

DYNAMIC GAMES

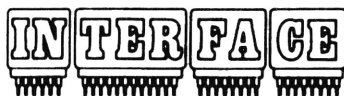
FOR YOUR T199/4A



SCOTT VINCENT

DYNAMIC GAMES FOR YOUR TI-99/4A

SCOTT VINCENT



PUBLICATIONS

44-46 Earls Court Road, London W8 6EJ

TI99IUC.it

SCOTT VINCENT

To Jim & Kath, two very dear people

First published by Interface Publications 1983
Copyright © Scott Vincent (Interface Publications) 1983

First printing September 1983

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

All rights reserved. No use whatsoever may be made of these programs except for private study by the purchaser of this volume, without the prior written permission of the copyright holder. Downloaded from www.ti99iuc.it

CONDITIONS OF SALE

This book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser.

Any enquiries regarding the contents of this book should be directed by mail to Interface Publications, 44-46 Earls Court Road, London, W8 6EJ.

ISBN 0 907563 39 2

COVER DESIGN – KEITH GIDLOW

Typeset by Twickenham Printers, Albion House, Twickenham

CONTENTS

CONTENTS

Title and Author	
Dedications and Acknowledgements	
Contents	
Introduction	
Nine Hole Golf	1
Mission Venus	5
Magic Square	10
Submarine Commander	13
Checkers	19
Gesswurd	26
Alien Encounter	30
Astornia	36
Roulette	43
Air Raid	51
Digit Brain	56
Reversi	59
Space Survival	65
Cubiks Rube	68
Life Simulation	75
Slimy Slither Supplies	82
Gold Miner	91
Hamadryad	96
Lunar Descent	101
Flippa	105
Star Patrol	109
Treasure Trail	116
Spin A Win	123
Foxes And Chickens	130
Alley Scare	134
Skittles	139
Number Juggle	144
Twenty One	147
Orchard Nim	151
Space Race	154

rights reserved. No use whatsoever is permitted without the prior written permission of the copyright owner. This work is intended for private study only. It is not to be retransmitted, reproduced, or otherwise used for general distribution, advertising, promotional purposes, for new or different advertising or promotional purposes, for creating new collective works, or for resale.

ACKNOWLEDGEMENTS

My thanks to Tim for providing some of the ideas for programs. Also many thanks to Liz and Clive for their very helpful advice, and to my family for their strong support.

INTRODUCTION

The TI-99/4A has been around for some time now but it is only recently that it has really taken off due to the massive price reduction, I presume. It is, however, a quite remarkable computer and stands up well against its supposedly more modern competitors. Special features such as a line renumbering routine and an automatic line number generator make it a delight to program and often save a considerable amount of time.

The games in this book have been carefully planned to give as much variation as possible and to give a good demonstration of the machine's capabilities.

Many hours have been spent in the preparation of this book and I am sure you will enjoy reading it and playing the games contained in it.

Finally, I must say that writing this book has given me immense pleasure and I acquired a considerable amount of knowledge myself whilst doing so.

Scott Vincent.
Ashford, Middlesex.
August, 1983.

NINE HOLE GOLF

With this program you are able to play nine hole golf with the computer telling 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 number between 1 and 99. You must always enter a two digit number therefore if you wish to enter 6, for example, you must type 06. This is because the program uses two CALL KEY commands rather than an INPUT command. The higher the number, the harder you hit the ball.

The graphic display at the beginning of the program is performed by lines 70 to 220. The special input routine, lines 450 to 570, wait for any key from 0 to 9 to be pressed. When a key is pressed the value is stored as a character in A\$. Then lines 540 and 550 make sure that the key has been released before the loop is repeated and the second key is entered. This key value is also converted to a character and added to A\$. Finally line 570 uses VAL to convert the complete string A\$ into a numeric value which is stored in the variable A. Hence there is now a two digit number stored in A which the computer uses in line 620 to decide how far the ball will travel. The distance varies by an amount dependant on the par for the hole you are playing. So if the hole has a very high par value the distance that the ball travels for a given stroke may vary quite considerably.

Lines 680 to 800 move the ball along the ground towards the hole and make it drop in the hole if the stroke was the correct strength. After all nine holes lines 1000 to 1060 tell you your hole average and allow you to play another round if you wish.

```

10 RANDOMIZE
20 CALL CHAR(128,"3C7E7E7E7E7E3CFF")
30 CALL CHAR(129,"0000000000000000FF")
40 CALL CHAR(130,"818181818181FFFF")
50 CALL CHAR(131,"BDFFFFFFFF")
60 CALL SCREEN(13)
70 CALL CLEAR
80 PRINT TAB(9);"NINE HOLE GOLF"
90 PRINT TAB(9);"~~~~~ ~~~~ ~~~~~"
100 FOR N=1 TO 10
110 PRINT
120 NEXT N
130 FOR N=1 TO 400
140 NEXT N
150 CALL HCHAR(11,32,128,1)
160 FOR N=32 TO 18 STEP -1
170 CALL HCHAR(11,N,32,1)
180 CALL HCHAR(11,N-1,128,1)
190 NEXT N
200 CALL HCHAR(11,17,32,1)
210 CALL HCHAR(12,17,128,1)
220 CALL SOUND(100,587,0)
230 FOR N=1 TO 500
240 NEXT N
250 C=0
260 FOR Z=1 TO 9
270 J=20+INT(RND*13)
280 SC=0
290 Q=3+INT(RND*4)
300 CALL CLEAR
310 PRINT TAB(6);"PAR FOR THIS HOLE:";Q-
1
320 PRINT :TAB(9);"HOLE NUMBER";Z
330 FOR N=1 TO 20
340 PRINT
350 NEXT N
360 CALL HCHAR(11,1,129,32)
370 CALL HCHAR(11,8,32,1)
380 CALL HCHAR(12,8,130,1)
390 CALL HCHAR(11,J,128,1)

