOTO1080
1100 PLAY"O2;L3;C;C;L8;C;L6;C;E;D;D;C;C;
O1;B;O2;L3;C;O1;G;C;":END
1110 CLS RND(8)
1120 PRINT@ 192, "YOU HAVE KILLED THE GR
EAT WHITE"
1130 PRINT@ 224, TAB(12) "SNARK"
1140 FOR T=1 TO 10
1150 CLS RND(8)
1160 FOR Q=4*T TO 4*T+10 STEP 2:SOUND Q,
1:NEXT
1170 PRINT@ 192, "YOU HAVE KILLED THE GR
EAT WHITE"
1180 PRINT@ 224, TAB(12) "SNARK"
1190 FOR Q=225 TO 230 STEP -T:SOUND Q,1:
NEXT:NEXT
1200 END
```

TREASURE HUNT

In a dark cave system, deep underground, a fortune in gold can be found it you are brave enough to hunt for it. You must be careful not to fall into pools of quicksand or bump into one of the strange creatures which live in the caves.

You have 25 minutes of 'game time' in which to wander around and obtain as much wealth as possible. The game ends when this time runs out, or even sooner if you don't avoid the monsters and quicksand.

At various times during the game the computer will provide you with a quick look at a map of the cave system. Your position will be marked with a letter H. A $ shows the location of the treasures. Quicksand is marked with a Q and the letter M can either mean a monster or a magic cave. A dot indicates an open space.

You begin your search in cave number 55. The computer will then ask you which direction you wish to move. Enter N for north, S for south and so on. The computer will also give you very brief advice about hazards or other features may be nearby. However, it won't tell you how close, or in which direction from you they are located.

The random number generator in line 880 gives you a two in three chance of being shown the cave map each turn. The plan is printed by lines 890 to 920. Z$ in line 250 is your input direction. Lines 300 to 370 act on this input and move you in the requested direction and then check to see what is lurking at your new location. The subroutine from lines 950 to 1020 provides the graphics at the start of the game.

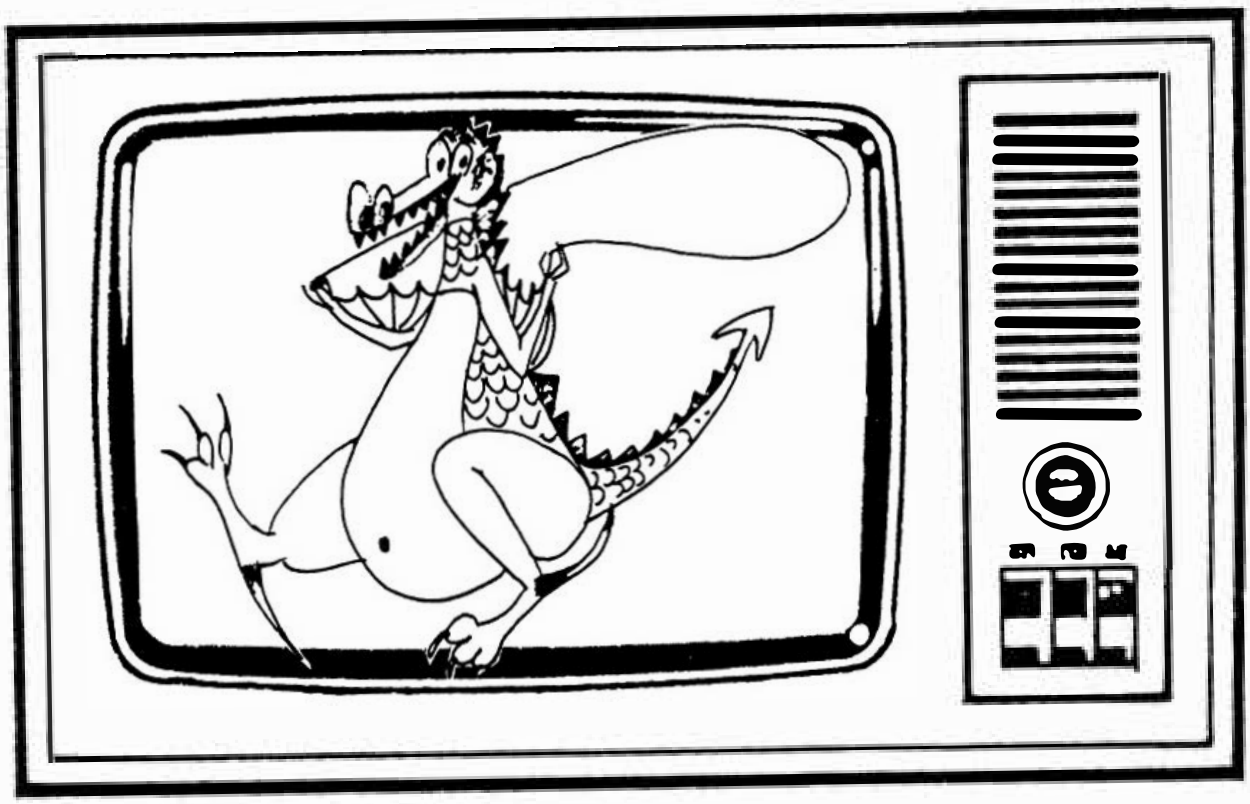
```
10 REM TREASURE HUNT
20 GOSUB 950
30 XX$="
40 GT=RND(10)+20
50 CLS RND(8)
60 PRINT@ 192, TAB(5) "### TREASURE HUNT
 ###"
70 DIMA(100):M=0:Q=0:L=0:G=0:AR=6
80 FOR B=1 TO 100:A(B)=46
90 IF B<12 OR B>90 OR 10*INT(B/10)=B OR
10*INT(B/10)=B-1 THEN A(B)=166
100 NEXT
110 FOR B=1 TO 5:RESTORE:FOR D=1 TO 5
120 Z=INT(RND(0)*76)+12:IF A(Z)=166 THEN
 120
130 READ C:A(Z)=C
140 NEXT:NEXT
150 DATA166,218,72,81,36
160 FOR B=1 TO 8:READ P(B):NEXT:DATA-11,
-10,-9,-1,1,9,10,11
170 E=55
180 A(E)=72
190 CLS:GOSUB 880
200 Q=INT(RND(0)*7)
210 IF Q=0 AND E<>55 GOSUB 880
220 PRINT@353,"CAVE";E
230 IF G>0 THEN PRINT@363,G;"GOLD"
240 GOSUB 610
250 GOSUB880:SOUND200,1:PRINT@448,;:INPU
T"DIRECTION "; Z$:U=0
260 PRINT@416,XX$
```

```
270 IF Z$="N" AND A(E-10)=166 OR Z$="S"
AND A(E+10)=166 OR Z$="E" AND A(E+1)=166
 THEN U=1
280 IF Z$="W" AND A(E-1)=166 THEN PRINT@
384,"blocked cave":FOR T=1 TO 2000:NEXT:
GOTO 250
290 A(E)=46:IF Z$="N" THEN E=E-10
300 IF Z$="S" THEN E=E+10
310 IF Z$="E" THEN E=E+1
320 IF Z$="W" THEN E=E-1
330 IF Z$="F" GOSUB 730
340 IF A(E)=218 GOSUB 400
350 IF A(E)=77 GOSUB 440
360 IF A(E)=81 GOSUB 510
370 IF A(E)=36 GOSUB 570
380 H=H+1:IF H>GT THEN 850
390 GOTO 200
400 PRINT "[illegible]"
410 A(E)=46
420 E=INT(RND(0)*200+12):IF A(E)=166 THEN
 420
430 RETURN
440 CLS:SOUND 10,5:PRINT@10,"monster here
"
450 FOR T=1 TO 1000:NEXT
460 M=RND(0):IF M<.2 THEN PRINT" IT IS R
UNNING AWAY":RETURN
470 PRINT "IT HAS SEEN YOU..."
480 FOR T=1 TO 1000:NEXT
490 IFM>.85 THEN PRINT "AND FLEES":FOR T
=1 TO 999:NEXT:RETURN
500 PRINT "AND EATS YOU!!!":SOUND20,40:F
OR T=1 TO 990:NEXT:Q=9:CLS:GOTO 850
```

```
510 CLS:FOR J=1 TO 20:PRINT TAB(5*J);"HO
RRORS...";
520 PRINT "quicksand":SOUND100-J,1
530 FOR T=1 TO 10*J:NEXTT:NEXTJ
540 SOUND20,40:FOR T=1TO1000:NEXT:CLS
550 Q=9
560 GOTO 850
570 CLS:FOR J=1 TO 20: PRINT TAB(5*J);"*
WEALTH!!*":SOUND200+J,1:FOR T=1 TO 10*J:
NEXT:NEXT:CLS
580 K=INT(RND(0)*100)+100
590 PRINT@230,"YOU HAVE FOUND GOLD"
600 PRINT@294,"WORTH $";K;"!!!!":G=G+K:S
OUND100,3:FOR T=1 TO 2000:NEXT:GOSUB880:
RETURN
610 Y=1
620 L=A(E+P(Y))
630 IF L<>46 THEN 660
640 IF Y<8 THEN Y=Y+1:GOTO 630
650 IF L=46 THEN RETURN
660 PRINT@416,"NEARBY IS...";
670 IF L=166 THEN PRINT"NO PATH"
680 IF L=218 THEN PRINT "MAGIC"
690 IF L=77 THEN PRINT "MONSTER"
700 IF L=81 THEN PRINT "QUICKSAND"
710 IF L=36 THEN PRINT "GOLD"
720 FOR T= 1 TO 3000:NEXT:RETURN
730 AR=AR-1:IF AR=0 THEN PRINT "NO ARROW
S LEFT":RETURN
740 PRINT@386,AR;"ARROWS LEFT":SS=0
750 PRINT@448,;:INPUT "WHICH DIRECTION";
S$:CLS
```

```
760 IF S$="N" AND A(E-10)=77 THEN SS=1:Y
T=E-10
770 IF S$="S" AND A(E+10)=77 THEN SS=1:Y
T=E+10
780 IF S$="E" AND A(E+1)=77 THENSS=1:YT=
E+1
790 IF SS=0 THEN PRINT "NOTHING THERE":G
OTO840
800 PRINT "*** A HIT ***"
810 FOR T= 1 TO 999:NEXT:IF RND(1)>.3 TH
EN 830
820 PRINT "*** MONSTER IS DEAD ***":A(YT
)=46:G=G+INT(RND(0)*100):GOTO 840
830 PRINT "THE MONSTER IS WOUNDED"
840 FOR T=1 TO 3000:NEXT:RETURN
850 IF Q=9 GOTO 870
860 CLS:SOUND50,15:PRINT"*** YOU HAVE RU
N OUT OF AIR ***":FORW=1TO1000:NEXT
870 PRINT "YOU SURVIVED FOR ";H;" MINUTE
S    AND FOUND $";G;" OF GOLD":FORC=200T
O150STEP-5:SOUNDC,1:NEXT:END
880 A(E)=72:IF RND(3)=1THEN940
890 PRINT@0,"";:FOR J=1 TO 100
900 PRINT TAB(11) CHR$(A(J));
910 IF 10*INT(J/10)=J THEN PRINT
920 NEXT:IF Q=9 THEN END
930 WA=RND(50):FOR DD=1TOWA:NEXT
940 PRINT@0,XX$;:FORDD=1TO10:PRINTXX$:NE
XTDD:RETURN
950 CLS0
960 FOR CC=1 TO 200
970 X=RND(20)+20:Y=RND(10)+10
980 C=RND(8)
```

```
990 SET(X,Y,C)
1000 NEXT CC
1010 FOR DD=1 TO 20:SOUND 225,1:NEXT DD
1020 RETURN
```



TUNNEL TREK

Now we come to Tunnel Trek, a game which almost defies description. Most adventure-type games are based on logic, and are cleverly planned, with many challenges for the player. Tunnel Trek is nothing like that. It is more of an ordeal than an adventure.

Lines 290 to 570 print the instructions on the screen, decide what is attacking you and with what. They also give you a choice of defensive weapons. The random number generator in lines 580 to 610 decide whether or not you have defeated the attacker. Lines 600 to 610 direct the program to the appropriate PRINT routine according to the result of the random number generator.

The variable W keeps track of where you are in the cave system, S is the amount of money you have and X counts the number of fights you have been in.

The question which now remains unanswered is: Do you have the nerve and the courage to tackle Tunnel Trek?

```
10 REM TUNNEL TREK
20 CLS
30 PRINT@ 198, "****TUNNEL TREK****"
40 GOSUB 1240:CLS
50 X=0:S=30:W=1
60 PRINT@ 64, "YOU ARE AT THE START OF A
 SYSTEM OF TUNNELS AND CAVES.  YOU HAVE
THIRTY PIECES OF GOLD
70 PRINT@ 192, "WHEN YOU REACH THE END O
F THE  TUNNELS YOU MUST HAVE AT LEAST
TWENTY PIECES OF GOLD TO PAY THE TOLL CO
LLECTOR"
```

```
80 PRINT@ 362,"PRESS ENTER"
90 INPUT A$
100 IF A$="" GOTO 80
110 GOSUB 910
120 IF W<1 THEN W=RND(8)
130 PRINT "THIS IS A TUNNEL MAZE"
140 PRINT
150 IF W=>10 THEN 970
160 PRINT "THIS IS CAVE NUMBER";W
170 PRINT
180 PRINT "CAVE NO 10 IS THE EXIT"
190 PRINT
200 X=X+1
210 PRINT "THIS IS CHALLENGE NUMBER";X
220 IF S<1 THEN S=3
230 PRINT
240 PRINT "YOU HAVE";S;"GOLD PIECES"
250 GOSUB 1240
260 GOSUB  910
270 PRINT
280 K=INT(4*RND(0))+2
290 PRINT "YOU ARE NOW FACING";K;"TUNNEL
S"
300 PRINT "WHICH ONE WILL YOU TRY?"
310 INPUT A
320 GOSUB 910
330 IF RND(0)< .1 THEN 630
340 IF A<>K THEN 360
350 IF A=K THEN 630
360 K=RND(4)
370 IF K=1 THEN E$="GIANT RAT"
380 IF K=2 THEN E$="WEREWOLF"
390 IF K=3 THEN E$="TROLL"
```

```
400 IF K=4 THEN E$="ZOMBIE"
410 PRINT "THE TUNNEL IS BLOCKED BY A"
420 E=RND(4)
430 IF E=1 THEN F$="POISONED SPEAR"
440 IF E=2 THEN F$="FLAMING SWORD"
450 IF E=3 THEN F$="BOW AND ARROW"
460 IF E=4 THEN F$="CROSSBOW"
470 PRINT E$;" ARMED WITH"
480 PRINT "A ";F$
490 PRINT
500 PRINT "WHICH WEAPON DO YOU CHOOSE?"
510 PRINT
520 PRINT "A POINTED STICK(1)"
530 PRINT
540 PRINT "BARE HANDS(2)"
550 PRINT
560 PRINT "A SMALL ROCK(3)
570 INPUT B
580 C=INT(3*RND(0))+1
590 GOSUB 910
600 IF B=C THEN GOSUB 1090
610 IF B<>C THENGOSUB 1160
620 GOTO 80
630 K=INT(4*RND(0))+1
640 ON K GOSUB 670,720,770,820
650 GOTO 80
660 PRINT
670 PRINT "YOU HAVE FALLEN THROUGH"
680 PRINT " A TRAPDOOR"
690 W=W-1
700 S=S-INT(2*RND(0))-1
710 RETURN
```

```
720 PRINT "A FLOOD WASHES YOU DOWN"
730 PRINT " A SIDE TUNNEL"
740 W=W+1
750 S=S-INT(2*RND(0))-1
760 RETURN
770 PRINT "YOU HAVE BEEN HELPED BY"
780 PRINT " A PASSING GNOME"
790 S=S+INT(5*RND(0))+1
800 W=W+INT(3*RND(0))+1
810 RETURN
820 PRINT "WHOOPPEE!!! A HOARD OF"
830 PRINT " GOLD, CHOOSE UP TO 5 PIECES"
840 PRINT "BUT BE CAREFUL, THE MORE"
850 PRINT " YOU TAKE, THE MORE IT WILL"
860 PRINT " COST YOU, HOW MANY TO TAKE?"
870 INPUT D
880 IF D>5 THEN 870
890 S=S+D:W=W-INT(D/2)
900 RETURN
910 CLS
920 FOR I=1 TO 5
930 PRINT
940 NEXT
950 RETURN
960 IF W<>10 THEN RETURN
970 PRINT "YOU ARE AT THE EXIT"
980 PRINT "DO YOU HAVE ENOUGH GOLD?"
990 PRINT "PRESS ENTER TO FIND OUT"
1000 INPUT C$:CLS
1010 IF S<20 THEN PRINT "THE GUARD HAS K
ILLED YOU":ELSE GOTO 1040
1020 PRINT "YOU COULD NOT PAY THE TOLL"
1030 IF S<20 GOTO 1080
```

```
1040 PRINT "YES YOU CAN PAY THE TOLL"
1050 PRINT "YOU HAVE WON THE GAME"
1060 PRINT " AND YOU MAY KEEP THE REMAIN
ING"
1070 PRINT S;"GOLD PIECES"
1080 END
1090 PRINT "YOU HAVE FOUGHT YOUR WAY PAS
T   THE ";E$
1100 S=S+INT(3*RND(0))+1
1110 PRINT "AND HAVE";S;"GOLD PIECES"
1120 W=W+INT(3*RND(0))+1
1130 PRINT
1140 PRINT "YOU ARE APPROACHING CAVE";W
1150 RETURN
1160 PRINT "THE ";E$;" BEAT YOU AND"
1170 S=S-INT(4*RND(0))-1
1180 IF S<0 THEN S=0
1190 PRINT "LEFT YOU WITH";S;"GOLD PIECE
S"
1200 W=RND(8)
1210 IF W<1 THEN W=1
1220 PRINT "AND SENT YOU TO";W
1230 RETURN
1240 FOR M=1 TO 2000:NEXT:CLS:RETURN
```

OUT IN THE ARCADE:

ONE ARMED BANDIT
TWENTY ONE
TEN PIN BOWLING
ROULETTE
TEE OFF

ONE ARMED BANDIT

This program allows you to play the dreaded one armed bandit. At the beginning of the game, you'll be given 20 computer dollars to gamble with. Unfortunately computer dollars can only be spent inside a computer. The computer will then invite you play by pressing the enter key. Each time you play you put a dollar into the one armed bandit.

The computer then spins the three coloured squares. Three of a kind pays out $5, and, one of each also pays out $5. Any other combination means you lose your money. The game ends when you either win more than $50 in total and break the bank, or when you lose all your computer dollars and are bankrupt.

Lines 130 to 180 choose the colours to be spun. Lines 610 to 690 produce the spinning effect. Variable M keeps track of your money. Line 390 checks to see if you have broken the bank and line 400 checks if you are bankrupt.

```
10 REM ONE ARMED BANDIT
20 CLS
30 GOSUB 1130
40 CLS:PRINT@ 70, "YOUR STAKE IS $20"
50 PRINT@ 130, "THREE OF A KIND PAYS $5"
60 PRINT@ 196, "ONE OF EACH PAYS $5"
70 FOR DD=1 TO 1500:NEXT
80 CLS RND(8)
90 PRINT@ 192, TAB(2) "##  PRESS ENTER T
O PLAY  ##";
100 INPUT A
```

```
110 CLS
120 M=20
130 A=0
140 B=0
150 C=0
160 FOR D=1 TO 3
170 E=RND(3)
180 ON E GOTO 190,240,290
190 A=A+1
200 FORX=150TO159
210 GOSUB610
220 NEXTX
230 GOTO 330
240 B=B+1
250 FORX=150TO159
260 GOSUB610
270 NEXTX
280 GOTO 330
290 C=C+1
300 FORX=150TO159
310 GOSUB610
320 NEXTX
330 NEXT D
340 SOUND RND(20)+150,3
350 IF A=3 OR B=3 OR C=3 THEN 530
360 IF A=1 AND B=1 AND C=1 THEN 570
370 M=M-1
380 PRINT@ 198, "YOU LOST THAT TIME"
390 IF M>49 THEN 730
400 IF M<1 THEN 950
410 PRINT@ 265, "STAKE: $";M
420 IF M<1 OR M>49 THEN STOP
430 PRINT
```

```
440 PRINT@ 323, "PRESS ENTER FOR NEXT SP
IN"
450 INPUT A$
460 PRINT@ 32, "
"
470 PRINT@ 192, "
"
480 PRINT@ 224, "
"
490 SOUND 175,1
500 FOR G=1 TO 500
510 NEXT G
520 GOTO 130
530 PRINT@ 40, "THREE OF A KIND"
540 PRINT@ 201, "YOU WIN $5"
550 M=M+5
560 GOTO 390
570 PRINT@ 202, "ONE OF EACH"
580 PRINT@ 235, "WIN $5"
590 M=M+ 5
600 GOTO 390
610 FORJ=1TO5
620 ON E GOSUB 700,710,720
630 X=X+K
640 PRINT@(98+J+5*D),CHR$(X)
650 PRINT@(130+J+5*D),CHR$(X)
660 PRINT@(162+J+5*D),CHR$(X)
670 X=X-K
680 NEXTJ
690 RETURN
700 K=0:RETURN
710 K=48:RETURN
```

```
720 K=64:RETURN
730 FOR XX=1 TO 1000:NEXT
740 CLS RND(8)
750 FOR SS=10 TO 250 STEP 10
760 SOUND SS,2:NEXT
770 FOR SS=240 TO 10 STEP -10
780 SOUND SS,2:NEXT
790 PRINT@ 192, "$$$ YOU HAVE BROKEN THE
 BANK $$$"
800 C=0
810 FOR SS=200 TO 250 STEP 10
820 SOUND SS,1:NEXT
830 FOR SS=240 TO 190 STEP -10
840 SOUND SS,1:NEXT
850 C=C+1
860 IF C=6 THEN 880
870 GOTO 810
880 CLS RND(8)
890 X=RND(30)+15
900 Y=RND(15)+8
910 C=RND(8)
920 SET(X,Y,C)
930 SOUND 225,1
940 GOTO 890
950 CLS RND(8)
960 FOR T=1 TO 15
970 FOR SS=10 TO 1 STEP -2
980 SOUND SS,1
990 FOR DD=1 TO T*2:NEXT
1000 NEXT:NEXT
1010 FOR DD=1 TO 20:NEXT
1020 CC=0
1030 SOUND 5,1
```

```
1040 SOUND 1,3
1050 FOR DD=1 TO 25:NEXT
1060 PRINT@ 192, TAB(7) "################
#"
1070 SOUND 5,3
1080 FOR DD=1 TO 25:NEXT
1090 CC=CC+1
1100 PRINT@ 192, TAB(7) "### BANKRUPT ##
#"
1110 IF CC=10 THEN END
1120 GOTO 1040
1130 FOR J=1 TO 16
1140 PRINT TAB(4) "$$$ ONE ARMED BANDIT
$$$"
1150 SOUND 75,2
1160 NEXT J
1170 FOR DD=1 TO 200:NEXT
1180 RETURN
```

TWENTY-ONE

This program plays a dice version of the card game Twenty-one.

The object of the game is to roll the dice until the total is either 21, or as close to it as you dare to go. Rolling more than 21 causes you to lose.

After you've had your turn at rolling the dice, the computer takes its turn. You'll find the computer plays this game quite well and will win more than fifty percent of the game, unless you are *very* careful.

When the program is run the computer will ask you to enter "1 to roll, 2 to stand, 3 to end". Enter "1" to roll the dice as many tiimes as you wish. When your total is as close to 21 as you're prepared to go, enter "2". This gives the computer its turn. When you are tired of playing enter "3" to end the game.

The GOSUB in line 20 runs the expanding circle at the start of the game. Line 60 gives you your choice of moves and waits for your input. Line 100 increments H (set at zero in line 40) with your dice roll. Line 120 prints the result on the screen.

If you go *over* 21, line 130 ends the program to the routine beginning at line 390. This tells you that you've gone bust.

After you've had your turn at rolling the dice, the computer takes its turn. You'll find the computer plays this game quite well and will win more than fifty percent of the games, unless you are *very* careful.