```

```

400 A$="STROKE?"
410 M=20
420 K=12
430 CALL SOUND(1000,110,0)
440 GOSUB 1070
450 A$=""
460 FOR N=1 TO 2
470 CALL KEY(3,M,K)
480 IF (M<48)+(M>57)<>0 THEN 470
490 CALL SOUND(100,220,0)
500 A$=A$&CHR$(M)
510 M=20
520 K=20
530 GOSUB 1070
540 CALL KEY(3,M,K)
550 IF K<>0 THEN 540
560 NEXT N
570 A=VAL(A$)
580 X=-1
590 IF J>8 THEN 610
600 X=1
610 K=J
620 J=J+X*INT(A/(1+INT(RND*Q)))
630 IF J<33 THEN 650
640 J=32
650 IF J>2 THEN 670
660 J=3
670 SC=SC+1
680 FOR N=K TO J-X STEP X
690 CALL HCHAR(11,N,129,1)
700 IF N<>8 THEN 730
710 CALL SOUND(100,-7,0)
720 CALL HCHAR(11,8,32,1)
730 CALL HCHAR(11,N+X,128,1)
740 CALL SOUND(100,-3,0)
750 NEXT N
760 IF J<>8 THEN 800
770 CALL HCHAR(11,8,32,1)
780 CALL HCHAR(12,8,131,1)
790 CALL SOUND(100,587,0)

```

```

800 CALL HCHAR(20,13,32,10)
810 IF SC>1 THEN 860
820 A$="SCORE FOR THIS HOLE="
830 M=23
840 K=6
850 GOSUB 1070
860 A$=STR$(SC)
870 M=23
880 K=26
890 GOSUB 1070
900 IF J<>8 THEN 400
910 FOR N=1 TO 500
920 NEXT N
930 C=C+SC
940 CALL CLEAR
950 PRINT TAB(4);"YOU DID IT IN";SC;"STR
OKES"
960 PRINT ::TAB(4);"SCORE FOR";Z;"HOLES
IS";C
970 FOR N=1 TO 800
980 NEXT N
990 NEXT Z
1000 PRINT ::TAB(7);"END OF THAT ROUND"
1010 AV=INT((C*10)/9)/10
1020 PRINT :TAB(6);"YOUR AVERAGE WAS";AV
1030 PRINT :::
1040 INPUT " ANOTHER ROUND? (YES/NO) ":T
$
1050 IF T$<>"NO" THEN 70
1060 STOP
1070 FOR X=1 TO LEN(A$)
1080 CALL HCHAR(M,K+X,ASC(SEG$(A$,X,1)),
1)
1090 NEXT X
1100 RETURN

```

MISSION VENUS

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 on 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 or equal to five. When your craft gets below 100 metres the radar screen changes to give you a larger scale for the final approach. A good landing is rewarded with a score based upon 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.

Initially, your height will be 500 metres, your rate of fall 15, and you will have 200 units of fuel. To slow your ship down you must apply some thrust by pressing any key from 1 to 9. Key 9 will apply maximum thrust and key 1 will apply only a small amount of thrust. If you apply too much thrust your ship may stop descending and will stay at the same height until the planet's gravity starts to make it fall again.

The main part of the program is from lines 450 to 980 and this controls the printout of your ship on the screen and updates the information displayed on the screen such as the amount of fuel left and the present height of the craft. It also scans the keyboard and alters the rate of descent accordingly if you are pressing a key from 1 to 9. Variable K controls the ship's distance across the screen, variable H is the height, V is the rate of descent, and F is the amount of fuel you have left.

```

10 RANDOMIZE
20 CALL CLEAR
30 CALL CHAR(128,"01071F36361F1060")
40 CALL CHAR(129,"80E0F86C6CF80806")
50 CALL CHAR(136,"000000E0FBFFFFFF")
60 CALL CHAR(137,"000060F0F3FFFFFF")
70 CALL CHAR(138,"000000000080C0E0")
80 CALL CHAR(139,"0000000001010307")
90 PRINT TAB(11);"PRESS ANY"
100 FOR N=1 TO 7
110 PRINT
120 NEXT N
130 PRINT TAB(3);"MISSION";TAB(15);CHR$(
128);CHR$(129);TAB(22);"VENUS"
140 FOR N=1 TO 7
150 PRINT
160 NEXT N
170 PRINT TAB(10);"KEY TO PLAY"
180 PRINT :::
190 RESTORE
200 N=1
210 READ A
220 CALL SOUND(80,A,0)
230 CALL KEY(3,K,KK)
240 N=N+1
250 IF N<13 THEN 270
260 GOTO 190
270 IF KK=0 THEN 210
280 K=3
290 F=200
300 V=15
310 H=500
320 Q=0
330 P=3
340 M=3
350 CALL CLEAR
360 CALL SCREEN(2)
370 FOR N=1 TO 8
380 CALL COLOR(N,16,1)
390 NEXT N