```
10 REM TWENTY ONE
20 GOSUB 660
30 PRINT
40 FOR CC=1 TO 8:FOR DD=1 TO 10
50 CLS CC
60 NEXT DD:SOUND 200,1:NEXT CC
70 CLS(4)
80 PRINT@ 192, TAB(4) "###*### TWENTY ONE
 ###*###"
90 FOR DD=1 TO 500:NEXT
100 CLS
110 H=0
120 C=0
130 PRINT@ 32, "1 TO ROLL, 2 TO STAND, 3
 TO END"
140 INPUT A:IF A=3 THEN 590
150 PRINT@ 64,"
160 IF A=2 THEN 240
170 H=H+RND(5)+1
180 SOUND 100,1
190 GOSUB 490
200 PRINT@ 133,"YOUR SCORE ";H
210 IF H>21 THEN GOSUB 520:GOTO 240
220 GOTO 140
230 GOSUB 490
240 IF C>H AND C<22 OR C>21 OR H>21 OR H
=21 AND C=21 GOTO 310
250 C=C+RND(5)+1
```

```
260 GOSUB 490
270 SOUND 100,1
280 PRINT@ 197,"MY SCORE";C
290 GOSUB 490
300 GOTO 240
310 GOSUB 490
320 IF H=C OR H>21 AND C>21 THEN 440
330 IF(C>H OR H>21) AND C<22 THEN CLS RN
D(8):PRINT@ 192,TAB(12) "I"
340 IF (C<H OR C>21) AND H<22 THEN GOSUB
 520:CLSRND(8):PRINT@ 192, TAB(11) "YOU"
350 PRINT@ 224, TAB(11) "WIN"
360 FOR CC=1 TO 1000:NEXT CC
370 FOR SS=10 TO 200 STEP 10
380 SOUND SS,1:NEXT SS
390 FOR SS=190 TO 10 STEP -10
400 SOUND SS,1:NEXT SS
410 GOSUB 490
420 CLS
430 GOTO 30
440 PRINT "DEAD HEAT"
450 FOR SS=10 TO 200 STEP 10
460 SOUND SS,1
470 NEXT SS
480 FOR DD=1 TO 200:GOTO 100
490 FOR E=1 TO 300
500 NEXT E
510 RETURN
520 FOR CC=1 TO 750:NEXT:CLS
530 FOR DD=1 TO 5
540 SOUND 75,1
550 FOR TT=1 TO 50
560 NEXT TT
```

```
570 NEXT DD
580 PRINT@ 206, "BUST":RETURN
590 PCLS
600 PMODE 1,1
610 SCREEN 1,1
620 FOR RR=190 TO 10 STEP -10
630 CIRCLE (128,92),RR:SOUND 200-RR,1
640 NEXT RR
650 FOR CC=1 TO 1000:NEXT CC:CLS:END
660 PCLS
670 PMODE 1,1
680 SCREEN 1,1
690 FOR RR=10 TO 90 STEP 5
700 CIRCLE (128,92),RR:SOUND 100+RR,1
710 NEXT RR
720 FOR CC=1 TO 20:SOUND 190,1:NEXT CC
730 CLS
740 RETURN
```

TEN PIN BOWLING

This program converts your computer into a Ten Pin Bowling alley. After a display of graphics the computer sets up the pins for you. The computer shows you the bowling alley as viewed from above.

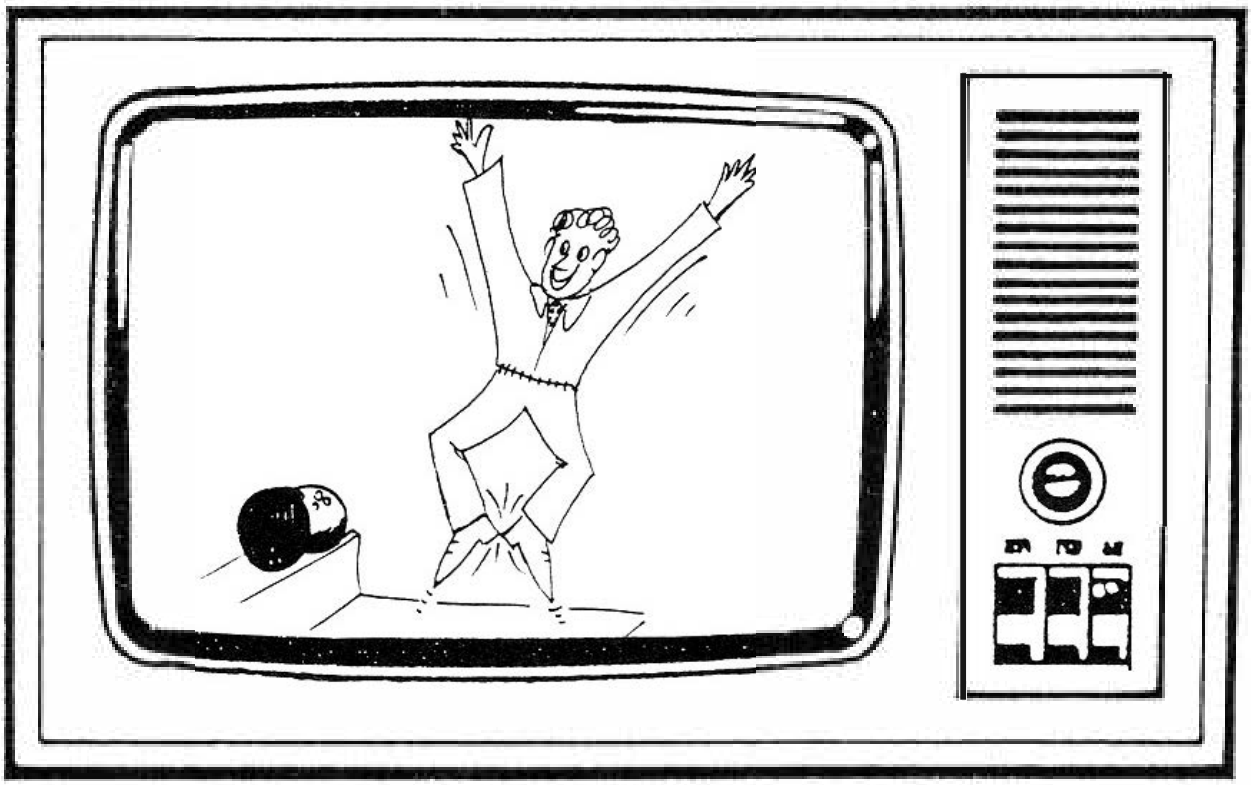
You will then be able to play a standard game of ten frames with two balls to each frame, except for the times when you're good enough to get a strike with the first ball. The computer will keep track of your score, tell you which frame and ball you are currently playing and reset the pins for you at the end of each frame. The game has a high score feature to add extra interest to the game.

You'll get a bonus score if all pins are knocked down with the two balls in the frame. A strike is scored if all the pins are knocked down with the first ball of the frame.

After the computer has set up the pins it will ask you to "PRESS ENTER TO BOWL". Pressing ENTER will then cause the square bowling ball to trundle up the screen into the pins. Lines 20 to 280 run the graphics at the start of the game. The GOSUB in line 290 sends the program to the title sequence in lines 940 to 1040. Line 310 is the start of the actual game.

Line 390 tells you that the computer is ready to bowl and waits for your input. After you have pressed ENTER, the GOSUB in line 410 goes to the routine in lines 860 to 930 which prints the ball on the screen (CHR$(128)) and makes it roll up the screen. Lines 770 to 830 set up all the pins at the beginning of each frame using the PRINT@ command.

Lines 510 to 560 print the pins after the ball has been bowled. This shows the results of your efforts with a combination of 0's and –'s. Line 590 advises you of a bonus and line 720 tells you if you scored a strike.



```
10 REM TEN PIN BOWLING
20 PCLS
30 PMODE 4,1
40 SCREEN 1,1
50 FOR T=5 TO 250 STEP 5
60 LINE (128,92)-(T,2),PSET
70 NEXT T
80 FOR W=2 TO 190 STEP 5
90 LINE (128,92)-(250,W),PSET
100 NEXT W
```

```
110 FOR S=250 TO 5 STEP -5
120 LINE (128,92)-(S,190),PSET
130 NEXT S
140 FOR V=190 TO 2 STEP -5
150 LINE (128,92)-(5,V),PSET
160 NEXT V
170 FOR T=5 TO 250 STEP 5
180 LINE (128,92)-(T,2),PRESET
190 NEXT T
200 FOR W=2 TO 190 STEP 5
210 LINE (128,92)-(250,W),PRESET
220 NEXT W
230 FOR S=250 TO 5 STEP -5
240 LINE (128,92)-(S,190),PRESET
250 NEXT S
260 FOR V=190 TO 2 STEP -5
270 LINE (128,92)-(5,V),PRESET
280 NEXT V
290 GOSUB 940
300 CLS
310 E=2
320 DIM A(10)
330 Y=0
340 S=0
350 FOR B=1 TO 10
360 GOSUB 770
370 FOR E=1 TO 2
380 IF E=1 THEN PRINT@ 466, " "
390 PRINT@ 384, "FRAME:";B;"PRESS ENTER
TO BOWL";
400 INPUT A$
410 GOSUB 860
420 PRINT@ 393, "
```

```
430 PRINT@ 7, "FRAME:";B;"BALL:";E
440 Z=0
450 FOR C=1 TO 10
460 IF E=2 GOTO 480
470 A(C)=79
480 IF RND(10)<7 THEN A(C)=45
490 IF A(C)=45 THEN Z=Z+1
500 NEXT C
510 PRINT@ 76,CHR$(A(10));" ";CHR$(A(9))
;" ";CHR$(A(8));" ";CHR$(A(7))
520 FOR T=1 TO 100:NEXT T
530 PRINT@ 109, CHR$(A(6));" ";CHR$(A(5)
);" ";CHR$(A(4))
540 PRINT@ 142, CHR$(A(3));" ";CHR$(A(2)
)
550 FOR T=1 TO 100:NEXT T
560 PRINT@ 175, CHR$(A(1))
570 PRINT@ 448, "SCORE THIS FRAME:";Z
580 IF Z=10 THEN Z=15
590 IF Z=15 THEN PRINT@ 202, "### BONUS
###":FOR T=1 TO 500:NEXT T:PRINT@ 202, "
                                "

600 IF E=2 THEN S=S+Z
610 IF E=2 AND B<>10 THEN PRINT@ 40, "SC
ORE SO FAR:";S
620 FOR T=1 TO 1000:NEXT
630 NEXT E
640 NEXT B
650 PRINT@ 40, "SCORE SO FAR:";S
660 PRINT@ 416, "SCORE FOR THAT GAME WAS
:";S
670 IF S<Y THEN 690
680 Y=S
```

```
690 PRINT@ 416, "HIGHEST SCORE SO FAR:";
Y
700 FOR T=1 TO 500:NEXT T
710 GOTO 340
720 PRINT@ 234, "*** STRIKE ***"
730 FOR T=1 TO 500:NEXT T
740 S=S+15
750 E=2
760 GOTO 580
770 FOR D=1 TO 10
780 A(D)=79
790 NEXT D
800 PRINT@ 76,CHR$(A(10));" ";CHR$(A(9))
;" ";CHR$(A(8));" ";CHR$(A(7))
810 PRINT@ 109,CHR$(A(6));" ";CHR$(A(5))
;" ";CHR$(A(4))
820 PRINT@ 142,CHR$(A(3));" ";CHR$(A(2))
830 PRINT@ 175,CHR$(A(1))
840 FOR T=1 TO 2000:NEXT T
850 RETURN
860 FOR JJ=367 TO 207 STEP -32
870 X$=CHR$(128)
880 PRINT@ JJ,X$
890 FOR VV=1 TO 10:NEXT
900 SOUND 50,1
910 PRINT@ JJ, " "
920 NEXT JJ
930 RETURN
940 FOR CC=2 TO 8
950 CLS CC
```

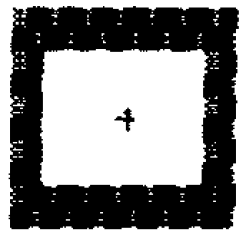
```
960 PRINT@ 192, TAB(4) "*** TEN PIN BOWL
ING ***"
970 FOR SS=100 TO 250 STEP 10
980 SOUND SS,1
990 NEXT SS
1000 FOR SS=240 TO 100 STEP -10
1010 SOUND SS,1
1020 NEXT SS
1030 NEXT CC
1040 RETURN
```

ROULETTE

The following two programs give you the opportunity to try one of the traditional Casino games without the danger of losing your life savings. The first program simulates the European game of roulette which uses a wheel numbered from one to thirty-six plus zero. The second program caters for the American system which uses one to thirty-six, zero and double zero.

Roulette uses a table divided into squares, marked with the numbers on the wheel and are used for placing bets. The numbers are in three rows of twelve. Some are red numbers and some are black. The red numbers are 1, 3, 5, 7, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30, 32, 34 and 36. The black numbers are 2, 4, 6, 8, 10, 11, 13, 15, 17, 20, 22, 24, 26, 28, 29, 31, 33 and 35.

```
THE BALL HAS STOPPED ON 4
```




```
YOU HAVE LOST
```

When you run the program, the computer will tell you how many chips you have to play with. It will then ask you to place your bet, followed by asking you how many chips you wish to gamble. The wheel is then spun. When the wheel stops, the

computer will tell you if you have won or lost. The game is over when you lose all your chips and creep away in disgrace. However, if you are clever enough to *win* a fortune of 1000 chips (plus your original 100 chips) you'll have broken the bank.

You are able to bet on a wide variety of numbers and combinations of numbers. The key to placing your bets is:-

A – any single number (1 to 36)
B – two adjoining numbers
C – three numbers in a row
D – six numbers in adjoining rows
E – four numbers in a square
F – first twelve numbers (1 to 12)
G – second twelve numbers (13 to 24)
H – third twelve numbers (25 to 36)
I – numbers 1 to 18
J – numbers 19 to 36
K – twelve numbers in a horizontal row
L – two adjacent columns
M – any red number
N – any black number
O – any even number
P – any odd number
Q – to quit the game

The computer will not allow you to bet more chips than you have. You may enter Q to quit the game at any time. The two programs are very similar, with the second one having a few extra lines to provide the double zero.

A quick look at the first program shows that the GOSUB in line 30 sets up the graphic routine at the beginning of the game. Variable K in line 40 provides a delay which is called from several different points in the game. Lines 150 to 300 process your input and send the computer to the relevant

sub-routine. These routines (lines 510 to 710) work out the required numbers and the payout odds if you win.

Lines 1120 to 1160 print the checkerboard pattern around the number displayed on the screen when the wheel is spun. Lines 340 to 350 provide the spinning effect of the wheel. The sound is provided by the GOSUB in line 340. A lengthening loop is used to give the effect of the wheel slowing down.

```
10 REM EUROPEAN ROULETTE
20 Z$=CHR$(185):W$="
30 CLS:GOSUB 1170: GOTO 50
40 FOR K=1 TO 1000:NEXT:RETURN
50 DIM B(24):CH=100
60 CLS:PRINT@ 39, "YOU HAVE";CH;"CHIPS"
70 GOSUB 40
80 PRINT@ 102, "LADIES AND GENTLEMEN":PR
INT@ 132, " PLEASE PLACE YOUR BETS"
90 INPUT A$:IF A$="" THEN 90
100 A=ASC(A$)-64:IF A<1 OR A>17 THEN 90
110 IF A=17 THEN 740
120 FOR Q=1 TO 24:B(Q)=-99:NEXT
130 INPUT "HOW MANY CHIPS";N:IF N>CH THE
N 130
140 CH=CH-N
150 IF A=1 THEN GOSUB 510
160 IF A=2 THEN GOSUB 520
170 IF A=3 THEN GOSUB 530
180 IF A=4 THEN GOSUB 550
190 IF A=5 THEN GOSUB 560
200 IF A=6 THEN GOSUB 580
210 IF A=7 THEN GOSUB 590
220 IF A=8 THEN GOSUB 600
230 IF A=9 THEN GOSUB 610
240 IF A=10 THEN GOSUB 620
```

```
250 IF A=11 THEN GOSUB 630
260 IF A=12 THEN GOSUB 650
270 IF A=13 THEN GOSUB 670
280 IF A=14 THEN GOSUB 690
290 IF A=15 THEN GOSUB 700
300 IF A=16 THEN GOSUB 710
310 GOSUB 40
320 CLS RND(8):PRINT@ 192,TAB(1) "#*# TH
E WHEEL IS SPINNING #*#":GOSUB 40
330 CLS
340 GOSUB 1120:FOR B=1 TO 50:C=RND(37)-1
:PRINT@ 237,C:PRINT@ 241,Z$:FOR T=1 TO 3
*B
350 NEXT:GOSUB 910:NEXT
360 GOSUB 40:PRINT@ 34, "THE BALL HAS ST
OPPED ON";C
370 Y=0:E=1
380 IF B(E)=C THEN Y=1:GOTO 400
390 IF E<24 THEN E=E+1:GOTO 380
400 IF Y=0 THEN 470
410 WI=OD*N:CH=CH+WI+N
420 PRINT@ 352, "CONGRATULATIONS YOU HAV
E WON";WI;" CHIPS"
430 PRINT "PLUS YOUR BET OF";N;"CHIPS"
440 GOSUB 40
450 GOSUB 480
460 GOTO 60
470 PRINT@360, "YOU HAVE LOST":GOTO 440
480 IF CH<1 THEN GOTO 810
490 IF CH>1100 GOTO 930
500 RETURN
510 INPUT "WHICH NUMBER";D:B(1)=D:OD=35:
RETURN
```

```
520 INPUT "WHICH TWO NUMBERS";B(1),B(2):
OD=17:RETURN
530 PRINT "WHICH NUMBER":INPUT "   IN LE
FT COLUMN";D
540 FOR E=0 TO 2:B(E+1)=D+E:NEXT:OD=11:R
ETURN
550 INPUT "FIRST NUMBER OF SIX";D:FOR E=
0 TO 5:B(E+1)=D+E:NEXT:OD=5:RETURN
560 INPUT "FIRST NUMBER IN SQUARE";D:FOR
 E=0 TO 3:B(E+1)=D+E:IF E=2 THEN D=D+1
570 NEXT:OD=8:RETURN
580 FOR E=1 TO 12:B(E)=E:NEXT:OD=2:RETUR
N
590 FOR E=1 TO 12:B(E)=E+12:NEXT:OD=2:RE
TURN
600 FOR E=1 TO 12:B(E)=E+24:NEXT:OD=2:RE
TURN
610 FOR E=1 TO 18:B(E)=E:NEXT:OD=1:RETUR
N
620 FOR E=1 TO 18:B(E)=E+18:NEXT:OD=1:RE
TURN
630 PRINT "LOW NUMBER AT END":INPUT "OF
LINE";D
640 FOR E=0 TO 11:B(E+1)=3*E+D:NEXT:OD=2
:RETURN
650 INPUT "LOW NUMBER IST COLUMN";D1:INP
UT "LOW NUMBER 2ND COLUMN";D2:IF ABS(D1-
D2)>1 THEN 650
660 FOR E=0 TO 11:B(E+1)=3*E+D1:B(E+13)=
3*E+D2:NEXT:OD=.5:RETURN
670 RESTORE
680 FOR E=1 TO 18:READ B(E):NEXT:OD=1:RE
TURN
```

```
690 RESTORE:FOR E=1 TO 18:READ Z:NEXT:FO
R E=1 TO 18:READ B(E):NEXT:OD=1:RETURN
700 FOR E=2 TO 36 STEP 2:B(E/2)=E:NEXT:O
D=1:RETURN
710 FOR E=1 TO 35 STEP 2:B((E+1)/2)=E:NE
XT:OD=1:RETURN
720 DATA 1,3,5,7,9,12,14,16,18,19,21,23,
25,27,30,32,34,36
730 DATA 2,4,6,8,10,11,13,15,17,20,22,24
,26,28,29,31,33,35
740 CLS RND(8):PRINT@ 192,TAB(3) "YOU AR
E WITHDRAWING FROM THE"
750 PRINT@ 224, TAB(7) "GAME WITH";CH;"C
HIPS"
760 FOR SS=1 TO 240 STEP 20
770 SOUND SS,1:NEXT
780 FOR SS=220 TO 1 STEP -20
790 SOUND SS,1:NEXT
800 END
810 FOR JJ=1 TO 15
820 PRINT TAB(8) "YOU HAVE LOST"
830 NEXT JJ
840 FOR DD=1 TO 100:NEXT
850 CLS RND(8)
860 PRINT@ 192, TAB(5) "YOU HAVE RUN OUT
 OF CHIPS"
870 FOR SS=100 TO 10 STEP -5
880 SOUND SS,1
890 NEXT
900 END
910 SOUND 150,1
920 RETURN
930 PCLS
```

```
940 PMODE 1,1
950 SCREEN 1,1
960 FOR XX=1 TO 256 STEP 30
970 FOR YY=1 TO 192 STEP 30
980 CIRCLE(XX,YY),30,7
990 NEXT:NEXT
1000 FOR DD=1 TO 500:NEXT
1010 FOR TT=1 TO 5
1020 FOR SS=100 TO 250 STEP 10
1030 SOUND SS,1
1040 NEXT SS
1050 FOR SS=240 TO 100 STEP -10
1060 SOUND SS,1
1070 NEXT SS
1080 NEXT TT
1090 CLS RND(8)
1100 PRINT@ 192, "** YOU HAVE BROKEN THE
 BANK **"
1110 END
1120 PRINT@ 171, Z$;Z$;Z$;Z$;Z$;Z$;Z$
1130 PRINT@ 203, Z$;W$;Z$
1140 PRINT@ 235, Z$;W$;Z$
1150 PRINT@ 267, Z$;W$;Z$
1160 PRINT@ 299, Z$;Z$;Z$;Z$;Z$;Z$;Z$:RE
TURN
1170 PCLS
1180 PMODE 1,1:SCREEN 1,1
1190 LINE(1,1)-(252,188),PSET,B
1200 LINE(20,20)-(232,158),PSET,B
1210 LINE(40,40)-(212,138),PSET,B
1220 LINE(60,60)-(192,118),PSET,B
1230 LINE(80,80)-(172,98),PSET,B
1240 PAINT(11,11),6,8
```