```

```

400 CALL COLOR(13,8,1)
410 CALL COLOR(14,14,1)
420 PRINT TAB(4);"FUEL 300";TAB(18);"HEI
GHT 500"
430 PRINT :TAB(8);"RATE OF FALL 15"
440 CALL HCHAR(20,1,136,32)
450 IF H>=100 THEN 470
460 GOTO 910
470 Z=(600-H)/27
480 K=K+RND*3-RND*2
490 IF K>3 THEN 510
500 K=3
510 IF K<30 THEN 530
520 K=30
530 CALL HCHAR(P,M,32,2)
540 CALL HCHAR(Z,K,128,1)
550 CALL HCHAR(Z,K+1,129,1)
560 P=Z
570 M=K
580 IF F=0 THEN 640
590 CALL KEY(3,Z,ZZ)
600 IF (Z<49)+(Z>57)<>0 THEN 640
610 Z=Z-48
620 CALL SOUND(-Z*100,-5,0)
630 GOTO 650
640 Z=0
650 IF Q=1 THEN 680
660 V=V+3+INT(RND*6)-Z*2
670 GOTO 690
680 V=V+1+INT((RND*3)-Z/2)
690 IF V>0 THEN 710
700 V=0
710 H=H-V
720 IF H>0 THEN 740
730 H=0
740 F=F-Z
750 IF F>0 THEN 770
760 F=0
770 A$=STR$(F)&" "
780 CALL HCHAR(21,11,ASC(SEG$(A$,1,1)),1
)

```

```

790 CALL HCHAR(21,12,ASC(SEG$(A$,2,1)),1
)
800 CALL HCHAR(21,13,ASC(SEG$(A$,3,1)),1
)
810 A$=STR$(H)&" "
820 CALL HCHAR(21,27,ASC(SEG$(A$,1,1)),1
)
830 CALL HCHAR(21,28,ASC(SEG$(A$,2,1)),1
)
840 CALL HCHAR(21,29,ASC(SEG$(A$,3,1)),1
)
850 A$=STR$(V)&" "
860 CALL HCHAR(23,23,ASC(SEG$(A$,1,1)),1
)
870 CALL HCHAR(23,24,ASC(SEG$(A$,2,1)),1
)
880 IF H>5 THEN 450
890 IF V>5 THEN 990
900 GOTO 1200
910 IF Q=1 THEN 950
920 Q=1
930 K=3
940 CALL HCHAR(20,1,137,32)
950 Z=20-H/5
960 IF Z>1 THEN 980
970 Z=1
980 GOTO 480
990 CALL HCHAR(P,M,32,2)
1000 CALL HCHAR(20,K,138,1)
1010 CALL HCHAR(20,K+1,139,1)
1020 FOR N=0 TO 30
1030 CALL COLOR(14,9,1)
1040 CALL COLOR(14,2,1)
1050 CALL SOUND(-400,-7,N)
1060 CALL COLOR(14,12,1)
1070 NEXT N
1080 CALL CLEAR
1090 PRINT TAB(5):"### CRASH LANDING ###
"

```

```

1100 PRINT ::TAB(8);"LEAVING A CRATER"
1110 PRINT :TAB(7);V*3;"METRES DEEP!"
1120 IF F>0 THEN 1140
1130 PRINT ::TAB(6);"YOU RAN OUT OF FUEL
."
1140 PRINT ::::
1150 PRINT TAB(7);"ANOTHER GO? (Y/N)"
1160 CALL KEY(3,K,KK)
1170 IF K=89 THEN 280
1180 IF K<>78 THEN 1160
1190 STOP
1200 CALL HCHAR(21,27,48,1)
1210 CALL HCHAR(P,M,32,2)
1220 CALL HCHAR(19,K,128,1)
1230 CALL HCHAR(19,K+1,129,1)
1240 FOR P=1 TO 2
1250 RESTORE
1260 FOR N=1 TO 12
1270 READ A
1280 CALL SOUND(100,A,0)
1290 NEXT N
1300 NEXT P
1310 CALL CLEAR
1320 PRINT TAB(3);"*** SUCCESSFUL LANDIN
G ***"
1330 PT=F-V*10
1340 IF PT>1 THEN 1360
1350 PT=2
1360 PRINT ::TAB(7);"SCORING";PT;"POINTS
."
1370 GOTO 1140
1380 DATA 392,349,392,294,330,294,262,44
0,262,294,392,349

```

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 by question marks.

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 100 to 290. Line 320 keeps a running total of the number of guesses you have had. This is done by incrementing variable J. Line 300 performs this function. Lines 340 to 420 print the grid on the screen.

```
10 RANDOMIZE
20 DIM A(9),B(9)
30 CALL CLEAR
40 PRINT TAB(4);"M A G I C   S Q U A R
E"
50 PRINT :::
60 FOR N=1 TO 800
70 NEXT N
80 J=0
90 M=0
100 C=1+INT(RND*9)
110 D=1+INT(RND*9)
120 IF C=D THEN 110
130 E=1+INT(RND*9)
140 IF (C=E)+(D=E)<>0 THEN 130
150 A(1)=C+D
160 A(2)=C-(D+E)
```

```

170 A(3)=C+E
180 A(4)=C-D+E
190 A(5)=C
200 A(6)=C+D-E
210 A(7)=C-E
220 A(8)=C+D+E
230 A(9)=C-D
240 FOR Z=1 TO 9
250 B(Z)=A(Z)
260 NEXT Z
270 B(C)=30
280 B(D)=30
290 B(E)=30
300 J=J+1
310 CALL CLEAR
320 PRINT TAB(10); "GUESS NO. "; J
330 PRINT ::
340 FOR Y=1 TO 7 STEP 3
350 FOR Z=0 TO 2
360 IF B(Y+Z)=30 THEN 390
370 PRINT TAB(10+Z*5); B(Y+Z);
380 GOTO 400
390 PRINT TAB(10+Z*5); "??";
400 NEXT Z
410 PRINT ::::
420 NEXT Y
430 IF M=9 THEN 550
440 PRINT TAB(7); "YOU HAVE"; M; "RIGHT."
450 PRINT :TAB(10); "WHAT NUMBER?"
460 INPUT U
470 M=0
480 FOR Z=1 TO 9
490 IF U<>A(Z) THEN 510
500 B(Z)=U
510 IF B(Z)=30 THEN 530
520 M=M+1
530 NEXT Z
540 GOTO 300
550 PRINT ::

```

```
560 PRINT :TAB(6);"YOU HAVE SOLVED IT."  
570 PRINT ::  
580 FOR P=1 TO 4  
590 X=P*50+P  
600 FOR C=110 TO 1010 STEP X  
610 CALL SOUND(100,C,0)  
620 NEXT C  
630 Y=250-P*50-P  
640 FOR C=1010 TO 110 STEP -Y  
650 CALL SOUND(100,C,0)  
660 NEXT C  
670 NEXT P  
680 STOP
```

SUBMARINE COMMANDER

You are the captain of a nuclear submarine on patrol in the Atlantic Ocean. Your task is to track down the enemy submarine and destroy it with your torpedoes. Your torpedoes are only effective at close range and you must hit the enemy twice to destroy it. Watch out for the enemy's return fire and do not get too close or there may be a disastrous underwater collision.

Your fuel consumption increases when you fire your torpedoes or you are hit by the enemy. The game ends when you either run out of fuel, run out of time or manage to sink the enemy submarine.

You will receive constantly updated reports on your position, the amount of energy you have left, the time left and the rough position of the enemy. You will also be warned when your energy is low and when you are under attack.

```
10 RANDOMIZE
20 CALL CLEAR
30 GOSUB 1360
40 CALL CLEAR
50 GOSUB 1260
60 GOSUB 970
70 IF L<0 THEN 710
80 PRINT :TAB(4);"WHAT IS YOUR ORDER. CAPTAIN?"
90 L=L-.25
100 PRINT :TAB(3);"NORTH, SOUTH, EAST, WEST,"
110 PRINT :TAB(6);"UP, DOWN OR TORPEDO"
120 CALL KEY(3,J,JJ)
```

```

130 IF (J=78)+(J=83)+(J=69)+(J=87)+(J=85
)+(J=68)+(J=84)=0 THEN 120
140 IF J<>84 THEN 160
150 GOSUB 370
160 X=X+(J=78)-(J=83)
170 X=X+(X>10)-(X<1)
180 Y=Y+(J=87)-(J=69)
190 Y=Y+(Y>10)-(Y<1)
200 Z=Z+(J=85)-(J=68)
210 Z=Z+(Z>10)-(Z<1)
220 IF T=2 THEN 1450
230 GOSUB 800
240 J=INT(RND*2)
250 A=A+INT(RND*3)*(J=0)-INT(RND*3)*(J=1
)
260 IF (A<1)+(A>10)=0 THEN 280
270 A=1+9*(A>10)
280 J=INT(RND*2)
290 B=B+INT(RND*3)*(J=0)-INT(RND*3)*(J=1
)
300 IF (B<1)+(B>10)=0 THEN 320
310 B=1+9*(B>10)
320 J=INT(RND*2)
330 C=C+INT(RND*3)*(J=0)-INT(RND*3)*(J=1
)
340 IF (C<1)+(C>10)=0 THEN 360
350 C=1+9*(C>10)
360 GOTO 60
370 L=L-.75
380 IF (ABS(A-X)>3)+(ABS(B-Y)>3)+(ABS(C-
Z)>3)=0 THEN 430
390 PRINT ::TAB(5);"\\\\" OUT OF RANGE //
//"
400 CALL SOUND(1000,110,0,117,0)
410 CALL SOUND(100,523,0)
420 RETURN
430 PRINT ::TAB(4);"ORDER TO FIRE UNDERS
TOOD":TAB(12);"CAPTAIN"
440 FOR J=110 TO 1000 STEP 10
450 CALL SOUND(-500,J,0,1110-J,0)

```

```

460 NEXT J
470 IF RND>.5 THEN 580
480 PRINT :TAB(5);">>>> YOU MISSED!!
<"
490 FOR J=1 TO 20
500 N=110+RND*400
510 O=510+RND*400
520 P=910+RND*400
530 CALL SOUND(50,N,0)
540 CALL SOUND(50,O,0)
550 CALL SOUND(50,P,0)
560 NEXT J
570 RETURN
580 PRINT :TAB(2);"*+*+* YOU DID IT, SIR
  *+*+*"
590 FOR J=1 TO 12
600 PRINT
610 NEXT J
620 T=T+1
630 RETURN
640 CALL CLEAR
650 PRINT ::TAB(10);"END OF GAME"
660 IF TI<>0 THEN 680
670 PRINT ::TAB(3);"YOU STAYED UNDER TOO
  LONG!"
680 IF L<=0 THEN 700
690 PRINT ::TAB(4);"YOU HAVE BEEN DEFEAT
  ED!"
700 IF L<>0 THEN 720
710 PRINT ::TAB(8);"ALL ENERGY USED"
720 PRINT ::::
730 GOTO 1470
740 PRINT ::TAB(4);"YOU HAVE COLLIDED WI
  TH"
750 PRINT :TAB(5);"THE ENEMY SUBMARINE!!
  "
760 FOR V=0 TO 30
770 CALL SOUND(200,-7,V)
780 NEXT V
790 GOTO 1470