```
1250 PAINT(45,45),7,8
1260 PAINT(82,82),8,8
1270 FOR DD=1 TO 500:NEXT
1280 CLS RND(8)
1290 PRINT@ 192, "**********************
**********":FOR DD=1 TO 20:NEXT
1300 SOUND 225,1
1310 PRINT@ 192, "**********  ROULETTE
**********":FOR DD=1 TO 30:NEXT:SOUND 20
0,1:CC=CC+1
1320 IF CC=20 THEN RETURN
1330 GOTO 1290
```

```
10 REM AMERICAN ROULETTE
20 Z$=CHR$(185):W$="      "
30 CLS:GOSUB 1180: GOTO 50
40 FOR K=1 TO 1000:NEXT:RETURN
50 DIM B(24):CH=100
60 CLS:PRINT@ 39, "YOU HAVE";CH;"CHIPS"
70 GOSUB 40
80 PRINT@ 102, "LADIES AND GENTLEMEN":PR
INT@ 132, " PLEASE PLACE YOUR BETS"
90 INPUT A$:IF A$="" THEN 90
100 A=ASC(A$)-64:IF A<1 OR A>17 THEN 90
110 IF A=17 THEN 750
120 FOR Q=1 TO 24:B(Q)=-99:NEXT
130 INPUT "HOW MANY CHIPS";N:IF N>CH THE
N 130
140 CH=CH-N
```

```
150 IF A=1 THEN GOSUB 520
160 IF A=2 THEN GOSUB 530
170 IF A=3 THEN GOSUB 540
180 IF A=4 THEN GOSUB 560
190 IF A=5 THEN GOSUB 570
200 IF A=6 THEN GOSUB 590
210 IF A=7 THEN GOSUB 600
220 IF A=8 THEN GOSUB 610
230 IF A=9 THEN GOSUB 620
240 IF A=10 THEN GOSUB 630
250 IF A=11 THEN GOSUB 640
260 IF A=12 THEN GOSUB 660
270 IF A=13 THEN GOSUB 680
280 IF A=14 THEN GOSUB 700
290 IF A=15 THEN GOSUB 710
300 IF A=16 THEN GOSUB 720
310 GOSUB 40
320 CLS RND(8):PRINT@ 192,TAB(1) "### TH
E WHEEL IS SPINNING ###":GOSUB 40
330 CLS
340 GOSUB 1130:FOR B=1 TO 50:C=RND(38)-2
:IF C=-1 THEN PRINT@ 238, "00" ELSE PRIN
T@ 237,C
350 PRINT@ 241,Z$:FOR T=1 TO 3*B
360 NEXT:GOSUB 920:NEXT
370 GOSUB 40:IF C=-1 THEN PRINT@ 34, "TH
E BALL HAS STOPPED ON"," DOUBLE ZERO" EL
SE PRINT@ 34, "THE BALL HAS STOPPED ON";
C
380 Y=0:E=1
390 IF B(E)=C THEN Y=1:GOTO 410
400 IF E<24 THEN E=E+1:GOTO 390
410 IF Y=0 THEN 480
```

```
420 WI=OD*N:CH=CH+WI+N
430 PRINT@ 352, "CONGRATULATIONS YOU HAV
E WON";WI;" CHIPS"
440 PRINT "PLUS YOUR BET OF";N;"CHIPS"
450 GOSUB 40
460 GOSUB 490
470 GOTO 60
480 PRINT@360, "YOU HAVE LOST":GOTO 450
490 IF CH<1 THEN GOTO 820
500 IF CH>1100 GOTO 940
510 RETURN
520 INPUT "WHICH NUMBER";D:B(1)=D:OD=35:
RETURN
530 INPUT "WHICH TWO NUMBERS";B(1),B(2):
OD=17:RETURN
540 PRINT "WHICH NUMBER":INPUT "    IN LE
FT COLUMN";D
550 FOR E=0 TO 2:B(E+1)=D+E:NEXT:OD=11:R
ETURN
560 INPUT "FIRST NUMBER OF SIX";D:FOR E=
0 TO 5:B(E+1)=D+E:NEXT:OD=5:RETURN
570 INPUT "FIRST NUMBER IN SQUARE";D:FOR
 E=0 TO 3:B(E+1)=D+E:IF E=2 THEN D=D+1
580 NEXT:OD=8:RETURN
590 FOR E=1 TO 12:B(E)=E:NEXT:OD=2:RETUR
N
600 FOR E=1 TO 12:B(E)=E+12:NEXT:OD=2:RE
TURN
610 FOR E=1 TO 12:B(E)=E+24:NEXT:OD=2:RE
TURN
620 FOR E=1 TO 18:B(E)=E:NEXT:OD=1:RETUR
N
```

```
630 FOR E=1 TO 18:B(E)=E+18:NEXT:OD=1:RE
TURN
640 PRINT "LOW NUMBER AT END":INPUT "OF
LINE";D
650 FOR E=0 TO 11:B(E+1)=3*E+D:NEXT:OD=2
:RETURN
660 INPUT "LOW NUMBER IST COLUMN";D1:INP
UT "LOW NUMBER 2ND COLUMN";D2:IF ABS(D1-
D2)>1 THEN 660
670 FOR E=0 TO 11:B(E+1)=3*E+D1:B(E+13)=
3*E+D2:NEXT:OD=.5:RETURN
680 RESTORE
690 FOR E=1 TO 18:READ B(E):NEXT:OD=1:RE
TURN
700 RESTORE:FOR E=1 TO 18:READ Z:NEXT:FO
R E=1 TO 18:READ B(E):NEXT:OD=1:RETURN
710 FOR E=2 TO 36 STEP 2:B(E/2)=E:NEXT:O
D=1:RETURN
720 FOR E=1 TO 35 STEP 2:B((E+1)/2)=E:NE
XT:OD=1:RETURN
730 DATA 1,3,5,7,9,12,14,16,18,19,21,23,
25,27,30,32,34,36
740 DATA 2,4,6,8,10,11,13,15,17,20,22,24
,26,28,29,31,33,35
750 CLS RND(8):PRINT@ 192,TAB(3) "YOU AR
E WITHDRAWING FROM THE"
760 PRINT@ 224, TAB(7) "GAME WITH";CH;"C
HIPS"
770 FOR SS=1 TO 240 STEP 20
780 SOUND SS,1:NEXT
790 FOR SS=220 TO 1 STEP -20
800 SOUND SS,1:NEXT
810 END
```

```
820 FOR JJ=1 TO 15
830 PRINT TAB(8) "YOU HAVE LOST"
840 NEXT JJ
850 FOR DD=1 TO 100:NEXT
860 CLS RND(8)
870 PRINT@ 192, TAB(5) "YOU HAVE RUN OUT
 OF CHIPS"
880 FOR SS=100 TO 10 STEP -5
890 SOUND SS,1
900 NEXT
910 END
920 SOUND 150,1
930 RETURN
940 PCLS
950 PMODE 1,1
960 SCREEN 1,1
970 FOR XX=1 TO 256 STEP 30
980 FOR YY=1 TO 192 STEP 30
990 CIRCLE(XX,YY),30,7
1000 NEXT:NEXT
1010 FOR DD=1 TO 500:NEXT
1020 FOR TT=1 TO 5
1030 FOR SS=100 TO 250 STEP 10
1040 SOUND SS,1
1050 NEXT SS
1060 FOR SS=240 TO 100 STEP -10
1070 SOUND SS,1
1080 NEXT SS
1090 NEXT TT
1100 CLS RND(8)
1110 PRINT@ 192, "** YOU HAVE BROKEN THE
 BANK **"
1120 END
```

```
1130 PRINT@ 171, Z$;Z$;Z$;Z$;Z$;Z$;Z$
1140 PRINT@ 203, Z$;W$;Z$
1150 PRINT@ 235, Z$;W$;Z$
1160 PRINT@ 267, Z$;W$;Z$
1170 PRINT@ 299, Z$;Z$;Z$;Z$;Z$;Z$;Z$:RE
TURN
1180 PCLS
1190 PMODE 1,1:SCREEN 1,1
1200 LINE(1,1)-(252,188),PSET,B
1210 LINE(20,20)-(232,158),PSET,B
1220 LINE(40,40)-(212,138),PSET,B
1230 LINE(60,60)-(192,118),PSET,B
1240 LINE(80,80)-(172,98),PSET,B
1250 PAINT(11,11),6,8
1260 PAINT(45,45),7,8
1270 PAINT(82,82),8,8
1280 FOR DD=1 TO 500:NEXT
1290 CLS RND(8)
1300 PRINT@ 192, "**********;*****;*******
**********":FOR DD=1 TO 20:NEXT
1310 SOUND 225,1
1320 PRINT@ 192, "**********  ROULETTE
**********":FOR DD=1 TO 30:NEXT:SOUND 20
0,1:CC=CC+1
1330 IF CC=20 THEN RETURN
1340 GOTO 1300
```

TEE OFF

Tee Off is a fully computerised, nine-hole golf course. The computer will tell you the par for each hole, the number of strokes taken for each hole, your running total of strokes and at the end of the game your total score and hole average.

After displaying the par for the hole and the hole number, the computer will ask "STROKE ?". To hit the ball enter a

```
PAR FOR THIS HOLE>> 5
### HOLE NUMBER 2 ###


                    O
---------------------- -------

AFTER THAT STROKE YOUR SCORE IS
 6
STROKE ?
```

number. The higher the number, the harder you hit the ball. The computer works out, in line 120, the distance the ball has travelled. Variable A is the number you used to hit the ball. From this, the computer determines the value of variable J, the position where the ball lands. This is displayed on the screen by lines 360 to 400 if the ball hasn't landed in the cup, and lines 420 to 470 if it has.

The cup is at position 24 and the computer decides if J = 24 in line 160. Variable SC is your score for each hole, and variable C the running total and final score. Your average for each hole is calculated in line 520.

Lines 570 onward have nothing to do with the game proper,

but provide the simple cartoon if you decide to stop playing. These lines may be deleted or altered to provide your own ending.

```
10 REM TEE-OFF
20 C=0
30 FOR Z=1 TO 9
40 J=RND(12)-1
50 SC=0
60 Q=RND(4)+2
70 CLS
80 GOSUB 300
90 PRINT"STROKE ";:INPUT A
100 CLS
110 IF J>24 THEN A=-A
120 J=J+INT(A/RND(Q))
130 GOSUB 300
140 SC=SC+1
150 PRINT"AFTER THAT STROKE YOUR SCORE I
S ";SC
160 IF J<>24 THEN 90
170 GOSUB 420
180 C=C+SC
190 PRINT"SCORE FOR ";Z;" HOLES IS ";C
200 FOR M=1 TO 1000:NEXT M
210 FOR L=1 TO 20
220 FOR Y=1 TO L
230 PRINT " ";
240 NEXT Y
250 FOR T=1 TO 40:NEXT
260 PRINT"STAND BY"
270 NEXT L
280 NEXT Z
```

```
290 GOTO 490
300 IF J>30 THEN J=30
310 SOUND 150,1:SOUND 150,1
320 PRINT "PAR FOR THIS HOLE>>";Q
330 PRINT "### HOLE NUMBER";Z;"###"
340 PRINT
350 PRINT
360 FOR K=1 TO J-1
370 PRINT" ";
380 NEXT K
390 PRINT"0"
400 PRINT "---------------------- -----
---"
410 RETURN
420 PRINT "----------------------0-----
-----"
430 FOR L=1 TO 10
440 PRINT"YOU DID IT IN ";SC;" STROKES"
450 FOR T= 1TO50
460 NEXT
470 NEXT L
480 RETURN
490 PRINT"END OF THAT ROUND"
500 PRINT
510 PRINT"YOU SCORED ";C
520 PRINT"YOUR AVERAGE WAS ";(C/9)
530 PRINT
540 PRINT"DO YOU WANT ANOTHER ROUND? (YE
S/NO)"
550 INPUT T$
560 IF T$<>"NO" THEN RUN
570 GOTO 580
580 CLS(6)
```

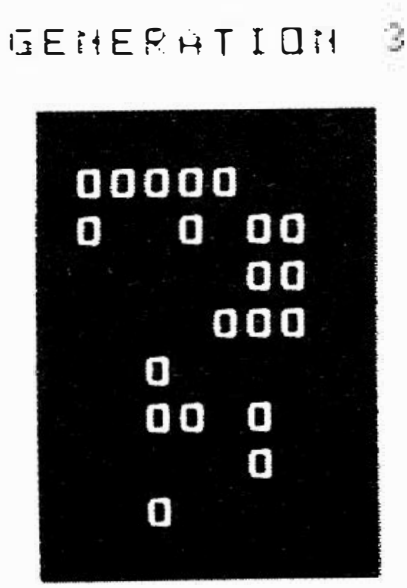
```
590 FOR H= 0 TO 63
600 FOR V=22 TO 31
610 SET(H,V,1)
620 NEXT V,H
630 FOR H=42 TO 44
640 FOR V=23 TO 24
650 SET (H,V,6)
660 NEXT V,H
670 FOR H=4 TO 5
680  V=21
690 SET(H,V,5)
700 NEXT H
710 IF F>55 THEN F=55:GOTO790
720  V=21
730 FOR H=4 TO 5
740 SET(1+F+H,V,6)
750 SET(3+F+H,V,5)
760 NEXT H
770 F=F+2
780 GOTO 710
790 SOUND 50,1
800 PRINT@ 448, "WHOOPS!!!"
810 GOTO 810
```

SIMULATIONS:
COLONY (LIFE)
ACME ZITHER COMPANY

COLONY

This program simulates the life cycle of an isolated colony of single celled organisms. Although the cells are simple creatures they have evolved a highly organised society governed by a strict set of laws. These laws control the birth, growth and death of the cells from one generation to the next.

It is possible for a cell in the interior of the colony to have up to

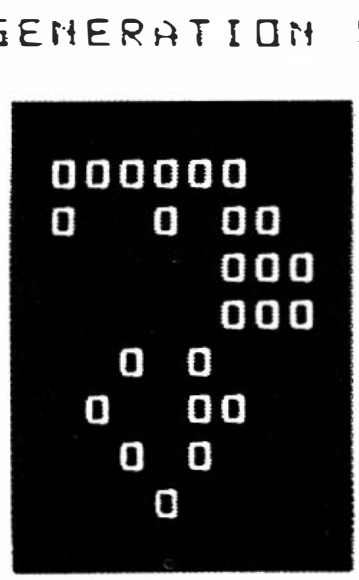


eight neighbours. A cell in an outside wall can have five neighbours and a corner cell three neighbours.

The laws state that a cell with two or three neighbours will survive to the next generation. A new cell will be born into an empty space if that space has three cells adjoining it. Any cell with four or more neighbours will die from over-population due to the competition for food and living space. A cell with no neighbours will die of loneliness.

The program begins by using DIM to create a ten by ten array. Although the colony occupies a nine by nine grid, a ten by ten space is required. The computer checks each cell for neighbours and it will show an error if it checks an outside wall or corner cell and doesn't find an empty space on the other side of it.

Lines 60 and 70 provide the grid locations. Lines 80 to 110 randomly decide if a grid location is occupied by a cell or if it is an empty space. Lines 310 to 390 print the grid onto the screen using an inverse letter O if the grid position is occupied by a cell, or a blank space (CHR$(128)) if the location is empty.



Lines 170 to 280 apply the laws to each grid position and adjust the value of M for the next generation. Line 130 increments the number of the generation and line 300 provides the sound effects. The idea for the program comes from John Conway's *LIFE.*

```
10 REM COLONY
20 G=0
30 CLS
40 DIM M(10,10)
50 DIM N(10,10)
60 FOR K=2 TO 9
70 FOR Z=2 TO 9
80 IF RND(100)<45 THEN M(K,Z)=1
90 N(K,Z)=M(K,Z)
100 NEXT Z
110 NEXT K
120 GOSUB 300
130 G=G+1
140 FOR K=2 TO 9
150 FOR Z=2 TO 9
160 C=0
170 IF M(K-1,Z-1)=1 THEN C=C+1
180 IF M(K-1,Z)=1 THEN C=C+1
190 IF M(K-1,Z+1)=1 THEN C=C+1
200 IF M(K,Z-1)=1 THEN C=C+1
210 IF M(K,Z+1)=1 THEN C=C+1
220 IF M(K+1,Z-1)=1 THEN C=C+1
230 IF M(K+1,Z)=1 THEN C=C+1
240 IF M(K+1,Z+1)=1 THEN C=C+1
250 IF M(K,Z)=1 AND C<>3 AND C<>2 THEN N
(K,Z)=0
260 IF M(K,Z)=0 AND C=3 THEN N(K,Z)=1
270 NEXT Z
280 NEXT K
290 GOTO 120
300  SOUND 100,1:SOUND 200,1:SOUND 100,1
310 PRINT @73,"GENERATION";G:PRINT
320 FOR K=1 TO 10
```

```
330 PRINT TAB(10);
340 FOR Z=1 TO 10
350 M(K,Z)=N(K,Z)
360 IF M(K,Z)=1 THEN PRINT "o"; ELSE PRI
NT CHR$(128);
370 NEXT Z
380 PRINT
390 NEXT K
400 RETURN
```

Here is a graphic illustration of the development of one colony over 10 generations.

```
GENERATION 0
OO OO
 O    O
  O  O O
 O     O
  O  OOO
OOO  O O
 OOO
  O
```

```
GENERATION 1
OOO
OO OOOO
 OO    O
 OO  O O
O O  O O
O   OO O
O  O
 OOO
```

```
GENERATION 2
O OOOO
   OOOO
       O
O  O   O
O OO O O
O  OOO
O  O
 OOO
```

```
GENERATION 3
  O   O
  O   O
   O O O
 OOOO  O
O O  O
O    OO
O
 OOO
```

GENERATION 4

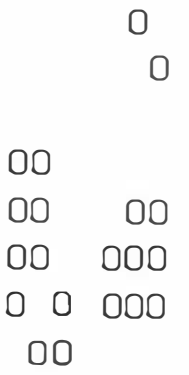
GENERATION 5

GENERATION 6

GENERATION 7

GENERATION 8

GENERATION 9



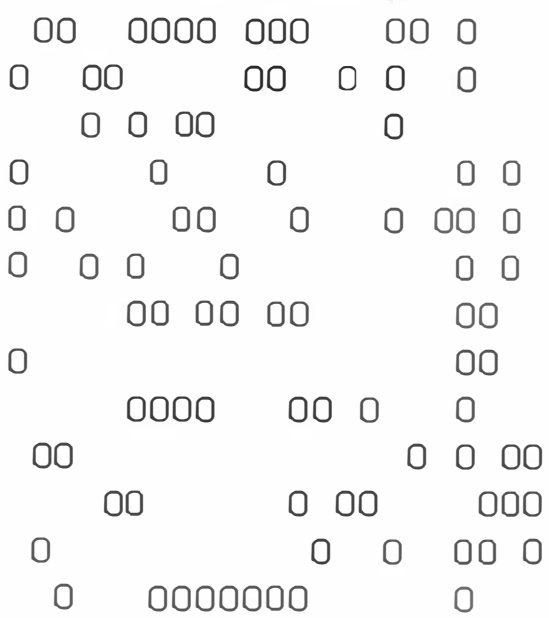
The following program listing is for you if you have access to a printer. The grid size has been enlarged to print out a bigger colony. This was done by changing the size of the arrays in lines 40 and 50. The values of K and Z also need to be changed.

This program is very slow but does make a nice demonstration program for a printer. You may speed up the execution of the program by making the arrays smaller. After the program listing, you'll see a sample printout of this version of the program.

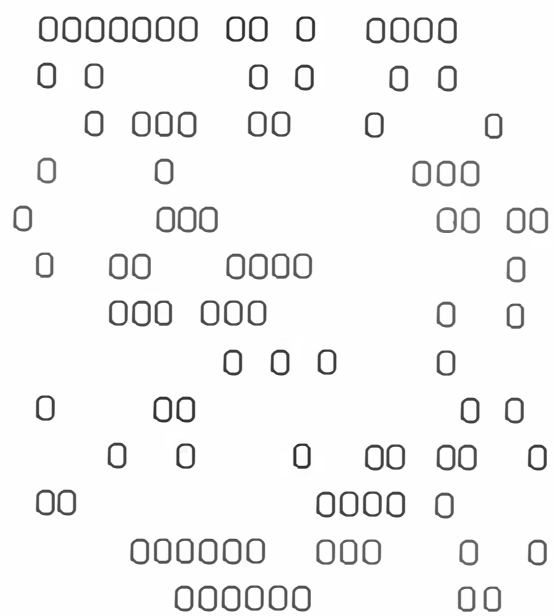
```
10 REM COLONY FOR PRINTER
20 G=0
30 CLS
40 DIM M(15,25)
50 DIM N(15,25)
60 FOR K=2 TO 14
70 FOR Z=2 TO 24
80 IF RND(100)<45 THEN M(K,Z)=1
90 N(K,Z)=M(K,Z)
100 NEXT Z
110 NEXT K
120 GOSUB 300
130 G=G+1
140 FOR K=2 TO 14
150 FOR Z=2 TO 24
160 C=0
170 IF M(K-1,Z-1)=1 THEN C=C+1
180 IF M(K-1,Z)=1 THEN C=C+1
190 IF M(K-1,Z+1)=1 THEN C=C+1
200 IF M(K,Z-1)=1 THEN C=C+1
210 IF M(K,Z+1)=1 THEN C=C+1
```

```
220 IF M(K+1,Z-1)=1 THEN C=C+1
230 IF M(K+1,Z)=1 THEN C=C+1
240 IF M(K+1,Z+1)=1 THEN C=C+1
250 IF M(K,Z)=1 AND C<>3 AND C<>2 THEN N
(K,Z)=0
260 IF M(K,Z)=0 AND C=3 THEN N(K,Z)=1
270 NEXT Z
280 NEXT K
290 GOTO 120
300  SOUND 100,1:SOUND 200,1:SOUND 100,1
310 PRINT #-2," "
320 PRINT#-2,"      GENERATION";G
330 FOR K=2 TO 14
340 PRINT TAB(10);
350 FOR Z=2 TO 24
360 M(K,Z)=N(K,Z)
370 IF M(K,Z)=1 THEN PRINT#-2, "O"; ELSE
 PRINT#-2," ";
380 NEXT Z
390 PRINT #-2," "
400 NEXT K
410 RETURN
```

GENERATION 1



GENERATION 2



ACME ZITHER COMPANY

Your computer has the ability to set up and manage a large number of variables, and the uses of this are only listed by your imagination. The 'Acme Zither Company' demonstrates just how effectively your computer can handle simulation information, manipulating it according to ground rules laid down by the programmer.

This program allows you to experience the excitement and despair of running a factory in a harsh economic environment.

The computer will provide you with a weekly balance sheet. This will contain all the information you require, such as the current week of trading, capital on hand, stock on hand, the selling price of your product, the current cost of production and the strength of your work force and its production potential.

You must keep a close watch on the amount of capital you have on hand as compared to your production costs and wages bill. This may be the key to your success or failure as a factory manager. The computer will give you the opportunity to hire or fire staff. However you may find that the workers have a strong union if you try to fire too many of them.

You will be able to set production targets (within the limits of your available capital and the ability of your workers). Then you must wait for your sales figures.

At various times you will have to cope with union demands for increased wages, suppliers putting up the price of raw materials and the occasional disaster. There will be opportunities to raise the price of your zithers but you must be careful. Do

not price them right out of the market place. The object of the game is to avoid the shame of bankruptcy and hopefully, to make a million dollars. The program listing is quite long but playing it will reward the patience required to key it in.

Acme Zither Company is based on a series of subroutines. GOSUB 1500 sets up the values of all the variables in the program. These include the number of staff, the cost of production, the initial selling price of zithers and number of zithers each worker can make.

GOSUB 950 provides the printout of the factory's status. This is called up at several points in the game to keep you informed of any changes. GOSUB 1220 controls the hiring and firing of staff, plus any union reaction to firing staff. GOSUB 1100 controls the manufacture of your product. It allows you to set a production target and tells you the production results.

GOSUB 790 is your sales team. It provides the good news or the bad news about the current weeks sales. The sales figures can include zithers from your warehouse as well as the current week's production. Any unsold zithers are automatically placed in storage.

GOSUB 360 controls the "unpredictables". This includes events such as union demands for pay increases, rising production costs and the occasional fire.

GOSUB 170 uses the commands CIRCLE and PAINT to provide a colorful opening to the game. Line 200 gives a variable value to P. P is then used as the X axis co-ordinate for the centre of the circle. The spiral effect is obtained by incrementing P.