```

```

800 IF (ABS(A-X)>3)+(ABS(B-Y)>3)+(ABS(C-
Z)>3)+(RND>.75)<>0 THEN 920
810 PRINT :TAB(8); "*** WARNING ***"
820 PRINT :TAB(5); "ENEMY IS FIRING AT US
"
830 FOR V=-5 TO -7 STEP -1
840 CALL SOUND(1000,V,0)
850 NEXT V
860 IF RND>.5 THEN 930
870 PRINT :TAB(2); "~~~~~ THE ENEMY HAS HI
T ~~~~~"
880 PRINT :TAB(7); "~~~~~ US, SIR ~~~~~"
890 L=L-7
900 FOR J=1 TO 600
910 NEXT J
920 RETURN
930 PRINT :TAB(8); "THE ENEMY MISSED"
940 FOR J=1 TO 600
950 NEXT J
960 RETURN
970 CALL CLEAR
980 PRINT :TAB(8); "ENERGY REMAINING":TAB
(11);L;"ERGS"
990 TI=TI-1
1000 IF TI=0 THEN 640
1010 PRINT :TAB(12); "TIME: ";TI
1020 CALL SOUND(200,300-TI-2*L,0)
1030 IF L>3 THEN 1050
1040 PRINT :TAB(6); "ENERGY LEVEL IS LOW"
1050 PRINT :TAB(12); "TALLY: ";T
1060 PRINT ::TAB(6); "YOU ARE AT";X;",";Y
;",";Z
1070 IF (A=X)+(B=Y)+(C=Z)=-3 THEN 740
1080 PRINT :TAB(10); "THE ENEMY IS"
1090 IF (A<>X)+(B<>Y)=0 THEN 1210
1100 PRINT :TAB(9); "TO THE:";
1110 IF A>=X THEN 1130
1120 PRINT "NORTH";
1130 IF A<=X THEN 1150
1140 PRINT "SOUTH";

```

```

1150 IF B<=Y THEN 1170
1160 PRINT "EAST"
1170 IF B>=Y THEN 1190
1180 PRINT "WEST"
1190 IF C<>Z THEN 1210
1200 PRINT :TAB(13);"OF YOU"
1210 IF C<=Z THEN 1230
1220 PRINT :TAB(11);"BELOW YOU"
1230 IF C>=Z THEN 1250
1240 PRINT :TAB(11);"ABOVE YOU"
1250 RETURN
1260 L=26+INT(RND*30)
1270 T=0
1280 TI=35
1290 A=1+INT(RND*10)
1300 B=1+INT(RND*10)
1310 C=1+INT(RND*10)
1320 X=1+INT(RND*10)
1330 Y=1+INT(RND*10)
1340 Z=1+INT(RND*10)
1350 RETURN
1360 LET A$="SUBMARINE COMMANDER"
1370 FOR N=2 TO 22 STEP 4
1380 FOR Y=1 TO 19
1390 CALL HCHAR(N,Y+N/2,ASC(SEG$(A$,Y,1)
),1)
1400 NEXT Y
1410 NEXT N
1420 FOR N=1 TO 500
1430 NEXT N
1440 RETURN
1450 PRINT ::::TAB(3);"ENEMY SUBMARINE D
ESTROYED"
1460 GOSUB 1530
1470 FOR J=1 TO 400
1480 NEXT J
1490 FOR J=1 TO 25
1500 PRINT
1510 NEXT J
1520 STOP

```

```
1530 FOR J=1 TO 5
1540 FOR N=2 TO 16
1550 CALL SCREEN(N)
1560 NEXT N
1570 NEXT J
1580 CALL SCREEN(4)
1590 RETURN
```

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 board is the normal black and white checkered colour. The computer is the light red pieces at the top of the board and you are the cyan pieces 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 grid followed by the number down the side, such as G3. Then enter the location of the square you wish to move to in the same way.

The computer always has the first move. The number of pieces you have both captured is displayed below 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 is not 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 are the same shape as the other pieces but have a black mark in the middle of them. The computer will reject any illegal moves.

Lines 1480 to 1590 draw the checkerboard pattern on the screen at the start of the game. Lines 180 to 560 control the computer's moves and lines 570 to 1010 control the player's moves and make sure that they are legal. Lines 1760 to 2150 move the pieces around the board and remove any pieces that are jumped.

```

10 RANDOMIZE
20 CALL CHAR(128,"003C7E7E7E7E7E3C00")
30 CALL CHAR(129,"003C7E66667E7E3C00")
40 CALL CHAR(136,"003C7E7E7E7E7E3C00")
50 CALL CHAR(137,"003C7E66667E7E3C00")
60 CALL CHAR(144,"FFFFFFFFFFFFFFFF")
70 CALL CHAR(152,"FFFFFFFFFFFFFFFF")
80 DIM A(99),X(2)
90 GOSUB 1030
100 Q=0
110 Z=24
120 FOR G=69 TO 72
130 IF A(G)<>128 THEN 170
140 A(G)=129
150 CALL SOUND(100,880,0)
160 CALL HCHAR(3,159-G*2,129,1)
170 NEXT G
180 IF (A(Z)=136)+(A(Z)=137)=0 THEN 330
190 IF (Z<28)+(A(Z)=136)<>-2 THEN 230
200 A(Z)=137
210 CALL SOUND(100,880,0)
220 CALL HCHAR(10,68-Z*2,137,1)
230 Y=1
240 IF (A(Z+X(Y))<130)+(A(Z+2*X(Y))=152)
<>-2 THEN 260
250 Q=X(Y)
260 IF (A(Z)=137)+(A(Z-X(Y))<130)+(A(Z-2
*X(Y))=152)<>-3 THEN 280
270 Q=-X(Y)
280 IF (Q<>0)+(Z+2*Q>23)=-2 THEN 360
290 Q=0
300 IF Y=2 THEN 330
310 Y=2
320 GOTO 240
330 Z=Z+1
340 IF Z<73 THEN 180
350 IF Q=0 THEN 390
360 A(Z+Q)=152
370 Q=Q*2
380 GOTO 540

```

```

390 U=0
400 Z=24+INT(RND*49)
410 U=U+1
420 IF (A(Z)=136)+(A(Z)=137) THEN 450
430 IF U<1000 THEN 400
440 GOTO 2200
450 Y=1
460 IF (A(Z+X(Y))<>152) THEN 480
470 Q=X(Y)
480 IF (A(Z)=137)+(A(Z-X(Y))=152)<>-2 TH
EN 500
490 Q=-X(Y)
500 IF Q<>0 THEN 540
510 IF Y=2 THEN 430
520 Y=2
530 GOTO 460
540 A(Z+Q)=A(Z)
550 A(Z)=152
560 GOSUB 1760
570 CALL SOUND(200,300,0)
580 A$="MOVE? ___ TO ___"
590 J=16
600 GOSUB 1710
610 CALL KEY(3,KA,KK)
620 IF KA=48 THEN 2160
630 IF (KA<65)+(KA>72)<>0 THEN 610
640 CALL SOUND(100,523,0)
650 CALL HCHAR(16,17,KA,1)
660 CALL KEY(3,KB,KK)
670 IF (KB<49)+(KB>56)<>0 THEN 660
680 CALL SOUND(100,587,0)
690 CALL HCHAR(16,18,KB,1)
700 CALL KEY(3,KC,KK)
710 IF (KC<65)+(KC>72)<>0 THEN 700
720 CALL SOUND(100,659,0)
730 CALL HCHAR(16,23,KC,1)
740 CALL KEY(3,KD,KK)
750 IF (KD<49)+(KD>56)<>0 THEN 740
760 CALL SOUND(100,698,0)
770 CALL HCHAR(16,24,KD,1)

```

```

780 Z=3-(INT((KA-65)/2))
790 Q=57-KB
800 IF Q/2<>INT(Q/2)THEN 830
810 Z=Z+17+13*(Q/2)
820 GOTO 840
830 Z=Z+11+13*((Q+1)/2)
840 ZZ=3-(INT((KC-65)/2))
850 Q=57-KD
860 IF Q/2<>INT(Q/2)THEN 890
870 ZZ=ZZ+17+13*(Q/2)
880 GOTO 900
890 ZZ=ZZ+11+13*((Q+1)/2)
900 Q=ZZ-Z
910 IF A(Z)>129 THEN 570
920 IF A(ZZ)<>152 THEN 570
930 IF (A(Z)=128)+(Q<0)=-2 THEN 570
940 IF (ABS(Q)=6)+(ABS(Q)=7)<>0 THEN 980
950 IF (ABS(Q)=12)+(ABS(Q)=14)=0 THEN 57
0
960 IF (A(Z+Q/2)=136)+(A(Z+Q/2)=137)=0 T
HEN 570
970 A(Z+Q/2)=152
980 A(Z+Q)=A(Z)
990 A(Z)=152
1000 CALL HCHAR(16,11,32,14)
1010 GOSUB 1760
1020 GOTO 100
1030 CALL CLEAR
1040 CALL SCREEN(12)
1050 CALL COLOR(13,8,2)
1060 CALL COLOR(14,10,2)
1070 CALL COLOR(15,16,16)
1080 CALL COLOR(16,2,2)
1090 B$="C H E C K E R S"
1100 J=12
1110 GOSUB 1710
1120 C$=""
1130 FOR N=1 TO 11
1140 C#=C#&CHR$(144)&CHR$(152)
1150 NEXT N