```
10 REM ACME ZITHER COMPANY
20 CLS
30 GOSUB 170
40 GOSUB 1500
50 WE=WE+1
60 GOSUB 950
70 GOSUB 1220
80 CLS
90 GOSUB 950
100 GOSUB 1100
110 GOSUB 950
120 GOSUB 790
130 GOSUB 360
140 CA=CA-INT(WG*WF)
150 GOTO 50
160 ST=100+RND(500)
170 PCLS
180 PMODE 3,1
190 SCREEN 1,0
200 FOR P=10 TO 250 STEP 10
210 Y=92
220 CIRCLE (P,Y),20
230 NEXT P
240 PAINT (0,0),4,4
250 PAINT (255,191),4,4
260 PAINT (0,0),3,1
270 GOSUB 1620
280 CLS(3):PRINT@ 160, " ":PRINT@ 192, T
AB(3) "*** ACME ZITHER COMPANY ***"
290 PRINT@ 224, " "
300 W=0
310 S=RND(50)+200:SOUND S,1:W=W+1
```

```
320 IF W=10 GOTO 340
330 GOTO 310
340 GOSUB 1620
350 RETURN
360 REM UNPREDICTABLES
370 CLS
380 IF RND(100)<45 THEN 460
390 A=RND(10)
400 SOUND 50,3:CLS RND(8):PRINT@ 192, "U
NIONS DEMAND A";A"% PAY INCREASE"
410 WG=INT(100*(WG+(A*WG/100)))/100
420 GOSUB 1620
430 PRINT "PAY PER EMPLOYEE IS NOW $";WG
440 GOSUB 1620
450 CLS
460 IF RND(100)<85 GOTO 560
470 SOUND 75,3:CLS RND(8):PRINT@ 192, "W
AREHOUSE FIRE DESTROYS SOME OF YOUR STOC
K...."
480 PRINT@ 256, TAB(3) "STANDBY FOR DAMA
GE REPORT:"
490 GOSUB 1620
500 A=INT(RND(0)*ST/2)+1
510 ST=ST-A
520 CLS RND(8):PRINT@ 192, "TOTAL OF STO
CK DESTROYED WAS";A;" ZITHERS, WORTH $";
A*SP;"RETAIL"
530 GOSUB 1620
540 PRINT "STOCK ON HAND IS NOW ";ST
550 GOSUB 1620
560 IF RND(100)>30 GOTO 670
570 SOUND 100,3:CLS RND(8):PRINT@ 160, T
AB(3) "SUPPLIER ANNOUNCES DRAMATIC"
```

```
580 PRINT@ 192, TAB(6) "PRICE RISE !!!"
590 GOSUB 1620
600 A=INT (RND(0)*100*CT/7)/100
610 IF A<.01 GOTO 700
620 PRINT@ 224, "THE COST OF MAKING ZITH
ERS HAS  GONE UP BY $";A;"EACH"
630 FOR Y=1 TO 500:NEXT
640 CT=CT+A
650 PRINT "IT NOW COSTS $";CT;"TO MAKE E
ACH ONE"
660 GOSUB 1620
670 IF RND(100)<65 AND MP<SP THEN RETURN
680 CLS RND(8)
690 SOUND 175,2:SOUND 150,2:PRINT@ 192,
"YOU HAVE AN OPPORTUNITY TO RAISE YOUR P
RICE"
700 PRINT@ 256, TAB(4) "ZITHERS NOW SELL
 FOR $";SP
710 GOSUB 1620
720 PRINT@ 256, TAB(1) "STATE THE PERCEN
TAGE OF YOUR  PRICE INCREASE";:INPUT A
730 IFA>0 THEN Z=Z+A
740 SP=INT(100*(SP+A*SP/100))/100
750 GOSUB1620
760 CLS RND(8):PRINT@ 192, "ZITHERS NOW
SELL FOR $";SP
770 GOSUB 1630
780 RETURN
790 REM SALE
800 PRINT "STOCK ON HAND IS ";ST
810 GOSUB 1620
820 CLS RND(8)
```

```
830 SOUND 120,2:FOR TT=1 TO 10:NEXT TT:S
OUND 100,2:PRINT@ 192, TAB(2) "STAND BY
FOR SALES REPORT..."
840 GOSUB 1620
850 CLS RND(8):FOR Y=1 TO 500:NEXT
860 A=INT (RND(0)*ST/(2/100))+1
870 IF A>ST GOTO 860
880 SOUND 200,2:SOUND 200,2
890 PRINT@ 192, TAB(2) "TOTAL ZITHERS SO
LD...";A
900 ST=ST-A
910 PRINT@ 224, TAB(2) "INCOME FROM SALE
S: $";A*SP
920 CA=CA+INT(A*SP)
930 GOSUB 1620
940 RETURN
950 REM PRINT OUT
960 CLS
970 IF CA+ST<1 GOTO 1370
980 IF CA+ST>999999 THEN 1640
990 SOUND 150,2:PRINT"FACTORY REPORT: WE
EK";WE
1000 PRINT "CAPITAL ON HAND IS $";CA
1010 PRINT "YOUR WAREHOUSE HOLDS";ST;"ZI
THERS..WORTH $";ST*SP
1020 PRINT "THEY SELL FOR $";SP;"EACH"
1030 PRINT "AND COST $";CT;"EACH TO MAKE
"
1040 PRINT "WORKFORCE IS";WF;"PEOPLE"
1050 PRINT "THEIR WAGES ARE $";WG;"EACH"
1060 PRINT "THIS WEEKS WAGE BILL IS $";W
G*WF
```

```
1070 PRINT "EACH PERSON CAN MAKE";PD;"ZI
THERS A WEEK"
1080 PRINT "A TOTAL OUTPUT OF";PD*WF;"ZI
THERS"
1090 RETURN
1100 INPUT "HOW MANY ZITHERS DO YOU WISH
 TO MAKE";MK
1110 IF MK=0 THEN RETURN
1120 IF MK*CT>CA THEN PRINT "NOT ENOUGH
MONEY TO MAKE THAT MANY":GOTO 1100
1130 IF MK>PD*WF THEN PRINT "NOT ENOUGH
EMPLOYEES TO MAKE THAT MANY":GOTO 1100
1140 CLSRND(8):PRINT@ 192, "TARGET WEEK"
;WE;"IS";MK"ZITHERS"
1150 MK=ABS(MK-((RND(2)-1)*RND(10)))
1160 GOSUB 1620
1170 PRINT "TOTAL MADE IN WEEK";WE;"WAS"
;MK
1180 ST=ST+MK
1190 CA=INT(CA-CT*MK)
1200 GOSUB1620
1210 RETURN
1220 REM PEOPLE
1230 INPUT "HOW MANY PEOPLE DO YOU WANT
TO  HIRE";A
1240 WF=WF+A
1250 CLSRND(8):PRINT@ 192,TAB(6) "TOTAL
WORKFORCE IS";WF
1260 GOSUB1630
1270 IF A<>0 GOTO 1350
1280 PRINT@ 224, TAB(1) "HOW MANY PEOPLE
 DO YOU WANT TO  FIRE";:INPUT A
```

```
1290 IF A=0 GOTO 1350
1300 IF A>WF THEN GOTO 1280
1310 A=INT(RND(0)*A+1)
1320 GOSUB 1630
1330 PRINT "UNIONS WILL ONLY ALLOW YOU T
O   FIRE";A
1340 WF=WF-A
1350 GOSUB 1630
1360 RETURN
1370 REM BANKRUPTCY
1380 L=0
1390 CLSRND(8):PRINT@ 192,TAB(7)"### BAN
KRUPT ###"
1400 SOUND 75,3:FOR C=1 TO 200:NEXT C
1410 CLS RND(8):PRINT@ 192, TAB(7) "####
############":FOR YY=1 TO 100:NEXT YY:L=
L+1:IF L=10 THEN 1430
1420 GOTO 1390
1430 CLS 0:GOSUB 1620
1440 CLS RND(8):PRINT@ 192, "YOU KEPT TH
E ACME EITHER COMPANY RUNNING FOR";WE;"W
EEKS":FOR FF=1 TO 4000:NEXT FF
1450 CLS RND(8):PRINT@ 192, "ENTER Y FOR
 ANOTHER TRY OR N TO END THE ORDEAL"
1460 A$=INKEY$
1470 IF A$="" GOTO 1460
1480 IF A$="Y" THEN RUN
1490 END
1500 REM VARIABLES
1510 CA=500+RND(500)
1520 ST=100+RND(500)
1530 SP=10+RND(5)
1540 CT=2+RND(5)
```

```
1550 IF CT>SP GOTO 1530
1560 WF=7+RND(10)
1570 WG=12+INT(RND(0)*(SP*5))
1580 PD=5+RND(5)
1590 WK=1
1600 Z=1
1610 RETURN
1620 FOR Y=1 TO 2000:NEXT:RETURN
1630 FOR Y=1 TO 1000:NEXT:RETURN
1640 C=0
1650 CLS RND(8):PRINT@ 192, "#*#*# YOUVE
 MADE A MILLION #*#*#":FOR T=1 TO 200:NE
XT T
1660 C=C+1
1670 SS=0
1680 S=RND(50)+200:SOUND S,1:SS=SS+1
1690 IF SS=10 GOTO 1710
1700 GOTO 1680
1710 PRINT@ 192, "#*#*#*#*#*#*#*#*#*#*#
#*#*#*#*#*":FOR YY=1 TO 100:NEXT YY:IF C
=10 GOTO 1450
1720 GOTO 1650
```

SPACE GAMES:

STAR SEARCH
ASTEROID
VENUS PROBE
SPACE RESCUE
MARTIAN MODULE

STAR SEARCH

You and your trusty computer are on star patrol in the vast regions of space. Alien beings are penetrating your section of the galaxy and it is your job to find them and stop them.

You have a limited amount of energy available and your computer will keep you informed of your energy level. Your space ship comes complete with long range and short range scanners to help you search. Short range scanners will tell you if alien ships are nearby but they won't tell you in what direction. The long range scanner will give you a positive or negative reading for a square two squares away from you in the specified direction. Both types of scanners use up precious energy. Long range scanning uses more energy than short range scanning.

When you locate an alien ship it will appear on the screen as an X. When you have destroyed it, it will turn into a blue square. It is possible to fly your ship through a blue square. Any attempt to move your ship through a square occupied by an alien ship will result in a collision. Your ship appears on the screen as an asterisk.

Your computer will give you a list of instructions telling you how to move your ship, use your scanners and fire. Your short range scanners can't be used when you are in the outside rows of the grid, and trying to do so will cause an FC error.

Lines 1190 to 1280 decide which area of the galaxy you are in. Lines 1100 to 1180 print the galactic grid onto the screen. A$ is your input direction. Lines 410, 420, 480 and 490 use this to modify your location in the galaxy. Lines 200 to 370 are the lines which operate your scanners. Variable E is the amount of energy available. It is set in line 1330. Line 70 tests E to see if you have energy as the program runs through each turn.

```
10 REM STAR SEARCH
20 GOSUB 1360
30 CLS
40 GOSUB 1290
50 GOSUB 1100
60 PRINT@ 384, "REMAINING ENERGY";INT(E)
70 IF E<1 THEN 780
80 IF AL>0 THEN PRINT "TALLY";AL
90 PRINT@ 416, "1-SCAN 2-MOVE 3-FIRE";
100 INPUT D
110 IF D=1 THEN GOSUB 150
120 IF D=2 THEN GOSUB 360
130 IF D=3 THEN GOSUB 570
140 GOTO 60
150 GOSUB1080
160 PRINT@448, "SCANNER:";
170 PRINT@416, " SHORT(1) OR LONG(2)";
180 INPUT K
190 E=E-10*K:F=0
200 IF K=2 THEN 250
210 IF A(B+1,C)=1 OR A(B+1,C+1)=1 OR A(B
,C+1)=1 OR A(B-1,C)=1 OR A(B-1,C-1)=1 TH
EN F=1:GOTO 230
220 IF A(B,C-1)=1 OR A(B+1,C-1)=1 OR A(B
-1,C+1)=1 THEN F=1
230 IF F=0 THEN PRINT "SCAN NEGATIVE":GO
TO 350
240 IF F=1 THEN PRINT Z$;" NEAR":GOTO 35
0
250 GOSUB1080
260 PRINT@416, "DIRECTION: N-1,S-2,E-3,W
-4";
```

```
270 INPUT N:Z=0
280 IF N=1 AND A(B-2,C)=1 THEN Z=1
290 IF N=2 AN■ A(B+2,C)=1 THEN Z=1
300 IF N=3 AN■ A(B,C+2)=1 THEN Z=1
310 IF N=4 AN■ A(B,C-2)=1 THEN Z=1
320 PRINT@416, "SCANNER IS:";
330 IF Z=1 THEN PRINT "POSITIVE"
340 IF Z=0 THEN PRINT "NEGATIVE"
350 FOR T=1 TO 2000:NEXT:RETURN
360 E=E-50:A(B,C)=0:B(B,C)=0
370 A$="":B$=""
380 GOSUB1080
390 PRINT@416, "DIRECTION (N/S)";
400 INPUT A$
410 IF A$="N" THEN B=B-1
420 IF A$="S" THEN B=B+1
430 PRINT "LOCATION:";B;" ";C
440 IFA$="N"ORA$="S"THEN510
450 GOSUB1080
460 PRINT@ 416, "DIRECTION (E/W)";
470 INPUT B$
480 IF B$="E" THEN C=C+1
490 IF B$="W" THEN C=C-1
500 PRINT "LOCATION:";B;" ";C
510 FOR I=1 TO 1000:NEXT
520 IF B(B,C)=1 THEN 920
530 IF A(B,C)=1 THEN 920
540 A(B,C)=2:B(B,C)=2
550 GOSUB 1100
560 RETURN
570 G=B:A$=""
580 PRINT "DIRECTION OF FIRE N,S,E,OR W"
;
590 INPUT A$:IF A$="N" THEN G=G-1
```

```
600 IF A$="S" THEN G=G+1
610 F=C
620 IFA$="N"ORA$="S"THEN660
630 IF A$="E" THEN F=F+1
640 IF A$="W" THEN F=F-1
650 IFRND(10)>4GOTO680
660 E=E-100
670 IF A(G,F)<>1 THEN 700
680 PRINT@352, "YOU HIT THE ";Z$
690 AL=AL+1:A(G,F)=4:B(G,F)=4:GOTO 770
700 B(G,F)=3:PRINT@352, "YOU MISSED"
710 FOR G=1 TO 1000:NEXT
720 PRINT@384, "THE ";Z$;" ARE SHOOTING
BACK"
730 FOR G=1 TO 1000:NEXT
740 IF RND(10)>5 THEN 760
750 PRINT@416, "THE ENEMY HAS HIT US!!":
E=E-100*RND(0):GOTO 770
760 PRINT@416, "THE ";Z$;" MISSED US!!"
770 FOR I=1 TO 1000:NEXT:GOSUB 1100:RETU
RN
780 CLS:FOR DD=1 TO 100:NEXT:PRINT@352,
"ENERGY BANKS EXHAUSTED"
790 PRINT@384, "YOU KILLED";AL;:IF AL>1
THEN PRINT "ALIENS" ELSE PRINT "ALIEN"
800 GOSUB1080
810 PRINT@416, "ON THIS MISION YOUR COMM
AND     RATING IS";1000*(AL/8):FOR DD=1
TO 1500:NEXT
820 FOR T=10 TO 100 STEP 5
830 SOUND T,1
840 FOR DD=1 TO T/10
850 NEXT:NEXT
```

```
860 FOR NN=1 TO 8
870 CLS NN
880 FOR DD=1 TO 25:NEXT DD
890 SOUND 200,1
900 NEXT NN
910 CLS 0:GOTO 910
920 FOR DD=1 TO 500:CLS:PRINT@ 192, "YOU
R SHIP HAS COLLIDED WITH A     ";Z$;" SH
IP"
930 FOR XX=1 TO 1500:NEXT
940 PRINT TAB(5) "YOUR SHIP HAS EXPLODED
"
950 FOR DD=1 TO 500:NEXT
960 PCLS
970 PMODE 4,1
980 SCREEN 1,1
990 FOR RR=1 TO 150 STEP 10
1000 CIRCLE (128,92),RR
1010 NEXT RR
1020 FOR SS=1 TO 100 STEP 10
1030 SOUND SS,1:NEXT
1040 FOR SS=90 TO 1 STEP -10
1050 SOUND SS,1:NEXT
1060 FOR DD=1 TO 1000:NEXT
1070 END
1080 PRINT@416,"

1090 RETURN
1100 CLS
1110 FOR Q=1 TO 10
1120 PRINTTAB(6)
1130 FOR P=1 TO 10
1140 IF B(Q,P)=0 THEN PRINT CHR$(046);"
";
1150 IF B(Q,P)=2 THEN PRINT "*";" ";
1160 IF B(Q,P)=3 THEN PRINT "X";" ";
```

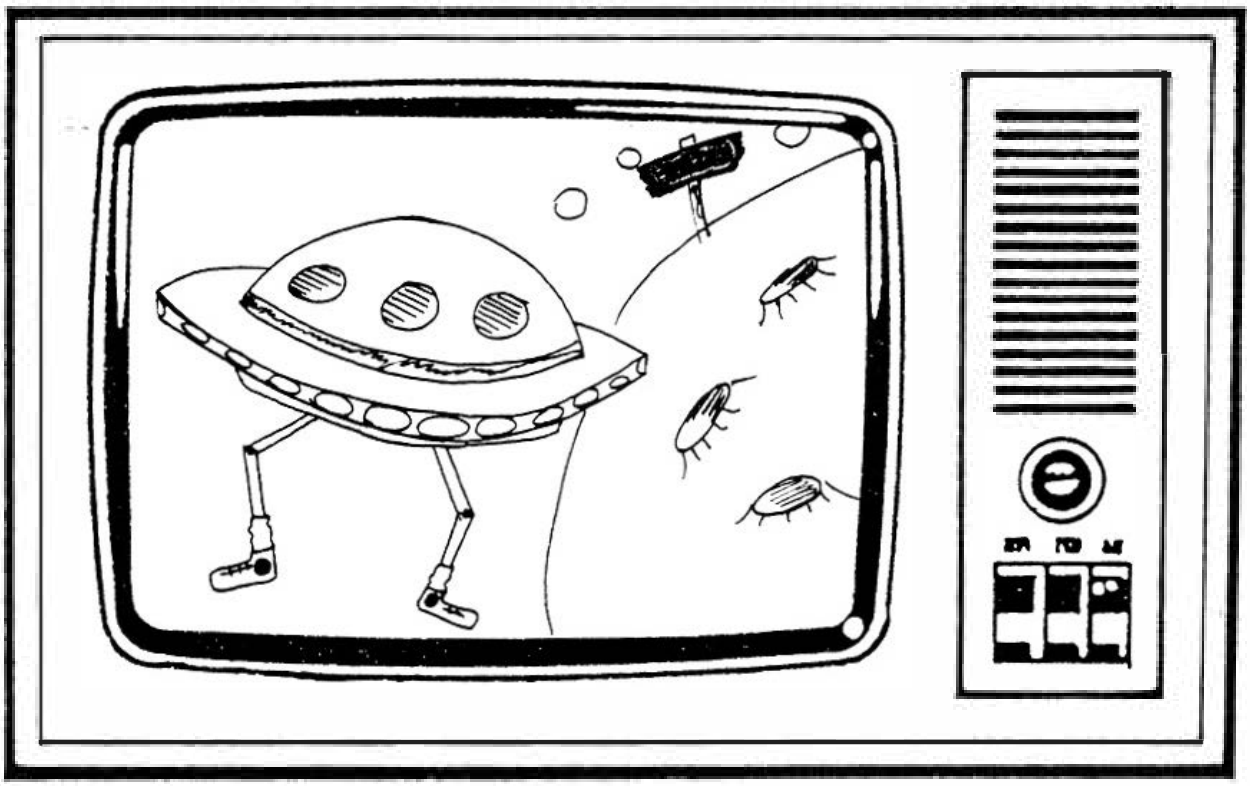
```
1170 IF B(Q,P)=4 THEN PRINT CHR$(175);"
";
1180 NEXT:PRINT:NEXT
1190 Q=8*C:PRINT@ 352, "POSITION:";
1200 IF Q<10 THEN PRINT " RIGEL";
1210 IF Q>9 AND Q<20 THEN PRINT " SIRIUS
";
1220 IF Q>19 AND Q<30 THEN PRINT " ALTAI
R";
1230 IF Q>29 AND Q<50 THEN PRINT " CANOP
US";
1240 IF Q>49 AND Q<70 THEN PRINT " VEGA"
;
1250 IF Q>69 AND Q<90 THEN PRINT " PROCY
ON"
1260 IF Q>89 THEN PRINT " ALDERBARAN";
1270 PRINT " SECTOR";B;C
1280 RETURN
1290 DIM A(10,10),B(10,10)
1300 FOR A=1 TO 15:X=RND(10):Y=RND(10):A
(X,Y)=1:NEXT
1310 B=5:C=5:A(2,C)=2:B(B,C)=2:AL=0
1320 Z$="RUSNORAN"
1330 E=1234
1340 SOUND 175,1
1350 RETURN
1360 CLS RND(8)
1370 FOR I=2 TO 10
1380 PRINT@ 192, "********** STAR  SEARCH
 **********"
1390 SOUND I*25,1
```

```
1400 FOR DD=1 TO 100/T
1410 NEXT
1420 PRINT@ 192,"**********************
**********
1430 FOR DD=1 TO 50:NEXT
1440 NEXT T
1450 RETURN
```

ASTEROID

You are the administrator of a mining colony which has been established on an asteroid deep in the asteroid belt. Your isolated colony survives by mining valuable copper ore, now very rare on earth, and trading it for food, oxygen and essential equipment.

You must manage the affairs of the colony so that it produces



enough ore to survive, but you must not waste your scarce resources at the same time.

You will be given a computer printout telling you all the information you need, including the state of your food stocks, amount of oxygen on hand, and cost of maintaining the

colony per year and a rare raid by asteroid bandits. By mining the optimum amount of ore you should be able to buy sufficient food and oxygen for your colony to survive.

Apart from the attacks by asteroid bandits, the game isn't based on random factors or chance. It will involve a great deal of management skill to survive for any length of time. Each time the game is run the price of copper ore, food units, oxygen units and the annual maintenance cost will be different.