```

```

1160 FOR J=9 TO 11
1170 Q=1-(INT(J/2)=J/2)
1180 A#=SEG$(C$,Q,21)
1190 GOSUB 1710
1200 NEXT J
1210 A#=SEG$(C$,2,3)&B#&SEG$(C$,2,3)
1220 J=12
1230 GOSUB 1710
1240 FOR J=13 TO 15
1250 Q=1-(INT(J/2)=J/2)
1260 A#=SEG$(C$,Q,21)
1270 GOSUB 1710
1280 NEXT J
1290 FOR N=1 TO 500
1300 NEXT N
1310 CO=0
1320 HU=0
1330 X(1)=-6
1340 X(2)=-7
1350 FOR Z=1 TO 99
1360 A(Z)=199
1370 IF (Z<73)+(Z>55)<>-2 THEN 1400
1380 IF (Z=67)+(Z=68)+(Z=60)+(Z=61)+(Z=62)<>0 THEN 1400
1390 A(Z)=136
1400 IF (Z<54)+(Z>42)<>-2 THEN 1430
1410 IF (Z=47)+(Z=48)+(Z=49)<>0 THEN 1430
0
1420 A(Z)=152
1430 IF (Z<41)+(Z>23)<>-2 THEN 1460
1440 IF (Z=34)+(Z=35)+(Z=36)+(Z=28)+(Z=29)<>0 THEN 1460
1450 A(Z)=128
1460 NEXT Z
1470 CALL CLEAR
1480 FOR N=3 TO 9 STEP 2
1490 FOR P=15 TO 21 STEP 2
1500 CALL HCHAR(N,P-1,144,1)
1510 Q=85-((N-1)/2)*13-(P-15)/2
1520 CALL HCHAR(N,P,A(Q),1)
1530 NEXT P

```

```

1540 FOR P=14 TO 20 STEP 2
1550 Q=79-((N-1)/2)*13-(P-14)/2
1560 CALL HCHAR(N+1,P,A(Q),1)
1570 CALL HCHAR(N+1,P+1,144,1)
1580 NEXT P
1590 NEXT N
1600 FOR N=3 TO 10
1610 CALL HCHAR(N,22,N+46,1)
1620 CALL HCHAR(11,11+N,N+62,1)
1630 NEXT N
1640 A$="I HAVE CAPTURED @"
1650 J=21
1660 GOSUB 1710
1670 A$="YOU HAVE CAPTURED @"
1680 J=23
1690 GOSUB 1710
1700 RETURN
1710 K=INT((33-LEN(A$))/2+.5)
1720 FOR Q=1 TO LEN(A$)
1730 CALL HCHAR(J,K+Q,ASC(SEG$(A$,Q,1)),
1)
1740 NEXT Q
1750 RETURN
1760 ZZ=Z
1770 CX=11
1780 ZZ=ZZ-6.5
1790 CX=CX-1
1800 IF ZZ>23 THEN 1780
1810 ZZ=INT(24+6.5*(10-CX))
1820 CY=21-(Z-ZZ)*2
1830 CY=CY+(INT(CX/2)=CX/2)
1840 CV=CX-SGN(Q)*(1-(ABS(Q)>7))
1850 CW=CY+(Q=7)+(Q=-6)+2*(Q=14)+2*(Q=-1
2)-(Q=6)-(Q=-7)-2*(Q=12)-2*(Q=-14)
1860 FOR N=1 TO 3
1870 CALL SOUND(500,-1,0)
1880 CALL HCHAR(CX,CY,A(Z+Q),1)
1890 CALL SOUND(500,-3,0)
1900 CALL HCHAR(CX,CY,152,1)
1910 NEXT N

```

```

1920 FOR N=1 TO 3
1930 CALL SOUND(500,-5,0)
1940 CALL HCHAR(CV,CW,152,1)
1950 CALL SOUND(500,-7,0)
1960 CALL HCHAR(CV,CW,A(Z+Q),1)
1970 NEXT N
1980 IF ABS(Q)<8 THEN 2150
1990 CX=CX+(CX>CV)-(CX<CV)
2000 CY=CY+(CY>CW)-(CY<CW)
2010 CALL SOUND(500,120,0)
2020 CALL HCHAR(CX,CY,152,1)
2030 HU=HU-(A(Z+Q)<130)
2040 CO=CO-(A(Z+Q)>135)
2050 A$=STR$(CO)&" "
2060 J=21
2070 K=24
2080 GOSUB 1720
2090 A$=STR$(HU)&" "
2100 J=23
2110 K=25
2120 GOSUB 1720
2130 IF CO=12 THEN 2160
2140 IF HU=12 THEN 2200
2150 RETURN
2160 CALL CLEAR
2170 PRINT TAB(13);"I WIN."
2180 PRINT :::
2190 STOP
2200 CALL CLEAR
2210 PRINT TAB(6);"I CONCEDE THE GAME."
2220 PRINT ::TAB(12);"YOU WIN."
2230 PRINT :::
2240 STOP

```

GESSWURD

This is the traditional word game known as hangman which involves the guessing of letters in an unknown word. The computer selects a word from its DATA store (lines 680 to 780) and displays a number of dashes on the screen corresponding to the number of letters in the word. When the prompt appears press any letter and if it is in the word it will be displayed in the correct place otherwise it will be placed in the incorrect letters list at the bottom of the screen. If you choose a letter which appears twice in the word then it will appear in two different positions on the screen. When you have found all of the letters the number of goes that you took is displayed and the computer then gives you another word to guess.

The program relies on a store of words in its DATA files. This program provides a list of 110 words. You can quite easily increase the computer's vocabulary by adding extra lines of DATA anywhere after line 780. The only other change which needs to be made to the program is in line 70. The variable Z has to equal the number of words in the DATA store.

Line 150 picks a number at random between 1 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, EAGLE, this word then becomes the string A\$.