```
10 REM ASTEROID
20 GOSUB 1520
30 CLS
40 X=9
50 GOSUB 780
60 YR=YR+1
70 FK=FK+FK/(2+RND(18))-FK/(3+RND(15))
80 GOTO 380
90 GOSUB 1490
100 FOR QQ=200 TO 203+RND(4):SOUND QQ,1:
NEXT QQ
110 PRINT@6,CHR$(181);"computers report"
;CHR$(186)
120 IF OX<OD*FK GOTO 1230
130 IF FD<FE*FK THEN 1260
140 IF CA<1 THEN 1290
150 IF FK<2 THEN 1320
160 IF FK<13 THEN PRINT "WARNING - POPUL
ATION IS","NEARING EXTINCTION"
170 IF OX<2*OD*FK THEN PRINT "WARNING -
OXYGEN SUPPLY IS LOW"
180 IF FD<2*FE*FK THEN PRINT "WARNING -
FOOD STOCKS LOW"
```

```
190 IF CASH<2000 THEN PRINT "WARNING - M
ONEY RUNNING LOW"
200 PRINT"THERE ARE";INT(FK);"PEOPLE ON
";
210 PRINT "THE       ASTEROID IN YEAR";YR
220 PRINT
230 GOSUB 1490
240 PRINT"MONEY CREDIT IS $";INT(CA)
250 GOSUB 1490
260 PRINT
270 PRINT"ANNUAL MAINTENANCE   $";RP
280 GOSUB 1490
290 PRINT
300 PRINT"OXYGEN TANKS HOLD";INT(OX);"UN
ITS"
310 PRINT"OXYGEN COSTS $";OC;"PER UNIT"
320 PRINT"OXYGEN NEED PER PERSON:";OD
330 GOSUB 1490
340 PRINT
350 PRINT"FOOD STOCKS STAND AT";FD
360 GOSUB 1490
370 RETURN
380 GOSUB 90
390 PRINT:PRINT
400 PRINT"COPPER ORE - HOW MANY TONS WIL
L YOU MINE AND SELL"
410 PRINT "THEY USE UP";AC;"UNITS OF"
420 PRINT "OXYGEN AND SELL FOR $";AP
430 INPUT B
440 PRINT
450 IF B*AC>OX THEN PRINT "NOT ENOUGH OX
YGEN"
460 IF B*AC>OX THEN 430
```

```
470 CA=CA+B*AC
480 CLS
490 GOSUB 90
500 PRINT:PRINT
510 PRINT "FOOD COSTS $";FC;"PER UNIT"
520 PRINT "EACH PERSON NEEDS";FE;"FOOD U
NITS"
530 PRINT "$";INT(FC*FE);"EACH, $";INT(F
K*FC*FE);"FOR STATION"
540 PRINT "THIS WILL LAST";INT(FD/(FE*FK
));"YEARS AT THE    PRESENT POPULATION"
550 PRINT:PRINT
560 PRINT "HOW MANY FOOD UNITS WILL YOU
BUY?"
570 INPUT C
580 PRINT
590 IF C*FC>CA THEN PRINT "NOT ENOUGH MO
NEY"
600 IF C*FC>CA THEN 570
610 FD=FD+C*FC
620 CA=CA-C*FC
630 CLS
640 GOSUB 90
650 PRINT "HOW MUCH OXYGEN WILL YOU BUY?
"
660 PRINT "CURRENT STOCKS WILL LAST FOR"
;INT(OX/(OD*FK));" YEARS AT THE PRESENT
POPULATION"
670 INPUT D
680 PRINT
690 IF D*OC>CA THEN PRINT"NOT ENOUGH MON
EY"
700 IF D*OC>CA THEN 670
```

```
710 CLS
720 IF RND(5)=2 GOSUB 920
730 FD=FD-FD*FE
740 CA=CA-RP-D*OC
750 OX=OX+D-FK*OD
760 GOSUB 1490:GOSUB 1490
770 GOTO 60
780 YR=RND(5)
790 A$="the space colony has perished"
800 FK=80+RND(40)
810 CA=INT(7*(700+RND(800))/RND(3))
820 FC=RND(7)
830 AC=1+RND(3)
840 FD=2000+RND(500)
850 OX=2000-RND(1500)
860 OC=RND(7)
870 AP=30*RND(AC)
880 RP=200+RND(400)
890 FE=1+RND(5)
900 OD=2+RND(3)
910 RETURN
920 CLS
930 J=RND(6)
940 GOSUB 1490
950 PRINT "THE COLONY WAS ATTACKED BY"
960 IF J=1 THEN PRINT "A FLEET OF SYRIAN
 SHIPS"
970 IF J=2 THEN PRINT "RENEGADE EARTHLIN
GS"
980 IF J=3 THEN PRINT "MARTIAN SPACE PIL
OTS"
990 IF J=4 THEN PRINT "UYRILLIEX OUTWORL
DERS"
```

```
1000 IF J=5 THEN PRINT "A LONE SHIP, APP
ARENTLY UNDER",,"ROBOT CONTROL"
1010 IF J=6 THEN PRINT "A PARRALEXIAN ES
CORT VESSEL"
1020 PRINT
1030 GOSUB 1490
1040 PRINT
1050 PRINT
1060 Z=1+INT(FK/(RND(15)+1))
1070 PRINT "THERE WERE";Z;" people kille
d"
1080 X=250+RND(250)
1090 GOSUB 1490
1100 PRINT "DAMAGE WAS $";X
1110 Y=RND(300)
1120 W=RND(300)
1130 PRINT "AND FOOD STOCKS HAVE FALLEN
BY";W
1140 FD=FD-W:IF FD<0 THEN FD=FD+W
1150 FK=FK-Z
1160 OX=OX-Y
1170 CA=CA-X
1180 GOSUB 1490
1190 PRINT"PRESS enter"
1200 INPUT U$
1210 CLS
1220 RETURN
1230 PRINT A$
1240 PRINT "YOU RAN OUT OF OXYGEN IN YEA
R";YR
1250 GOTO 1450
1260 PRINT A$
```

```
1270 PRINT "FOOD SUPPLIES WERE EXHAUSTED
 IN YEAR";YR
1280 GOTO 1450
1290 PRINT A$
1300 PRINT "THE TREASURY RAN ORY DURING
YEAR";YR
1310 GOTO 1450
1320 PRINT "YOUR POPULATION HAS FALLEN"
1330 FK=RND(26)
1340 PRINT"TO";FK;". DO YOU WANT TO"
1350 CA=RND(300)
1360 PRINT "COMMIT SUICIDE PAINLESSLY
1370 PRINT "NOW (1) OR AWAIT A SAD AND"
1380 PRINT "LINGERING DEATH (2)?"
1390 INPUT B
1400 CLS
1410 IF B=1 THEN 1440
1420 PRINT "I HOPE YOU HAVE CHOSEN WELL"
1430 GOTO 60
1440 PRINT@ 204, "goodbye"
1450 FORDD=1 TO 2000:NEXT:CLS 0
1460 X=RND(63):Y=RND(31):C=RND(8)
1470 SET(X,Y,C):SOUND 225,1
1480 GOTO 1460
1490 FOR QQ=1 TO 300:NEXT QQ
1500 SS=RND(50)+200:SOUND SS,1
1510 RETURN
1520 CLS 0
1530 FOR JJ=1 TO 75
1540 X=RND(63):Y=RND(31):C=RND(8)
1550 SET(X,Y,C)
1560 NEXT JJ
1570 X=1:Y=RND(10)+10:C=5
```

```
1580 SET(X,Y,C)
1590 SET (X+1,Y,C)
1600 RESET(X,Y)
1610 SOUND 175,1
1620 X=X+1
1630 IF X=63 THEN 1650
1640 GOTO 1590
1650 CLS RND(8)
1660 PRINT@ 192, TAB(6) "**** ASTEROID *
***"
1670 FOR JJ=1 TO 6
1680 FOR SS=200 TO 250 STEP 10
1690 SOUND SS,1
1700 NEXT
1710 FOR SS=240 TO 190 STEP -10
1720 SOUND SS,1
1730 NEXT
1740 NEXT
1750 RETURN
```

VENUS PROBE

You are in command of an exploration team attempting to land on the planet Venus. You must pilot your landing craft through thick fog and erratic winds to a safe landing on the surface.

By using the graphic display of your radar screen and the information provided by your computer you must apply the correct amount of thrust to touch down on Venus. The final landing speed of your craft must be less than ten. When your descent gets below 100 meters the radar screen changes to give you a larger scale for the final approach.

A good landing is rewarded with a points score based on your final touch-down speed, and the amount of fuel you have left. A bad landing will result in a crater on the surface of Venus. The size of the crater is related to your final speed.

The loop from line 220 controls the printout of your ship on the screen. Your ship is printed lower and lower on the screen, as your velocity (V) and your height (H) change. Variable K controls the distance across the screen. As K is altered your ship drifts left or right. K is altered in line 310. Line 290 provides the two graphic characters that make up the image of your space ship.

Line 30 sets up your initial fuel supply (F), your rate of fall (V) and your starting height (H). Lines 340 and 350 provide the computer read out on your screen and ask for the amount of thrust you wish to use (T).

Lines 370 to 390 act on this input making alterations to speed, fuel and height. Lines 400 to 420 test these variables and decide if you are still in flight or if you have reached the ground.

The radar printout for your descent below one hundred meters is provided by lines 440 to 530. Lines 540 to 580 tell you if you have touched down safely, and lines 590 to 740 give the bad news if you haven't.

```
10 REM VENUS PROBE
20 CLS
30 K=2:F=300:V=15:H=500:Z=H
40 PMODE 1,1
50 PCLS
60 SCREEN 1,1
70 FOR RADIUS=1 TO 100 STEP 20
80 SOUND RADIUS,1
90 CIRCLE(128,96),RADIUS
100 NEXT RADIUS
110 PAINT (0,0),7,0
120 FOR C=250 TO 255 STEP .25:SOUND C,1:
NEXT
130 CLS RND(8):PRINT@ 192,TAB(6) "### VE
NUS PROBE ##"
140 FOR T= 1 TO 10
150 PRINT @192,TAB(6);"**********
**"
160 FOR Q=4*T TO 4*T+10 STEP 2:SOUND Q,1
:NEXT
170 PRINT @192,TAB(6);"### VENUS PROBE #
*#"
180 FOR Q=255 TO 230 STEP -T:SOUND Q,1:N
EXT:NEXT
190 CLS
200 IF H<99 THEN CLS:GOTO 440
210 IF V<1 THEN CLS
```

```
220 PRINT@ 0, " "
230 FOR J=2 TO (600-Z)/50
240 PRINT
250 NEXT J
260 FOR J=1 TO K
270 PRINT " ";
280 NEXT J
290 PRINT CHR$(137);CHR$(134)
300 PRINT@ 364,  "#----#"
310 K=K+RND(3)-RND(2)
320 SOUND 10*(K+1),1:SOUND 11*(K+1),1:SO
UND 10*(K+1),1   OMIT LINE 320.
330 Z=H
340 PRINT@ 384, "FUEL:";F,"HEIGHT:";H
350 PRINT "RATE OF FALL:";U;"  ";:INPUT"
THRUST";T
360 SOUND (I+1),1 OMIT LINE 360
370 U=U+5-T+RND(3)
380 H=H-U-RND(2)-1
390 F=F-ABS(T)
400 IF F<1 THEN 590
410 IF H>10 THEN 200
420 IF U>10 THEN 590
430 GOTO 540
440 HE=(100-H)/7:IF HE>12 THEN HE=3
450 FOR J=3 TO HE
460 PRINT
470 NEXT J
480 FOR J=1 TO K
490 PRINT " ";
500 NEXT J
510 PRINT CHR$(137);CHR$(134)
520 PRINT@ 369, "#----#"
```

```
530 GOTO 310
540 FOR C=1 TO 8:CLS(C):FOR G=1 TO200:NE
XT G:NEXT C
550 PRINT@ 192,TAB(4) "## SUCCESSFUL LAN
DING ##"
560 PLAY "L16;G;F;G;D;E;D;C;A;C;D;G;F"
570 PRINT @ 224,TAB(6) "# SCORING"(F-V)*
100+RND(147);"POINTS #"
580 END
590 CLS(0):FOR C=1 TO 100:NEXT:CLS(4):FO
R C=1 TO 100:NEXT:CLS(0):FOR C=1 TO 100:
NEXT:CLS(4)
600 FOR M=1 TO 4:CLS
610 PRINT@ 192, TAB(6) "### CRASH LANDIN
G ###"
620 SOUND 50,3
630 FOR P=1 TO 200:NEXT
640 CLS:FOR Q=1 TO 200:NEXT Q
650 NEXT M
660 PRINT@ 192, TAB(6) "### CRASH LANDIN
G ###"
670 PRINT@ 224,  "LEAVING A CRATER";ABS(
V*3);"METERS DEEP"
680 SOUND 50,2:SOUND 75,2:SOUND 50,2:SOU
ND 25,2:SOUND 50,4
690 IF F<2 THEN PRINT@ 264, "FUEL EXHAUS
TED"
700 FOR X=3 TO 1 STEP -.5
710 FOR T=230 TO 255 STEP X
720 SOUND T,1:NEXT:NEXT
730 FOR T=1 TO 10:SOUND RND(20)+235,1:NE
XT
740 GOTO 740
```

SPACE RESCUE

An astronaut is missing on a space walk. You have been sent into orbit in the Shuttle to find and rescue him. Mission control has narrowed his location down to an eight by eight kilometer cube of space. You must search this area of space and find the astronaut before his air runs out. You have just ten hours to find him.

The computer-controlled radar on board the shuttle will help you to find him. Unfortunately the computer has been programmed in a hurry. The best it can do for you is to tell you the general direction of the lost astronaut along the three axes of travel.

The computer will ask you what area of the cube you wish to search. Enter three numbers from one to eight separated by commas. The computer will look at this area and tell you that you have either found the lost astronaut or it will give you clues to help you look again.

Line 260 accepts your co-ordinates (D,E,F). Line 290 checks these against the location of the astronaut (A,B,C) to see if you have found him. If you haven't, lines 330 to 370 check to see where the astronaut is in relation to you and then provide the clues. Variable J counts the number of turns and keeps track of the astronaut's oxygen supply by using a loop.

```
10 REM SPACE RESCUE
20 CLS
30 FOR SS=1 TO 200 STEP 10
40 SOUND SS,1:NEXT
50 FOR SS=220 TO 20 STEP -10
60 SOUND SS,1:NEXT
```

```
70 CLS RND(8)
80 CC=0
90 PRINT@ 192, TAB(5) "### SPACE RESCUE
###"
100 SOUND 2,3
110 FOR DD=1 TO 100:NEXT
120 PRINT@ 192,
130 FOR DD=1 TO 50:NEXT
140 CC=CC+1
150 IF CC=6 GOTO 170
160 GOTO 90
170 CLS RND(8)
180 PRINT@ 192, "YOU HAVE 10 HOURS TO FI
ND AN    ASTRONAUT LOST IN AN 8 KILOMETE
R CUBE OF SPACE"
190 FOR DD=1 TO 1500:NEXT
200 A=RND(8)
210 B=RND(8)
220 C=RND(8)
230 FOR J=1 TO 10
240 CLSRND(8)
250 PRINT@ 192, TAB(3)"WHERE DO YOU WISH
 TO SEARCH"
260 PRINT TAB(6)"ENTER CO-ORDINATES";:IN
PUT D,E,F
270 CLS
280 SOUND 225,2
290 IF A=D AND B=E AND C=F THEN 520
300 PRINT@ 32, TAB(6) "ASTRONAUT NOT THE
RE"
310 PRINT@ 64, TAB(4) "HOURS OF AIR REMA
INING:";10-J
320 PRINT@ 140, "MOVE:"
```

```
330 IF A<D THEN PRINT@ 204, "UP"
340 IF A>D THEN PRINT@ 204, "DOWN "
350 IF B<>E THEN PRINT@ 268, "ACROSS"
360 IF C>F THEN PRINT@ 332, "FORWARDS"
370 IF C<F THEN PRINT@ 332, "BACKWARDS"
380 IF J=9 THEN PRINT "DANGER ASTRONAUTS
 DEATH IMMINENT"
390 FOR JJ=1 TO 5
400 SOUND 5,1
410 FOR DD=1 TO 50:NEXT
420 SOUND 10,1
430 NEXT JJ
440 FOR DD=1 TO 100:NEXT
450 NEXT J
460 CLS RND(8)
470 PRINT@ 160, TAB(6) "YOU HAVE FAILED"
480 PRINT TAB(4)"ASTRONAUT OUT OF AIR"
490 PRINT TAB(4) "ASTRONAUT WAS AT"A;B;C
500 FOR DD=1 TO 2500:NEXT
510 GOTO 600
520 CLS RND(8)
530 FOR SS=10 TO 100 STEP 10
540 SOUND SS,1:NEXT
550 FOR SS=90 TO 10 STEP -10
560 SOUND SS,1:NEXT
570 PRINT@ 192, TAB(2) "YOU HAVE FOUND T
HE ASTRONAUT"
580 PRINT TAB(2) "HE HAD ";10-J;" HOURS
OF AIR LEFT"
590 FOR DD=1 TO 2000:NEXT
600 CLS RND(8)
610 PRINT@ 192, "DO YOU WISH TO TRY AGAI
N (Y/N)";:INPUT H$
```

```
620 IF H$="Y" THEN 180
630 CLS 0
640 X=RND(30)+15
650 Y=RND(20)+5
660 C=RND(8)
670 SS=RND(50)+200
680 SET(X,Y,C)
690 SET(X,Y+1,C)
700 SET(X+1,Y,C)
710 SET(X+1,Y+1,C)
720 SOUND SS,1
730 GOTO 640
```

MARTIAN MODULE

Do you have the skill to become the first Spaceperson to land a spacecraft on the surface of Mars? Many have tried, none have succeeded.

Your on-board computer will keep you informed of your progress. It will monitor your descent and provide you with details of fuel remaining, your height above the surface, rate of fall and your time in flight. The computer will also warn you if the unthinkable should happen, and a problem develop. By entering the amount of thrust required you can manoeuvre your craft down to a soft landing. However, if you misjudge your rate of fall.

The range of thrust available is from −50 to 50. The negative numbers provide thrust against gravity. The positive numbers provide thrust toward the surface.

The progress uses CIRCLE and PAINT to show you the proposed landing site (lines 100–120). Lines 130 to 170 provide a sound loop to give suitable sound effects at the beginning of the game. Variable F is the amount of energy available for thrust. The value of F is decided in line 220.

Variable B controls the malfunction-routine. If B ceases to equal one, your craft develops a fault. This occurs in line 1070 and is tested in line 560. Variable Q controls the air supply in the craft. This is altered in line 1040 and tested in line 530. T is the time the flight has been in progress. T is incremented by E, the length of time thrust is used each turn.

```
10 REM MARTIAN MODULE
20 CLS
30 M=0
40 T=0
50 S=0
```

```
60 H=3000
70 PMODE 1,1
80 PCLS(1)
90 SCREEN 1,0
100 CIRCLE (128,92),75
110 PAINT (0,0),3,4
120 PAINT (128,92),4,4
130 C=0
140 S=RND(50)+200
150 SOUND S,1
160 C=C+1
170 IF C=20 THEN 190
180 GOTO 140
190 PRINT@ 33,  "WELCOME TO MARTIAN MODU
LE.     YOUR MISSION: TO PILOT THE FIRST
 MARS LANDING CRAFT."
200 PRINT "GOOD LUCK":GOSUB 1100
210 CLS:PRINT@ 38,"MARTIAN MODULE"
220 F=600/RND(3)
230 Q=-17
240 B=1
250 PRINT
260 PRINT
270 SOUND 200,1:SOUND 200,1:SOUND 200,1
280 GOTO 510
290 PRINT "POSITVE NOS. TOWARDS SURFACE"
300 PRINT "NEGATIVE NOS. AWAY FROM SURFA
CE"
310  INPUT Z
320 IF Z<-50 OR Z>50 THEN 600
330 INPUT "FOR HOW LONG (SECONDS)";E
340 CLS
```

```
350 PRINT@ 105, "THRUST:";Z
360 PRINT@ 169, "FOR";E;"SECONDS"
370 GOSUB 1030
380 CLS
390 T=T+E
400 S=S+10+3*E*((Z+1)/B)
410 IF Z=0 THEN GOTO 430
420 F=F-3*E*ABS(Z*RND(2)+1)
430 H=H-E*(S+RND(5))
440 IF H<100 AND H>=0 AND S<40 THEN 690
450 IF H<=0 THEN 620
460 IF F<0 THEN 620
470 X=RND(10)+1
480 IF X=5 AND M<>2 GOSUB 730
490 PRINT
500 SOUND 200,1:SOUND 220,1:SOUND 200,1
510 PRINT "## HEIGHT ABOVE SURFACE";H"##
"
520 IF Q<>-17 THEN Q=Q-RND(16)-1
530 IF Q<0 AND Q>-17 THEN 620
540 IF Q<>-17 THEN PRINT "##AIR LEFT:";Q
;"##"
550 PRINT "## RATE OF FALL:";S;"##"
560 IF B<>1 THEN PRINT "## WARNING ##...
THRUST ERRATIC"
570 PRINT "## ENERGY LEFT:";F;"##"
580 PRINT "## TIME IN FLIGHT:";T;"##"
590 PRINT
600 PRINT "AMOUNT OF THRUST (-50 TO 50)?
"
610 GOTO 290
620 CLS:PRINT@ 170,"**crash***"
```

```
630 IF Q<0 AND Q>-17 THEN PRINT@ 202, "A
IR SUPPLY EXHAUSTED"
640 IF F<0 THEN PRINT@ 232, "ENERGY EXHA
LISTED"
650 FOR W=1 TO 100 STEP 10:SOUND W,1:NEX
T
660 FOR W=90 TO 1 STEP -10:SOUND W,1:NEX
T
670 PRINT@ 264,"HIT SURFACE AT";ABS(S)
680 END
690 CLS:PRINT@ 164,"<<<SUCCESSFULL LANDI
NG>>>"
700 PRINT@ 264,"FINAL VELOCITY:";ABS(S)
710 FOR W=10 TO 240 STEP 20:SOUND W,1:NE
XT:FOR W=220 TO 10 STEP -20:SOUND W,1:NE
XT:END
720 RETURN
730 CLS
740 M=M+1
750 FOR V=1 TO 4
760 PRINT
770 NEXT V
780 U=RND(10)
790 FOR V=1 TO4
800 PRINT@ 71, "*** DANGER ***"
810 PRINT@ 135, "MISSION CONTROL"
820 PRINT@ 198, "WE HAVE A PROBLEM"
830 PRINT@ 263, "*** DANGER ***"
840 GOSUB 1030
850 CLS
860 C=0
870 PRINT@ 230, "### MALFUNCTION ###"
880 C=C+1
```

```
890 SOUND 50,5:FOR X=1 TO 200:NEXT X
900 PRINT@ 230, "####################"
910 SOUND 50,5
920 FOR X=1 TO 200:NEXT
930 IF C=4 GOTO 950
940 GOTO 870
950 CLS
960 PRINT@ 97, "USE ACCESS CODE";U;"FOR
DETAILS"
970 INPUT U
980 CLS
990 IF U<>U THEN 620
1000 ON INT (2*RND(0))+1GOSUB 1040,1070
1010 INPUT "PRESS ENTER TO RETURN TO FLI
GHT";V$
1020 CLS
1030 FOR X=1 TO 1000:NEXT:RETURN
1040 Q=INT(19*RND(0))+101
1050 PRINT@ 196, "OXYGEN METER UNRELIABL
E"
1060 RETURN
1070 B=B+RND(3)+1
1080 PRINT@ 196, "THRUST CONTROL ERRATIC
"
1090 RETURN
1100 FOR Y=1 TO 2000:NEXT:RETURN
```

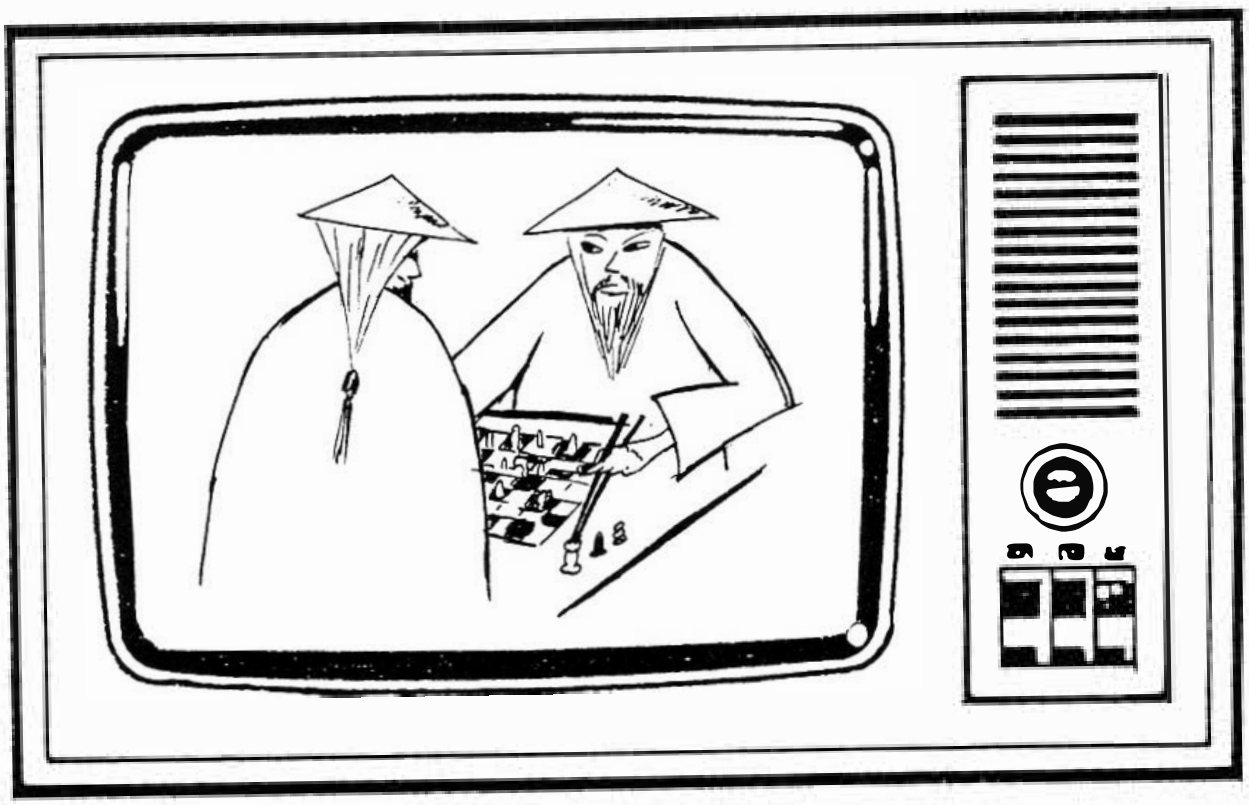
BOARD GAMES:

CHINESE CHESS
(REVERSI/OTHELLO)
CHECKERBOARD NIM
CHECKERS
HAMADRYAD

CHINESE CHESS (OTHELLO/REVERSI)

This game is based on an old board game popular in the latter half of the last century.

It is played on an eight by eight grid. The game begins with the centre four locations of the grid being occupied by two of your playing pieces (white) and two of the computers playing pieces (black). Each player takes turns placing a piece on the board adjacent to one of his opponent's pieces. Any of your opponent's pieces *between* the piece you have just placed, and *another* of your pieces are captured. They are left on the board but become yours.



The game ends when all the grid positions are occupied or when neither player is able to move. The winner is the player with the most pieces on the board.

At the beginning of the game the computer will ask you if you wish to have the first move. To place a playing piece on the board enter its required location starting with the number down the side of the grid followed by the horizontal location. Enter the numbers as a two-digit figure, for example 45.

The computer will keep track of the score and automatically replace captured pieces with the opponent's pieces. You will find that your computer plays this game slowly but it will play very well. In fact, you will find it very difficult to win a game.

Lines 750to 890 use PRINT@ to print up the grid pattern and keep it in place during the course of the game. The lines 320 to 630 contain the logic for the computer's moves. Lines 640 to 720 control your moves. Lines 900 to 1020 act on the moves and adjust the various values used in the printout.

```
10 REM CHINESE CHESS
20 CLS
30 CC=0
40 SS=RND(20)+175:SOUND SS,RND(2)
50 CC=CC+1
60 IF CC=6 THEN 80
70 GOTO 40
80 GOSUB 1130
90 CLS
100 CC=0
110 SS=RND(25)+150:TT=RND(2)
120 SOUND SS,TT
130 CC=CC+1:IF CC=15 GOTO 150
140 GOTO 110
```

```
150 X=ASC(CHR$(128)):O=ASC(CHR$(207))
160 DIM A(10,10)
170 FOR B=1 TO 10
180 FOR C=1 TO 10
190 IF B<>1 AND C<>1 AND B<>10 AND C<>10
 THEN A(B,C)=ASC(".")
200 NEXT C
210 NEXT B
220 A(5,5)=X
230 A(6,6)=X
240 A(6,5)=O
250 A(5,6)=O
260 P=0
270 PRINT@196,"DO YOU WISH TO GO FIRST"
280 PRINT@233,"YES-1      NO-2"
290 SOUND200,1:PRINT@268,"  ";:INPUT W:CL
S
300 GOSUB 740
310 IF W=1 THEN 640
320 SOUND100,1:PRINT@459,"MY MOVE"
330 S=O
340 T=X
350 H=0
360 FOR A=2 TO 9
370 FOR B=2 TO 9
380 IF A(A,B)<>46 THEN 580
390 Q=0
400 FOR C=-1 TO 1
410 FOR D=-1 TO 1
420 K=0:F=A:G=B
430 IF A(F+C,G+D)<>S THEN 460
440 K=K+1:F=F+C:G=G+D
450 GOTO 430
```

```
460 IF A(F+C,G+D)<>T THEN 480
470 Q=Q+K
480 NEXT D
490 NEXT C
500 IF A=2 OR A=9 THEN Q=Q*2
510 IF B=2 OR B=9 THEN Q=Q*2
520 IF A=3 OR A=8 THEN Q=Q/2
530 IF B=3 OR B=8 THEN Q=Q/2
540 IF (A=2 OR A=9) AND (B=3 OR B=8) THE
N Q=Q/2
550 IF (A=3 OR A=8) AND (B=2 OR B=9) THE
N Q=Q/2
560 IF Q<H OR Q=0 OR (RND(10)>3 AND Q=H)
 THEN 580
570 H=Q:M=A:N=B
580 NEXT B
590 NEXT A
600 IF H=0 AND R=0 THEN 1030
610 IF H=0 THEN 630
620 GOSUB 900
630 GOSUB 740
640 SOUND200,1:PRINT@458," ";:INPUT "YOU
R MOVE";R
650 S=X
660 T=0
670 IF R=0 THEN 720
680 IF R<11 OR R>88 THEN 640
690 M=INT(R/10)+1
700 N=R-10*INT(R/10)+1
710 GOSUB 900
720 GOSUB 740
730 GOTO 320
740 C=0:H=0
```

```
750 PRINT@ 5, "*** CHINESE CHESS ***"
760 PRINT @76, "12345678"
770 FOR B=2 TO 9
780 PRINT TAB(9) B-1;
790 FOR D=2 TO 9
800 PRINT CHR$(A(B,D));
810 IF A(B,D)=X THEN C=C+1
820 IF A(B,D)=O THEN H=H+1
830 NEXT D
840 PRINT B-1
850 NEXT B
860 PRINT TAB(12) "12345678"
870 PRINT@419,"I HAVE";C
880 PRINT@435," YOU HAVE";H
890 RETURN
900 FOR C=-1 TO 1
910 FOR D=-1 TO 1
920 F=M:G=N
930 IF A(F+C,G+D)<>S THEN 960
940 F=F+C:G=G+D
950 GOTO 930
960 IF A(F+C,G+D)<>T THEN 1010
970 A(F,G)=T
980 IF M=F AND N=G THEN 1010
990 F=F-C:G=G-D
1000 GOTO 970
1010 NEXT D:NEXT C
1020 RETURN
1030 C=0:H=0:FOR B=2 TO 9
1040 FOR D=2 TO 9
1050 IF A(B,D)=X THEN C=C+1
1060 IF A(B,D)=O THEN H=H+1
1070 NEXTD:NEXTB:CLS:PRINT@392," ";
```

```
1080 FOR F=200TO250STEP5:SOUNDF,1:NEXTF
1090 IF C>H THEN PRINT "i won";C;H
1100 IF H>C THEN PRINT "you won";H;C
1110 IF H=C THEN PRINT "it is a draw";C;
H
1120 END
1130 CLS RND(8)
1140 XX=0
1150 PRINT@ 192, TAB(5) "## CHINESE CHE
SS ##":FOR CC=1 TO 10:NEXT CC:SOUND RND
(10)+190,1
1160 PRINT@ 192, TAB(5) "#*#*#*#*#*#*#*#
*#*#*#":FOR CC=1 TO 100:NEXT CC
1170 XX=XX+1:IF XX=10 THEN RETURN
1180 GOTO 1150
1190 RETURN
```

CHECKERBOARD NIM

This program sets up a checkerboard and places a random number of playing pieces on the green squares.

You and the computer then take turns removing pieces from the board. The one who is forced to take the last piece is the loser.

Each game the computer will place a different number of pieces on the board and set a different maximum number of pieces which may be removed at any one time. The computer will display on the screen the number of pieces on the board, the maximum number you are allowed to remove, the number of pieces both you and the computer removed that turn and the graphic display of the checkerboard.

The computer has been programmed not to play perfectly. Line 360 puts a random factor into the computer's moves. Without this line, it would be impossible to beat.

Lines 120 to 210 print the board onto the screen. This is done by two loops using the values of A and B. Variable Z is the number of pieces on the board. Z is modified by your input (C) in line 290 and by the computer's move in line 390 (D). Lines 330 to 390 are the lines the computer uses to make its moves.

```
10 REM CHECKERBOARD NIM
20 GOSUB 490
30 S=0
40 C=0
50 Z=20+RND(11)
60 F=2+RND(3)
70 DIMA(32)
80 GOTO 410
90 IF S=0 THEN PRINT@ 6, "PIECES ON BOAR
D ";Z
```

```
100 IF S=0 THEN PRINT@ 34, "MAXIMUM NUMB
ER TO REMOVE ";F
110 IF C>0 AND S=0 THEN PRINT@ 68, "YOU
TOOK ";C;", I TOOK ";D
120 FOR A=0 TO 3
130 PRINT
140 FOR B=0 TO 3
150 PRINTTAB(11)CHR$ (A(29+A-B*8));CHR$(
128);
160 NEXT B
170 PRINT
180 FOR B=0 TO 3
190 PRINT TAB(11)CHR$(128);CHR$(A(25+A-B
*8));
200 NEXT B
210 NEXT A
220 IF S=1 THEN PRINT@ 331"YOU WIN":GOTO
 780
230 IF S=2 THEN PRINT@ 332,"I WIN":GOTO
780
240 IF S>0 THEN END
250 PRINT
260 PRINT@ 420, "HOW MANY WILL YOU TAKE"
;
270 INPUT C
280 IF C<1 OR C>F THEN GOTO 270
290 Z=Z-C
300 IF Z>0 THEN 330
310 S=2
320 IF S=2 THEN 410
330 D=Z-1-INT((Z-1)/(F+1))*(F+1)
340 IF D=0 AND NOT Z=1 THEN D=RND(F)
350 IF NOT D<Z THEN 340
```

```
360 IF Z<F+2 AND RND(4)=4 THEN D=D+RND(2
)-RND(2)
370 IF D>F THEN GOTO 330
380 IF D=0 THEN D=1
390 Z=Z-D
400 IF Z=0 THEN S=1
410 FOR A=1 TO Z
420 A(A)=94
430 NEXT A
440 FOR A=Z+1 TO 32
450 A(A)=143
460 NEXT A
470 CLS
480 GOTO 90
490 PCLS
500 PMODE 1,1
510 SCREEN 1,1
520 FOR JJ=1 TO 260 STEP 20
530 CIRCLE(JJ,96),50
540 NEXT JJ
550 SOUND 200,3
560 FOR JJ=1 TO 260 STEP 20
570 CIRCLE(JJ,48),20
580 NEXT JJ
590 SOUND 210,3
600 FOR JJ=1 TO 260 STEP 20
610 CIRCLE(JJ,144),20
620 NEXT JJ
630 SOUND 220,3
640 FOR DD=1 TO 500:NEXT DD
650 FOR ZZ=1 TO 10
660 CLS RND(8)
```

```
670 PRINT@ 192,TAB(4) "##################
#######"
680 FOR DD=1 TO 15:NEXT
690 SOUND RND(20)+175,2
700 FOR DD=1 TO 15:NEXT DD
710 SOUND RND(20)+175,1
720 FOR DD=1 TO 30:NEXT
730 PRINT@ 192, TAB(4) "### CHECKERBOARD
 NIM ###"
740 FOR DD=1 TO 20:NEXT DD
750 SOUND RND(20)+150,2
760 NEXT ZZ
770 RETURN
780 FOR SS=10 TO 250 STEP 10
790 SOUND SS,1
800 NEXT SS
810 FOR SS=240 TO 1 STEP -10
820 SOUND SS,1
830 NEXT SS
840 FOR DD=1 TO 1000:NEXT DD
850 CLS RND(8)
860 PRINT@ 192, "DO YOU WANT ANOTHER GAM
E (Y/N)";
870 INPUT A$
880 IF A$="Y" THEN RUN 10
890 END
```

CHECKERS

Here is a simple Checkers program which you will find quite entertaining to play. The game follows standard rules except that there is no penalty for failing to capture an opponent's piece. However the computer will try to jump one of your pieces whenever it can.

The computer is the red pieces at the top of the board and you are the orange squares at the bottom.

To move a piece just enter the location of the piece you wish to move beginning with the letter across the bottom of the screen then the number down the side, such as G9. Then enter the location of the square you wish to move *to* in the same way.

After you have jumped one of your opponent's pieces the computer will ask you "JUMP AGAIN (Y/N)". If you wish to jump again, enter Y then the new location. If you can't jump again enter N.

The computer always has the first move. The number of pieces you have both captured is shown above the board. The game ends when one of you has lost all of your pieces, or the computer decides it is in a hopeless position and concedes the game.

Strategy isn't the computer's greatest virtue in this program but it will be very reluctant to move into danger and it will put up a spirited defence when under attack. Kings are created automatically and appear on the board as two smaller, diagonally joined squares of the appropriate colour.

There is no mechanism in the program to prevent you from cheating but there is no point in doing so. Illegal moves

confuse the program and pieces moved illegally may tend to disappear.

The subroutine starting at line 970 uses LINE and PAINT to draw the checkerboard pattern on the screen at the start of the game. The routine starting at line 710 prints the board on the screen using PRINTS and CHR$)

```
10 REM CHECKERS
20 GOSUB 970
30 CLS
40 PRINT@232,"PLEASE STAND BY"
50 GOSUB 880
60 Z=24
70 Q=0
80 FOR G=69 TO 72:IF A(G)=-1 THEN A(G)=-
2
90 NEXT
100 GOSUB 710
110 IF A(Z)=9 OR A(Z)<1 THEN 200
120 IF Z<28 AND A(Z)=1 THEN A(Z)=2
130 Y=1
140 IF A(Z+X(Y))<0 AND A(Z+2*(X(Y)))=0 T
HEN Q=X(Y)
150 IF A(Z)=2 AND A(Z-X(Y))<0AND A(Z-2*X
(Y))=0 THEN Q=-X(Y)
160 IF Q<>0 AND Z+2*Q>23 THEN 230
170 Q=0
180 IF Y=2 THEN 200
190 Y=2:GOTO 140
200 Z=Z+1
210 IF Z<73 THEN 110
220 IF Q=0 THEN 300
```

```
230 A(Z+Q)=0:A(Z+2*Q)=A(Z):A(Z)=0
240 Z=Z+2*Q:CO=CO+1:GOSUB 710
250 Q=0:Y=1
260 IF A(Z+X(Y))<0 AND A(Z+2*(X(Y)))>0 T
HEN Q=X(Y)
270 IF Q<>0 AND Z+2*Q>23 THEN 230
280 IF Y=1 THEN Y=2:GOTO 260
290 GOTO 450
300 U=0:Q=0
310 Z=24+INT(RND(0)*49):U=U+1
320 IF(A(Z)=9 OR A(Z)=-1 OR A(Z)=-2 OR A
(Z)=0) AND U<1000 THEN310
330 Y=1
340 IF A(Z+X(Y))<>0 THEN 400
350 IF A(Z+X(Y))=0 AND A(Z+2*X(Y))>-1 AN
D A(Z+2*X(Y)+1)>-1 AND A(Z+2*X(Y)-1)>-1
THEN Q=X(Y)
360 IF A(Z+X(Y))=0 AND U>150 THEN Q=X(Y)
370 IF A(Z)=2 AND A(Z-X(Y))=0 AND A(Z-2*
X(Y))>-1 AND A(Z-2*X(Y)+1)>-1 THEN Q=-X(
Y)
380 IF U>600 AND A(Z)=2 AND A(Z-X(Y))=0
THEN Q=-X(Y)
390 IF Q<>0 THEN 430
400 IF Y=1 THEN Y=2:GOTO 350
410 IF U<1000 THEN 310
420 CLS:PRINT@230,"I CONCEDE THE GAME":E
ND
430 A(Z+Q)=A(Z):A(Z)=0
440 GOSUB 710
450 PRINT@398,"                ":PRINT@388,"M
OVE FROM:";:INPUTA$:PRINT@404,"TO:";:INP
UTB$
```

```
460 IF U$<>""THEN U$=""
470 FOR W=1 TO 2:Z=0
480 IF W=1 THEN C$=A$
490 IF W=2 THEN C$=B$
500 Z=-24*(C$="G9")-25*(C$="E9")-26*(C$=
"C9")-27*(C$="A9")-30*(C$="H8")-31*(C$="
F8")
510 IF Z<>0 THEN 610
520 Z=-32*(C$="D8")-33*(C$="B8")-37*(C$=
"G7")-38*(C$="E7")-39*(C$="C7")-40*(C$="
A7")
530 IFZ<>0 THEN 610
540 Z=-43*(C$="H6")-44*(C$="F6")-45*(C$=
"D6")-46*(C$="B6")-50*(C$="G4")-51*(C$="
E4")
550 IF Z<>0 THEN 610
560 Z=-52*(C$="C4")-53*(C$="A4")-56*(C$=
"H3")-57*(C$="F3")-58*(C$="D3")-59*(C$="
B3")-63*(C$="G2")
570 IF Z<>0 THEN 610
580 Z=-64*(C$="E2")-65*(C$="C2")-66*(C$=
"A2")-69*(C$="H1")-70*(C$="F1")
590 IF Z<>0 THEN 610
600 Z=-71*(C$="D1")-72*(C$="B1")
610 IF W=1 THEN D=Z
620 IF W=2 THEN E=Z
630 NEXT: A$="":B$="":C$=""
640 A(E)=A(D):A(D)=0
650 IF ABS(D-E)>7 THEN A((D+E)/2)=0:HU=H
U+1
660 GOSUB 710
```

```
670 IF ABS(D-E)>7 THEN PRINT@421,"JUMP A
GAIN (Y/N)";:INPUTU$:PRINT@421,"
                ":IF U$<>"N" THEN 450
680 IF HU<12 AND CO<12 THEN 60
690 IF HU>11 THEN CLS:PRINT@236,"YOU WIN
":END
700 IF CO>11 THEN 1120
710 FOR M=24 TO 72
720 A(M)=-191*(A(M)=1)-185*(A(M)=2)-207*
(A(M)=0)-255*(A(M)=-1)-249*(A(M)=-2)-9*(
A(M)=9)
730 NEXT
740 PRINT@14,"SCORE":PRINT@36,"COMPUTER:
";CO:PRINT@51,"HUMAN:";HU
750 PRINT@76,"ABCDEFGH"
760 T=-2:FOR K=0 TO 3:PRINT TAB(12);:FOR
 J=0 TO 3
770 PRINT CHR$(128);CHR$(A(72-J-13*K));:
NEXT:T=T+1
780 PRINT INT((J+K)/2)+T
790 FOR J=0 TO 3:PRINT TAB(12)CHR$(A(66-
J-13*K));CHR$(128);
800 NEXT:T=T+1
810 PRINT INT((J+K)/2)+T
820 NEXT
830 PRINT TAB(12)"ABCDEFGH"
840 FOR M=24 TO 72
850 A(M)=-(A(M)=191)-2*(A(M)=185)+0*(A(M
)=207)+(A(M)=255)+2*(A(M)=249)-9*(A(M)=9
)
860 NEXT
870 RETURN
880 DIM A(99):X(1)=-6:X(2)=-7
```

```
890 FOR Z=1 TO 99:A(Z)=9:NEXT
900 FOR Z=1 TO 32: READ B:READ C:A(B)=C:
NEXT
910 DATA 72,1,71,1,70,1,69,1,66,1,65,1,6
4,1,63,1,59,1,58,1,57,1,56,1
920 DATA 53,0,52,0,51,0,50,0,46,0,45,0,4
4,0,43,0
930 DATA 40,-1,39,-1,38,-1,37,-1,33,-1,3
2,-1,31,-1,30,-1,27,-1,26,-1,25,-1,24,-1
940 CO=0:HU=0
950 SOUND 200,1
960 CLS:RETURN
970 PCLS:PMODE0,1:SCREEN1,1
980 LINE(84,0)-(84,192),PSET
990 LINE(172,0)-(172,192),PSET
1000 LINE(0,64)-(256,64),PSET
1010 LINE(0,128)-(256,128),PSET
1020 PAINT(88,0),5,5:PAINT(88,138),5,5:P
AINT(0,126),5,5:PAINT(176,126),5,5
1030 CC=0
1040 SS=RND(50)+150:SOUND SS,1:CC=CC+1
1050 IF CC=15 GOTO 1070
1060 GOTO 1040
1070 CLS
1080 PRINT@ 160, TAB(6) CHR$(207);CHR$(1
28);CHR$(207);CHR$(128);CHR$(207):PRINT@
 181, CHR$(207);CHR$(128);CHR$(207);CHR$
(128);CHR$(207)
1090 PRINT@ 192, TAB(6) CHR$(128);CHR$(2
07);CHR$(128);CHR$(207);CHR$(128);" CHEC
KERS ";CHR$(128);CHR$(207);CHR$(128);CHR
$(207);CHR$(128)
```

```
1100 PRINT@ 224, TAB(6) CHR$(207);CHR$(1
28);CHR$(207);CHR$(128);CHR$(207):PRINT@
 245,CHR$(207);CHR$(128);CHR$(207);CHR$(
128);CHR$(207)
1110 FOR TT=1 TO 1000:NEXT:RETURN
1120 A$=CHR$(175):B$="  ":C$="  ":CLS
1130 FOR C=200TO250 STEP5:SOUNDC,1:NEXT
1140 PRINT@131,A$;B$;A$
1150 PRINT@144,A$;B$;A$;B$;A$;A$;C$;A$
1160 PRINT@163,A$;C$;A$;C$;A$;C$;A$;C$;A
$;B$;A$;" ";A$;B$;A$
1170 PRINT@195,A$;B$;B$;A$;" ";A$;" ";A$
;" ";A$;B$;B$;A$;B$;A$;B$;A$;" ";A$
1180 PRINT@227,A$;B$;C$;A$;C$;A$;C$;B$;A
$;B$;A$;C$;A$;A$
```

HAMADRYAD

A hamadryad is a member of a class of nymphs in classical mythology, who inhabits a tree and dies with it. The game named after this creature is a kind of mythological Checkers, demonstrating how the game might have developed from different starting premises.

Hamadryad is played on a nine by eight board. The computer is the V pieces at the top of the screen, and you are the + signs at the bottom. The hash signs (#) denote the black squares.

The object of the game is to capture six of your opponent's pieces. A piece is captured by landing one of your own pieces on the same square, not by jumping over it as in Checkers. Pieces move diagonally, and may be moved both forwards and backwards. This feature has been added so that both you and the computer can retreat from danger when needed.

The computer has the first move. It will then ask you "MOVE FROM SQUARE?". You then enter the co-ordinates of the piece you wish to move, starting with the number down the side of the board, then a comma, followed by the number at the bottom of the board. For example, if you wished to move the piece at the far right of your first row, you would type 6,8. After you've done this, the computer will ask "MOVE TO SQUARE?". Simply enter the co-ordinates of the square you wish to move to in exactly the same way.

Hamadryad uses DATA statements to set up the board and playing pieces. This information is positioned on the screen by the PRINT@ in line 520.

The variable CS is the computer's score, and HS is the human player's score. These are displayed on the screen by line 550.

Lines 220 and 230 use PRINT@ and several blank spaces to remove your previous moves from the screen. These blank spaces clear out the answers to the questions the computer asks you in lines 240 and 250.

```
20 CLS
30 DATA "#123456789#"
40 DATA "1U#U#U#U#U1"
50 DATA "2#U#U#U#U#2"
60 DATA "3U#U# #U#U3"
70 DATA "4# # #U# #4"
80 DATA "5 # # # # 5"
90 DATA "6#+#+#+#+#6"
100 DATA "7+#+#+#+#+7"
110 DATA "8#+#+#+#+#8"
120 DATA "#123456789#
130 DIMS$(12,13)
140 FOR A=1 TO 10
150 READ B$
160 FOR B=1 TO 11:S$(A,B)=MID$(B$,B,1):N
EXT B:NEXT A
170 IF INT(RND(0)+.5)=0 THEN 190
180 S$(5,5)="U":S$(5,7)=" "
190 CLS
200 GOSUB 520
210 IF CS=6 THEN PRINT@ 10,  "i win!!!":
END
220 PRINT@ 433, "    "
230 PRINT@ 463, "    "
240 PRINT@ 416, "MOVE FROM SQUARE";
250 INPUT A,B
260 PRINT@ 448, "MOVE TO SQUARE";
270 INPUT C,D
```

```
280 IF ABS(A-C)=1 AND ABS(B-D)=1 THEN 30
0
290 PRINT@ 10, "illegal move":GOTO 220
300 IF S$(C+1,D+1)="V" THEN HS=HS+1
310 S$(A+1,B+1)=" ":S$(C+1,D+1)="+"
320 GOSUB520
330 IF HS=6 THEN PRINT@ 10,  "you win":E
ND
340 A$="+":GOSUB 410
350 IF FL=1 THEN 370
360 A$=" ":GOSUB 410
370 S$(E,F)=" "
380 IF S$(E+G,F+H)="+" THEN CS=CS+1
390 S$(E+G,F+H)="V"
400 GOTO 200
410 E=2:F=2:G=0:H=0
420 FL=0
430 IF S$(E,F)<>"V" THEN 490
440 IF S$(E+1,F+1)=A$ THEN G=1:H=1
450 IF S$(E+1,F-1)=A$ THEN G=1:H=-1
460 IF S$(E-1,F+1)=A$ THEN G=-1:H=1
470 IF S$(E-1,F-1)=A$ THEN G=-1:H=-1
480 IF G<>0 AND H<>0 THEN FL=1:RETURN
490 E=E+1:IF E>10 THEN E=2:F=F+1
500 IF F>11 THEN RETURN
510 GOTO 420
520 PLAY "L8;A;D;B;E":PRINT@ 9, S$:FOR A
=1 TO 10:PRINT:FOR B=1 TO 11
530 PRINT TAB(10) S$(A,B);
540 NEXT:NEXT
550 PRINT@ 384, "COMPUTER:";CS,"HUMAN:";
HS
560 RETURN
```

BRAIN TWISTERS:

CUBIK'S RUBE
MAGIC SQUARE
CODE CRACKER
HANGMAN
CAT AND MOUSE
REVERSE
FLIP FLOP

CUBIK'S RUBE

This program is a simple version of a cube puzzle. The computer will print up sixteen coloured blocks on a four by four grid. Four colours are used to show the blocks. At the beginning of the game the computer will show you how the grid should look. All blocks of the same colour will be grouped together to form a neat pattern.

The computer will then ask you if you want the computer to randomly scramble the cube or if you wish to mix it up yourself. If you wish the computer to scramble the cube the routine in lines 490 to 590 makes 10 random moves of the cube.

There are eight moves possible. Each of the four vertical columns and each of the four horizontal columns can be moved. Entering a number between one and four will cause the relevant vertical line to scroll one space upward. A number between five and eight will cause the relevant line to rotate one space to the left.

After the cube has been scrambled you will be able to spend many frustrated hours trying to get it back into the correct order. You may change the colours of the blocks by altering the CHR$ values in lines 760 to 790.

```
10 REM CUBIKS RUBE
20 PCLS
30 PMODE 1,1
40 SCREEN 1,1
50 FOR X=1 TO 256 STEP 40
60 FOR Z=1 TO 256 STEP 40
70 LINE (X,0)-(Z,192),PSET
80 NEXT:NEXT
```

```
90 FOR ZZ=1 TO 500:NEXT
100 FOR SS=200 TO 240 STEP 5
110 SOUND SS,1:NEXT
120 FOR SS=235 TO 200 STEP -5
130 SOUND SS,1:NEXT:CLS RND(8)
140 PRINT@ 192,TAB(4) "##### CUBIKS RUBE
 #####"
150 FOR DD=1 TO 500:NEXT
160 FOR T=2 TO 20 STEP 2
170 FOR SS=100 TO 160 STEP T
180 SOUND SS,1
190 NEXT:NEXT
200 F=100
210 DIMA(4,4),B(4,4)
220 FORX=1TO2
230 FORY=1TO2
240 A(X,Y)=1
250 NEXT
260 FORY=3TO4
270 A(X,Y)=3
280 NEXTY
290 NEXTX
300 FORX=3TO4
310 FOR Y=1TO2
320 A(X,Y)=2
330 NEXTY
340 FORY=3TO4
350 A(X,Y)=4
360 NEXTY:NEXT X
370 GOSUB880
380 CLS:PRINT@160," "
390 FORX=1TO4
400 PRINTTAB(13)
```

```
410 FORY=1TO4
420 IFA(X,Y)=1GOSUB760
430 IFA(X,Y)=2GOSUB770
440 IFA(X,Y)=3GOSUB780
450 IFA(X,Y)=4GOSUB790
460 NEXT
470 PRINT
480 NEXT
490 IF F=10 THEN 600
500 IF F<10 THEN 590
510 FOR G=1 TO 500:NEXT G
520 IF E$="N" THEN 600
530 E$=""
540 PRINT@ 416, "DO YOU WANT THE CUBE SC
RAMBLED   (Y/N)";:INPUT E$
550 IF E$="N" THEN 600
560 FOR F=I TO 10
570 C=RND(8)
580 GOTO 690
590 NEXT F
600 PRINT@ 320, TAB(13) "^^^^^"
610 PRINT@ 352, TAB(13) "1234"
620 PRINT@ 209, "<5"
630 PRINT@ 241, "<6"
640 PRINT@ 273, "<7"
650 PRINT@ 305, "<8"
660 PRINT@ 416, "
                "
670 PRINT@ 416, "TWIST LINE";:INPUT C
680 SOUND 200,2
690 IFC>4GOTO800
700 FORX=1TO3
```

```
710 A(X,C)=B(X+1,C)
720 NEXT X
730 A(4,C)=B(1,C)
740 GOSUB880
750 GOTO380
760 PRINTCHR$(159);:RETURN
770 PRINTCHR$(255);:RETURN
780 PRINTCHR$(175);:RETURN
790 PRINTCHR$(191);:RETURN
800 D=C-4
810 FORX=1TO3
820 G=X+1
830 A(D,X)=B(D,G)
840 NEXT X
850 A(D,4)=B(D,1)
860 GOSUB880
870 GOTO380
880 FORX=1TO4
890 FORY=1TO4
900 B(X,Y)=A(X,Y)
910 NEXT:NEXT
920 RETURN
```

MAGIC SQUARE

Magic Square is an exercise in mental arithmetic. The computer generates a three by three grid. Some of the grid positions are occupied by numbers, some are left blank.

The computer challenges you to fill in the blank spaces with the correct numbers. The only clue you need is that the total of each column (horizontal and vertical) is the same. Negative numbers and zero are legal.

Numbers for the grid are selected and assigned in lines 60 to 260. Line 280 keeps a running total of the number of guesses you have had. This is done by incrementing variable J. Line 270 performs this function. Lines 320 to 340 print the grid onto the screen using PRINT@ to keep the display static.

```
10 DIM A(9)
20 DIM B(9)
30 W=-99
40 A=RND(9)
50 J=0
60 B=RND(9)
70 C=RND(9)
80 IF A=B OR A=B OR A=C OR B=C THEN GOTO
 60
90 A(1)=A+B
100 A(2)=A-(B+C)
110 A(3)=A+C
120 A(4)=A-B+C
130 A(5)=A
140 A(6)=A+B-C
150 A(7)=A-C
160 A(8)=A+B+C
170 A(9)=A-B
```

```
180 FOR Z=1 TO 9
190 B(Z)=A(Z)
200 NEXT Z
210 K=ABS(A)
220 B(K)=0
230 K=ABS(B)
240 B(K)=0
250 K=ABS(C)
260 B(K)=0
270 J=J+1
280 CLS:PRINT@ 0,"GUESS NO. ";J
290 PRINT
300 PRINT
310 FOR Z=1 TO 9
320 PRINT@ 137,B(1):PRINT@ 143,B(2):PRIN
T@ 149,B(3)
330 PRINT@ 201,B(4):PRINT@ 207,B(5):PRIN
T@ 213,B(6)
340 PRINT@ 265,B(7):PRINT@ 271,B(8):PRIN
T@ 277,B(9)
350 IF M=9 THEN PRINT@ 416, "YOU HAVE SO
LVED IT"
360 IF M=9 THEN 480
370 PRINT@ 416, "YOU HAVE";M;"RIGHT"
380 SOUND190,2
390 INPUT "WHAT NUMBER "; U
400 M=0
410 FOR Z=1 TO 9
420 IF U=-99 THEN GOTO 440
430 IF A(Z)=U THEN B(Z)=U
440 IF B(Z)<>0 THEN M=M+1
450 IF A(Z)=0 AND U=0 THEN M=M+1
460 NEXT Z
```

```
470 GOTO 270
480 FOR S=20 TO 220 STEP 20
490 SOUND S,1
500 NEXT
510 FOR S=200 TO 20 STEP -20
520 SOUND S,1
530 NEXT:END
```

CODE CRACKER

Code Cracker will test your skills of deduction. The computer generates a four-digit code number using numbers from one to nine. Zero is not used and no number can be used twice, so numbers such as 7046 and 9922 will not occur.

The computer will ask you to guess the code. You are required to enter a four-digit number. The computer will consider your guess than give you coded clues to its accuracy. Black means you have correctly guessed a number and have it in the correct place. White means you have the number correct but in the wrong place.

The computer gives you 15 guesses to get the sequence right. After 15 guesses it will tell you the correct answer.

After you have had some practice at Code Cracker you may wish to increase the difficulty to make it a little more challenging. This is done quite simply. Just alter the value of C in line 150.

The computer uses lines 100 to 130 to generate its code number. Line 130 prevents the same number being picked twice.

Line 140 combines the four numbers into a four digit number. That is it would make 1,2,3,4 become one thousand two hundred and thirty four. It then assigns this value to D.

Line 170 accepts your number and assigns it to X. Lines 180 to 210 reverse the process of line 140. That is it converts your input of one thousand two hundred and thirty four into 1,2,3,4. Line 220 then checks to see if D equals X, in case you have guessed the code number correctly.

Lines 230 to 350 check your numbers against the computer's code numbers and increments N (black) and W (white) accordingly. Line 410 then prints the results of your guess onto the screen.

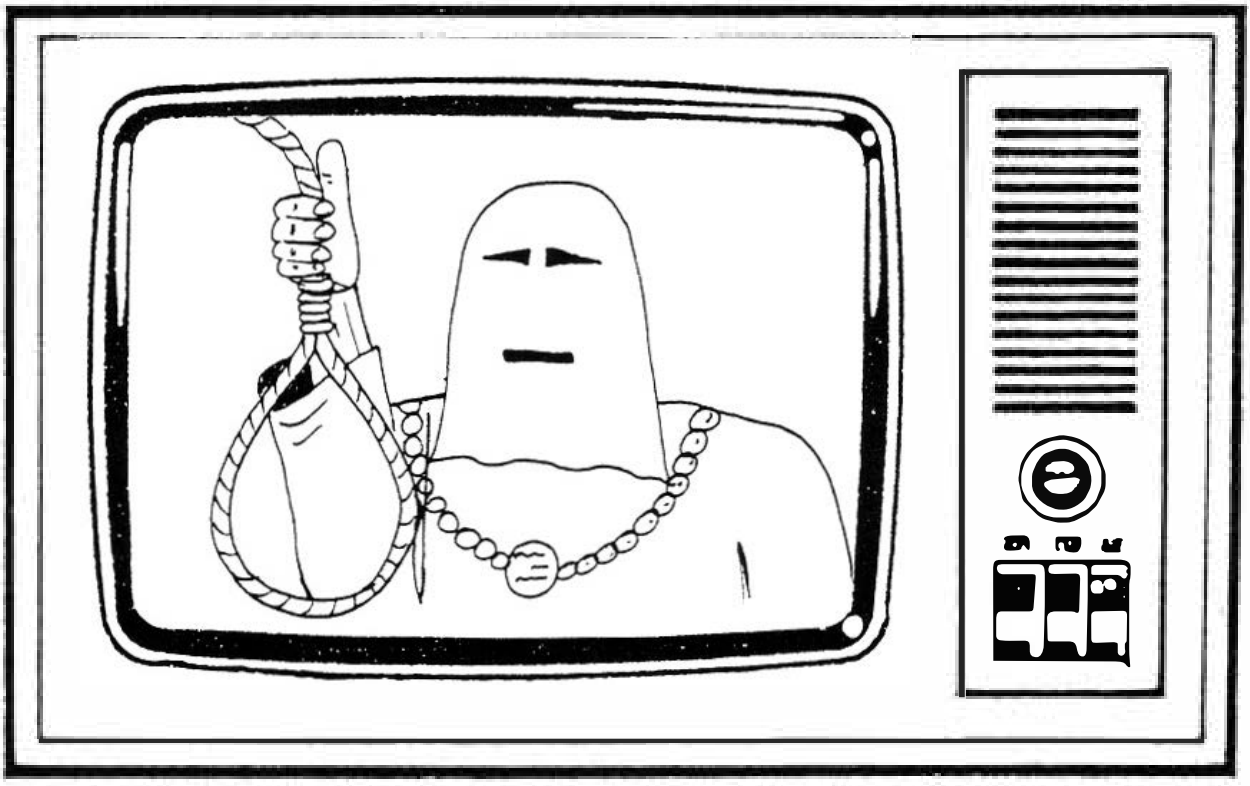
See if you can improve the interest of the print out by getting the computer to print out the appropriate number of black squares (CHR$(128)) and white squares (CHR$(207)), instead of the rather bland statement currently provided.

```
10 REM CODE CRACKER
20 GOSUB 490
30 A=0
40 PRINT@ 192, TAB(7) "**********************
*":FOR C=1 TO 50:NEXT C:CLS
50 PRINT@ 192, TAB(7) "** CODE CRACKER *
*":FOR C=1 TO 100:NEXT C:CLS
60 A=A+1
70 IF A=>10 THEN 90
80 GOTO 40
90 CLS:GOSUB 480
100 FOR Z=1 TO 4
110 A(Z)=RND(9)
120 NEXT Z
130 IF A(1)=A(2) OR A(1)=A(3) OR A(1)=A(
4) OR A(2)=A(3) OR A(2)=A(4) OR A(3)=A(4
) GOTO 100
140 D=1000*A(1)+100*A(2)+10*A(3)+A(4)
150 FOR C=1 TO 15
160 PRINT "GUESS ";C
170 INPUT X
180 B(1)=INT(X/1000)
190 B(2)=INT((X-1000*B(1))/100)
```

```
200 B(3)=INT((X-1000*B(1)-100*B(2))/10)
210 B(4)=X-1000*B(1)-100*B(2)-10*B(3)
220 IF O=X THEN 450
230 N=0:W=0
240 FOR E=1 TO 4
250 IF A(E)<>B(E) THEN 280
260 N=N+1
270 A(E)=0
280 NEXT E
290 FOR F=1 TO 4
300 IF A(F)=0 THEN 350
310 FOR E=1 TO 4
320 IF B(F)<>A(E) THEN 340
330 W=W+1
340 NEXT E
350 NEXT F
360 A(1)=INT(D/1000)
370 A(2)=INT((D-1000*A(1))/100)
380 A(3)=INT((D-1000*A(1)-100*A(2))/10)
390 A(4)=D-1000*A(1)-100*A(2)-10*A(3)
400 PRINT "YOU SCORED";
410 PRINT N;"BLACKS AND";W;"WHITES"
420 NEXT C
430 PRINT "MY NUMBER WAS ";A(1);A(2);A(3
);A(4):GOSUB 480
440 END
450 PRINT "CONGRATULATIONS"
460 PRINT "YOU GOT IT IN ";C:GOSUB 480
470 END
480 FOR T=200 TO 100 STEP -5:SOUND T,1:N
EXT:FOR U=1 TO 2000:NEXT:RETURN
490 FOR Y=1 TO 8:CLS(Y):FOR U=1 TO 200:N
EXT U:NEXT Y:RETURN
```

HANGMAN

Hangman is a traditional word game which involves the guessing of letters in an unknown word. The computer selects a word from its DATA store (lines 320–400) and tells you how many letters the word contains. The computer will also tell you how many guesses it will allow you to have to discover the word.



After each guess the computer will either place a correctly guessed letter in its position in the word or place an incorrect guess in the INCORRECT LETTERS list at the bottom of the screen. For each incorrect guess the computer subtracts one from the number of guesses you have left.

The program relies on a store of words in its DATA files. This program provides a list of 73 words. You can quite easily

increase the computers vocabulary by adding extra lines of DATA between lines 320 and 400. Just follow the format used in these lines. The only other change which needs to be made to the program is in line 30. The variable **Z** has to equal the number of words in the DATA store. Make **Z** equal the number of words in the store after you have made your additions.

Line 50 picks a number at random between one and Z. The program then reads the DATA store down to this number. For example if the random number was 45, the computer reads down to the 45th word, FOLLOW, this word then becomes the string A$.

Line 90 works out the length of A$ and then line 110 converts the letters in the string to their ASCII codes. MID$ is then used to check the contents of the string against the letter guessed by the operator. Line 220 converts C$ the INPUT letter to its ASCII code. Lines 230–240 check to see if C$ matches a letter in A$) If it doesn't find the INPUT letter, it places the letter in the INCORRECT LETTER line on the screen (line 460). The program then returns to line 210 for another INPUT.

If the computer finds a correct letter it is placed in the correct position in the word and displayed on the screen (line 440). The program goes back to line 210 for the next INPUT. The lower case letters in lines 270, 290 and 300 print in reverse on the screen. Line 490 plays an arpeggio at the finish of a game.

```
10 REM HANGMAN
20 CLS
30 Z=73:REM Z= NUMBER OF WORDS
40 PRINT@ 417, "INCORRECT LETTERS"
50 FOR G=1 TO RND(Z)
60 READ A$
70 NEXT
```

```
80 Y=0:W=0
90 N=LEN(A$)
100 FOR G=1 TO N
110 B(G)=ASC(MID$(A$,G,1))
120 D(G)=B(G)
130 NEXT
140 Q=N:PRINT@ 38, "YOU HAVE";Q;" GUESSE
S"
150 FOR J=1 TO Q:Y=Y+1
160 GOSUB360
170 IF H=N THEN 290
180 PRINT@ 262,"YOU HAVE"; Q+1-J;" GUESS
ES LEFT"
190 R=0
200 SOUND 150,5
210 PRINT@ 353,"NEXT GUESS";:INPUT C$
220 F=ASC(C$)
230 FOR G=1 TO N
240 IF D(G)=F THEN D(G)=0:J=J-1
250 NEXT:NEXT
260 GOSUB 360
270 PRINT@ 417, "sorry time is up the wo
rd was                "
280 PRINT A$:GOTO 490
290 PRINT@ 417, TAB(5);"well done"
300 PRINT "you guessed the word in ";Y-1
:GOTO 490
310 GOTO310
320 DATA"TERROR","HORROR","POSTURE","ELE
PHANT","TRIUMPHANT","STATUS","BACHELOR",
"ANSWER"
```

```
330 DATA"TENOR","FRANTIC","TERRIER","BAN
ANA","FIGURE","IDIOT","NARCOTIC","PATHET
IC"
340 DATA"WIZARD","LIZARD","WICKED","WIZE
NED","EVIL","WEEVIL","WHEELS","BLIZZARD"
350 DATA"PARTICLE","ATOM","ARTICLE","ELE
CTRON","STARTED","PARTED","FAMISHED","EA
GLE","LEGAL"
360 DATA"WATER","WANTED","WAITED","WAITE
R","MINISTER","SINISTER","FINISHED"
370 R=H
380 DATA"PERSONAL","ASSUME","PROGRAM","P
OSTAGE","FOLLOW","WALLOW","CHALICE","MAL
ICE","SCHEME","DISASTER","REACH","SEARCH
","EVER"
390 DATA"POWER","LOWER","FLOWER","DOUBLE
","TROUBLE","LIGHTLY","PLIGHT","FLIGHT",
"SIGHTED","FOUGHT","BOUGHT","SOUGHT","TR
OUGH"
400 DATA"COUGH","BOUGH","DRAUGHT","CAUGH
T","WELCOME","CHECKED","ASSIGN","SHRUGGE
D","PATRONISE","RECORD","DECODE","MIDDLE
","FIDDLE"
410 H=0
420 FOR E=1 TO N
430 IF B(E)=D(E) THEN PRINT@ 138+E, "-";
440 IF B(E)<>D(E) THEN PRINT@ 138+E, CHR
$(B(E));:H=H+1
450 NEXT
```

```
460 IF R=H THEN W=W+1:PRINT@434+W,CHR$(F
)
470 IF H<>N THEN PRINT@ 198,"YOU HAVE";H
;" CORRECT"
480 RETURN
490 PLAY "L4;T4;O2;A;O3;L8;A;C;E;C;A;D;E
;O3;A;E;O;L4;E;A;D;L8;A;D;E;O;G;A;"
```

CAT AND MOUSE

This is your big opportunity to be an ecologist. The game Cat and Mouse requires you to set up a population of cats and mice on a desert island in such a way that it becomes self-supporting. Too many cats and the mice won't be able to breed fast enough to feed them. Too many mice and they will overrun the island and crowd out the cats.

The computer will give you a month by month read-out of the island's population. Each time the program is RUN the number of mice required to feed each cat is selected randomly in line 80.

The program also has a HIGH SCORE feature which will tell you, at the end of each game, your best effort so far. This of course returns to zero each time the program is RUN.

Line 550 provides the screen background colour for each read out. Lines 120, 190 and 330 provide the sound effects. The extra spaces in lines 40, 240 and 490 (following "or") are to space the PRINT statements neatly on the screen.

The lower case letters in line 530 are achieved by using shift and zero together. This then prints on the screen in reverse.

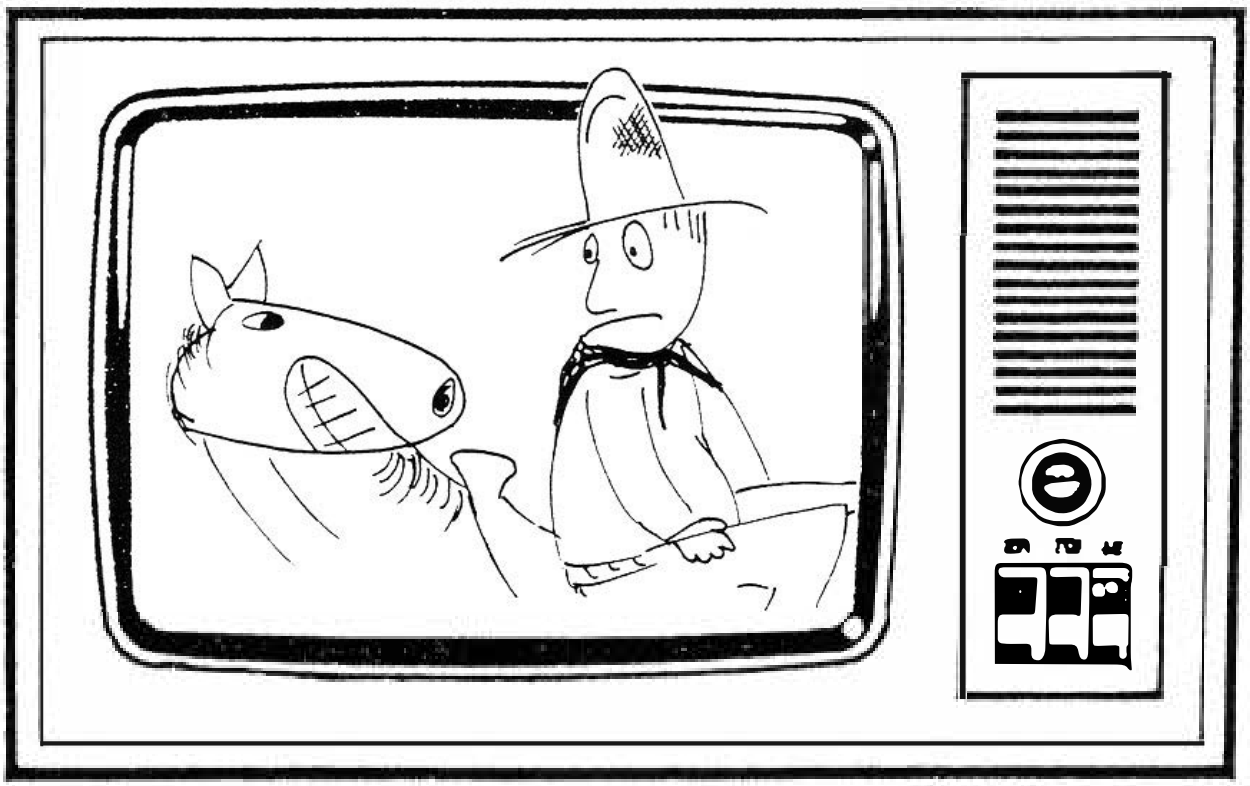
```
10 REM CAT AND MOUSE
20 CLS
30 PRINT "WELCOME TO CAT AND MOUSE"
40 PRINT "THE OBJECT OF THE GAME IS TO
  CREATE A POPULATION OF CATS AND MICE W
HICH WILL SURVIVE FOR AS  LONG AS POSSIB
LE ON A DESERT    ISLAND"
50 PRINT "PRESS ENTER TO CONTINUE"
60 INPUT A$:IF A$=" " THEN 60
70 HI=0
```

```
80 FD=RND(0)
90 CLS
100 PRINT "HOW MANY CATS WILL YOU START
   WITH (1-99)?"
110 INPUT CP:IF CP>99 THEN 110
120 SOUND 100,1:SOUND 150,1:SOUND 100,1
130 PRINT "CATS:"
140 PRINT "POPULATION:"CP
150 CP=CP/3
160 PRINT "HOW MANY MICE ARE ON THE ISLA
ND (1-99)?"
170 INPUT MP
180 IF MP>99 THEN 170
190 SOUND 100,1:SOUND 150,1:SOUND 100,1
200 PRINT "MICE:"
210 PRINT "POPULATION:";MP
220 FOR X=1 TO 800:NEXT X:CLS
230 MP=MP/3
240 PRINT "I AM GETTING YOUR CATS AND MI
CE          PLEASE STAND BY"
250 GOSUB 550
260 DA=0
270 DA=DA+1
280 PRINT@ 224, "MONTH";DA:IF DA=1 THEN
300
290 PRINT@256," "
300 IF CP>MP/FD THEN CP=MP/FD
310 CP=ABS(CP+((.8*CP-CP*MP/3)*FD))
320 MP=ABS(MP+((.4*MP-MP*CP)*.01))
330 SOUND 150,1:SOUND 100,1:SOUND 150,1
340 PRINT@ 234,  INT(CP);"CATS"
```

```
350 IF INT(MP)=1 THEN PRINTTAB(10)INT(MP
);"MOUSE":GOTO 440
360 PRINT TAB(10)INT(MP);"MICE"
370 GOSUB 550
380 IF CP<2 OR MP<2 GOTO 400
390 GOTO 270
400 IF MP<2 AND CP<2 GOSUB 550 ELSE GOTO
 420
410 PRINT@ 224, "WE HAVE RUN OUT OF CATS
 AND MICE":GOTO 460
420 IF CP<2 AND MP>2 THEN GOSUB 550
430 PRINT@ 224, "WE HAVE RUN OUT OF CATS
":GOTO 460
440 IF MP<2 AND CP>2 THEN GOSUB 550
450 PRINT@ 224, "WE HAVE RUN OUT OF MICE
"
460 PRINT@ 256, "THE POPULATION OF CATS
AND MICE SURVIVED FOR ";DA;"MONTHS"
470 IF DA>HS THEN HS=DA
480 FOR X=1 TO 200:NEXT X:PRINT "THE LON
GEST SO FAR IS";HS
490 PRINT "PRESS (Y) FOR A NEW ISLAND OR
   (ENTER) TO STOP"
500 INPUT A$:IF A$=" " THEN 500
510 IF A$="Y" GOTO 90
520 CLS
530 PRINT@ 224,TAB(11)"goodbye"
540 END
550 FOR X=1 TO 1500:NEXT X:CLS RND(8)
560 FOR X=1 TO 200:NEXT X:RETURN
570 PRINTRND(0):GOTO570
```

REVERSE

In this program the computer displays the numbers from zero to nine in a random order. Your task is to unscramble the numbers and put them into the sequence 0 1 2 3 4 5 6 7 8 9 in as few moves as possible. This is done by either reversing the whole sequence or just a part of it. For example the computer generates 7354098621. It will then ask you "REVERSE No.?". Entering 1 would reverse the entire



sequence to read 1268904537. Entering 5 would then reverse the numbers from the fifth number onwards. That is 1268735409. The game finishes when you put the numbers into order from 0 to 9.

The program performs this clever little juggling trick by placing the randomly selected numbers in a character string (A$). The number is selected in line 50. Line 70 checks the number to make sure it hasn't been selected before and then places it in the string.

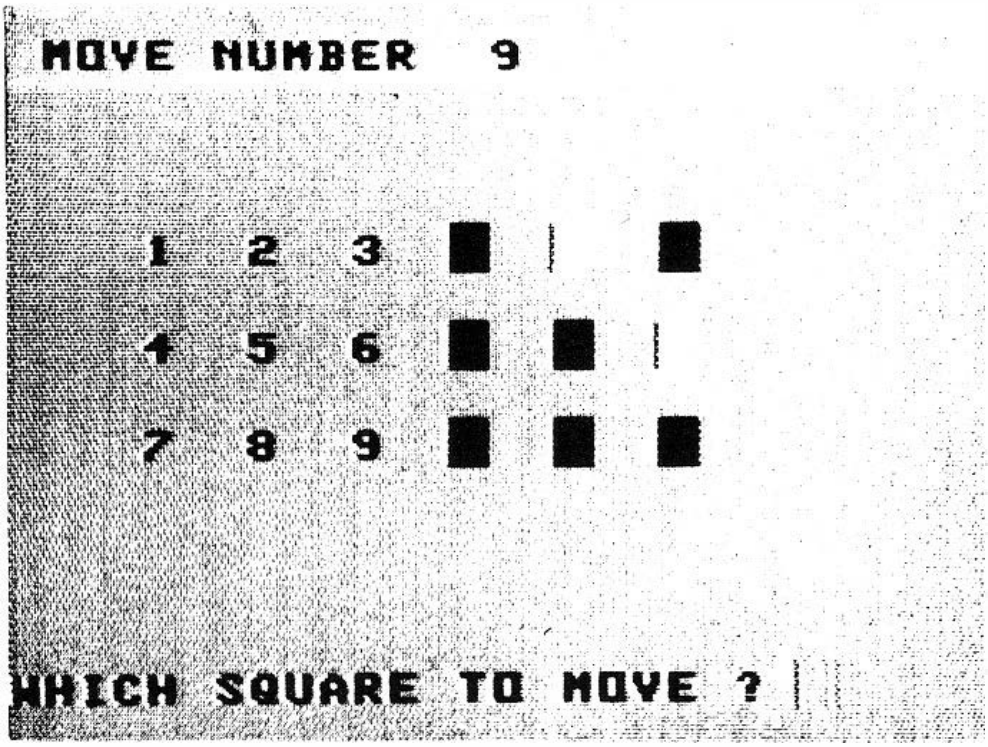
The function CHR$ converts the number to its ASCII code so that it can be stored in a string. The program then uses MID$ to manipulate the numbers within the string after each input. Lines 110 and 260 play a little fanfare at suitable times during thee game. Feel free to alter these lines to play any tune you like.

```
10 REM REVERSE
20 CLS
30 M=1:X=0:A$=""
40 FOR T=0 TO 9
50 L=RND(10)+47
60 Q=1
70 IF MID$(A$,Q,1)=CHR$(L)THEN 50
80 IFQ<T THEN Q=Q+1:GOTO 70
90 A$=A$+CHR$(L)
100 NEXT
110 CLS RND(8):SOUND150,1:SOUND200,1:SOU
ND150,1
120 PRINT@224,"MOVE NO.";M;":  ";:PRINT
A$
130 PRINT" ":PRINT@256,"REVERSE NUMBER "
;
140 INPUT R:IF R<1 OR R>9 THEN 140
150 B$=""
160 FOR T= 10 TO R STEP-1
170 B$=B$+MID$(A$,T,1)
180 NEXT T
```

```
190 A$=LEFT$(A$,R-1)+B$
200 IF A$="0123456789" THEN 220
210 M=M+1:GOTO 110
220 CLS3
230 PRINT@192,TAB(11)A$
240 PRINT@ 224, TAB(10) "YOU DID IT!!"
250 PRINT TAB(8) "IT TOOK";M;"MOVES"
260 SOUND50,1:SOUND100,1:SOUND150,1:SOUN
D200,1:SOUND150,1:SOUND100,1:SOUND50,1
```

FLIP FLOP

Flip Flop is a simple little puzzle which can cause a great deal of thought and frustration. The computer places nine squares on a three by three grid.



These squares are randomly coloured black or white. The object of the game is to finish with a white square in the centre (position 5) and black squares in the eight outside positions. This is achieved by 'flipping' black squares to make white squares. You are not allowed to flip over white squares.

Just to make it more interesting, when you flip a corner square the computer flips the three adjoining squares. For example, if you flip square one, the computer will flip squares two, four and five as well. Flipping the square in the centre of

one of the four outside rows also flips the two squares next to it. Flipping the centre square results in squares two, four, six and eight flipping as well.

The computer will ask you which square you wish to flip and will also keep a running total of the number of moves you have made. When you get the squares in the correct order the computer will reward your patience with a little fanfare and a graphics display. (Lines 450 to 630)

If you enter an illegal number (zero or a number greater than nine) the whole display will scroll up the screen causing an unreadable scramble. Fear not it can be saved. Just hit the CLEAR key and then enter a legal number, the display will return. However, you will be penalised one move.

In the program Lines 30–70 select the graphic characters for the beginning of the game. Line 300 sets Q as a black square (code 128)) and X as a white square (code 207). Actually 207 is the code for a BUFF square but it shows up as white on the green background. Lines 50 and 60 randomly decide if a square is white or black.

Lines 230–250 then uses CHR$ to print the squares on the screen. Note that in this lines PRINT@ is used. This to make sure that the squares remain in the same place and don't scroll up the screen after each input.

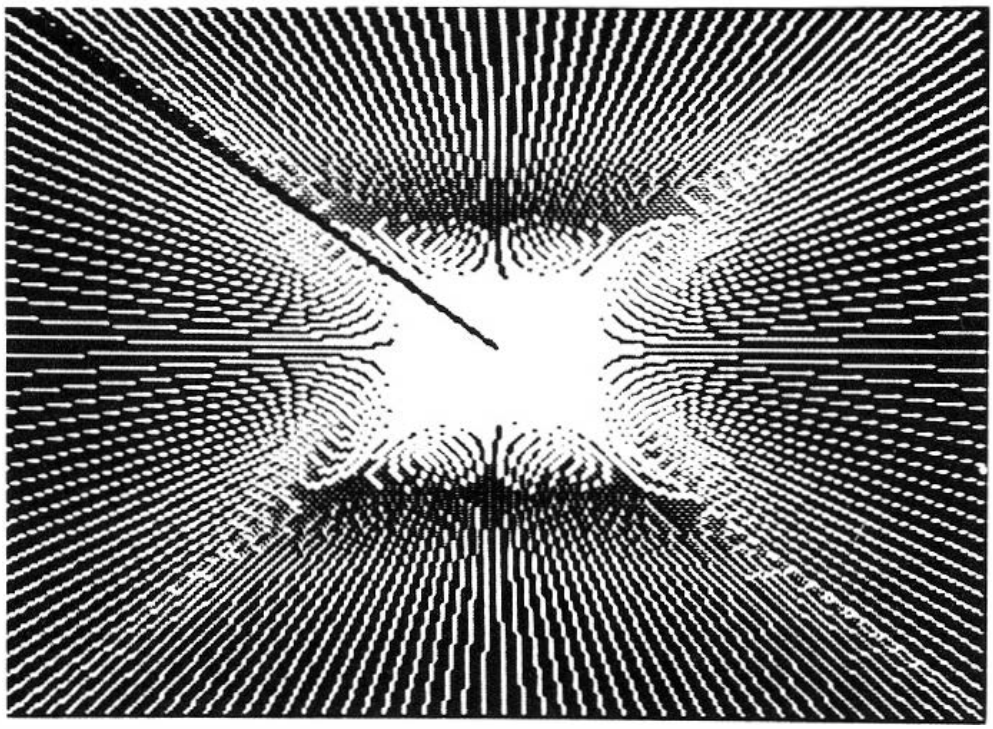
Lines 300 to 380 set out the parameters of the game. After each input the computer checks through these lines to see which other squares it is required to flip.

```
20 CLS
30 M=0:Q=128:X=207
40 FOR C=1 TO 9
```

```
50 B=RND(2)
60 A(C)=Q:IF B=1 THEN A(C)=X
70 NEXT C
80 GOSUB 220
90 N=0
100 FOR C=1 TO 9
110 IF A(C)=X THEN N=N+1
120 NEXT C
130 IF N=1 AND A(5)=X THEN 270
140 M=M+1
150 PRINT@ 33, "MOVE NUMBER ";M
160 PRINT@ 448, "WHICH SQUARE TO MOVE ";
170 INPUT A$:IF A$="" GOTO 170
180 N=VAL(A$):IF N<1 OR N>9 THEN170
190 GOSUB 290
200 GOTO 80
210 END
220 SOUND 200,1:FOR W=1 TO 10:NEXT:SOUND
 200,2
230 PRINT@ 164, "1  ";"2  ";"3  ";CHR$(A
(1));"  ";CHR$(A(2));"  ";CHR$(A(3))
240 PRINT@ 228, "4  ";"5  ";"6  ";CHR$(A
(4));"  ";CHR$(A(5));"  ";CHR$(A(6))
250 PRINT@ 292, "7  ";"8  ";"9  ";CHR$(A
(7));"  ";CHR$(A(8));"  ";CHR$(A(9))
260 RETURN
270 PRINT@448,"YOU SOLVED IT IN ";M
280 GOTO 450
290 IF A(N)=X THEN RETURN
```

```
300 IF N=1 THEN F(1)=2:F(2)=4:F(3)=5:F(4
)=10
310 IF N=2 THEN F(1)=1:F(2)=3:F(3)=10:F(
4)=10
320 IF N=3 THEN F(1)=2:F(2)=5:F(3)=6:F(4
)=10
330 IF N=4 THEN F(1)=1:F(2)=7:F(3)=10:F(
4)=10
340 IF N=5 THEN F(1)=2:F(2)=4:F(3)=8:F(4
)=6
350 IF N=6 THEN F(1)=3:F(2)=9:F(3)=10:F(
4)=10
360 IF N=7 THEN F(1)=4:F(2)=5:F(3)=8:F(4
)=10
370 IF N=8 THENF(1)=7:F(2)=9:F(3)=10
380 IF N=9 THEN F(1)=8:F(2)=5:F(3)=6:F(4
)=10
390 FOR G=1 TO 4
400 IF A(F(G))=X THEN A(F(G))=Q:GOTO 420
410 IF A(F(G))=Q THEN A(F(G))=X
420 NEXT G
430 A(N)=X
440 RETURN
450 SOUND 150,3:FORY=1 TO 20:NEXT:SOUND
100,3:FOR Y=1 TO 25:NEXT
460 SOUND 175,3:FOR Y=1 TO 25:NEXT:SOUND
 150,10
470 CLS RND(9)-1
480 A=RND(32)-1
490 B=RND(16)-1
500 C=RND(9)-1
510 GOSUB 540
```

```
520 GOTO 480
530 IF C=1 OR RND(3)=1 GOTO 590
540 SET(31-A,16+B,C)
550 SET(31-A,15-B,C)
560 SET(32+A,16+B,C)
570 SET(32+A,15-B,C)
580 RETURN
590 RESET(31-A,16+B)
600 RESET(31-A,15-B)
610 RESET(32+A,16+B)
620 RESET(32+A,15-B)
630 RETURN
```



APPENDIX:

ERROR MESSAGES

CODE	EXPLANATION
/O	an attempt was made to divide a number by O
AO	an attempt was made to open a file which was already open
BS	subscript in an array was out of range
CN	can't continue; appears after editing a line in a running program
DD	an attempt was made to redimension an array
DN	device number error, an incorrect number was used with OPEN, CLOSE or PRINT
DS	there is a direct statement in a data file could be caused by a missing line number
FC	an illegal function was used
FD	bad file data
FM	bad file mode occurs if an attempt is made to INPUT into a file open for OUTPUT
ID	an illegal direct statement was used
IE	input past end of file
IO	input/output error possibly caused by trying to load from a bad tape
LS	a string is too long
NF	a NEXT has been used without a corresponding FOR
NO	file is not open
OD	out of data
OM	out of memory

OS	out of string space; use CLEAR to reserve space
OV	overflow; a number is too big for the computer to handle
RG	a RETURN was used without a matching GOSUB
SN	syntax error caused by incorrect spelling of a command, incorrect punctuation etc.
ST	string formula too long
TM	caused by assigning numeric data to a string variable or vice versa
UL	undefined line; an attempt was made to call up a line number which does not exist.

A good computer program starts with one original idea, which is nurtured and allowed to grow in its own good time.

From the moment the first keys are pressed on the Dragon 32, to the time when the final program starts to roll off the printer, programmers often find they are undergoing a process of discovery. So it was with this book. Many times, the authors found the computer was bringing its own influence to bear, helping to shape programs into their final form. In many cases, they found the Dragon appeared to have as much to do with the creative process as the programmers did, as the screen format and extensive colour and sound facilities *demanded* to be used to the full.

Robert Young, Roger Bush and Robert Shrimpton did not hurry this book. They wanted the programs to unfold, so they would be a true reflection of the capabilities of the Dragon 32. The extensive range of programs in this book suggests they approached the task in the right way. You're sure to have as much fun running and developing the games in this book as the two Roberts and Roger did when writing them.

ISBN 0 907563 26 0

£4.95

Dynamic Games for your Dragon 32

Robert Young
Roger Bush
Robert Shrimpton

INTERFACE