Lines 200 to 260 work out the length of the word and display the correct number of dashes on the screen. Lines 270 to 320 then wait for a key to be pressed and make sure that it is a letter. The letter is then compared with each letter of the word in turn, lines 350 to 470, and the letters which have been correctly guessed are stored in the string B\$. Line 550 compares the complete word, A\$, with the string B\$ and if they are the same control is sent to line 570 which tells you how many goes it took, otherwise control is sent back to line 270 which waits for another input.

```

10 RANDOMIZE
20 CALL CLEAR
30 PRINT TAB(12):"GESSWURD"
40 FOR N=1 TO 22
50 PRINT
60 NEXT N
70 Z=110
80 C$="INCORRECT LETTERS:"
90 J=18
100 K=3
110 GOSUB 640
120 GE=0
130 H=18
140 I=22
150 Y=1+INT(RND*Z)
160 RESTORE
170 FOR N=1 TO Y
180 READ A$
190 NEXT N
200 Z=LEN(A$)
210 B$="_"
220 FOR N=2 TO Z
230 B#=B#&"_"
240 NEXT N
250 S=INT((34-Z)/2)
260 CALL HCHAR(9,S+1,95,Z)
270 C$="LETTER ?"
280 J=14
290 K=13
300 GOSUB 640
310 CALL KEY(3,J,K)
320 IF (J<65)+(J>90)<>0 THEN 310
330 GE=GE+1
340 CALL HCHAR(14,14,32,8)
350 LT=0
360 FOR N=1 TO Z
370 IF ASC(SEG$(A$,N,1))<>J THEN 470
380 D$=""
390 IF N=1 THEN 410
400 D$=SEG$(B$,1,N-1)

```

```

410 D$=D$&CHR$(J)
420 IF N=Z THEN 440
430 D$=D$&SEG$(B$,N+1,Z-N)
440 B$=D$
450 LT=1
460 CALL HCHAR(9,S+N,J,1)
470 NEXT N
480 IF LT=1 THEN 550
490 CALL HCHAR(H,I,J,1)
500 CALL HCHAR(H,I+1,44,1)
510 I=I+2
520 IF I<>32 THEN 550
530 I=4
540 H=H+2
550 IF A$=B$ THEN 570
560 GOTO 270
570 C$="THAT TOOK YOU "&STR$(GE)&" GOES"
580 J=14
590 K=6
600 GOSUB 640
610 FOR N=1 TO 600
620 NEXT N
630 GOTO 20
640 FOR N=1 TO LEN(C$)
650 CALL HCHAR(J,K+N,ASC(SEG$(C$,N,1)),1)
660 NEXT N
670 RETURN
680 DATA COUGH,BOUGH,DRAUGHT,CAUGHT,WELC
OME,ASSIGN,CHECKED,SHRUGGED,RECORD,PATRO
NISE,DECODE,MIDDLE,FIDDLE
690 DATA TERROR,HORROR,POSTER,ELEPHANT,T
RIUMPHANT,STATUS,ANSWER,BACHELOR
700 DATA TENOR,FRANTIC,TERRIER,BANANA,FI
GURE,IDIOT,NARCOTIC,PATHETIC
710 DATA WIZARD,LIZARD,WICKED,MISER,EVIL
,WEASEL,WHEELS,BLIZZARD
720 DATA PARTICLE,ARTICLE,ATOM,ELECTRON,
STARTED,PARTED,FAMISHED,EAGLE,LEGAL
730 DATA WATER,WANTED,WAITED,WAITER,MINI
STER,SINISTER,FINISHED

```

740 DATA PERSONAL, ASSUME, PROGRAM, POSTAGE
, FOLLOW, WALLOW, CHALICE, MALICE, SCHEME, DIS
ASTER, REACH, SEARCH, EVER
750 DATA FOWER, LOWER, FLOWER, DOUBLE, TROUB
LE, LIGHTLY, PLIGHT, FLIGHT, SIGHTED, FOUGHT,
BOUGHT, SOUGHT, TROUGH
760 DATA RHYTHM, RHYTHMICAL, RICOCHET, RIVE
T, RIVULET, GNASH, GNAW, GNAT, GNU, GNOME, QUAR
TZ, QUASH, QUEUE, QUADRILATERAL
770 DATA FYLON, PYGMY, PYTHON, XYLOPHONE, ZO
DIAC, ZEPHYR, ZOOM, BAZOOKA, BARREL, BARRON, B
ARROW
780 DATA GYRO, GYBE, GYM, GYPSY, GYRATE, FYJA
MAS

ALIEN ENCOUNTER

This is a moving graphics game which involves blasting aliens out of the sky before they reach the edge of the screen and escape. You control a laser base which is positioned at the bottom of the screen and can be moved left and right using keys S and D respectively. The aliens will take turns in trying to get from one side of the screen to the other and you must shoot them down with your laser.

To fire the laser press the space bar. Only one of your bullets can be on the screen at one time so aim accurately or you may have to wait a while before you can fire again. The game ends after eight aliens have appeared.

To make the game more interesting there is a high score feature and a very impressive display at the start. Also the number of points you score for hitting an alien depends on its distance from your base.

Lines 140 to 330 provide the display at the beginning of the program. Lines 450 to 840 are the main part of the program which connect the various subroutines together and control the aliens' movements. Finally, Lines 850 to 1160 end the game, update the high score if necessary and send control back to line 140 for a new game.

```
10 RANDOMIZE
20 CALL CHAR(128,"0018181818181800")
30 CALL CHAR(129,"183C3C7EFFFFFFFF")
40 CALL CHAR(136,"C629193C5A7E42CE")
50 CALL CHAR(137,"6394983C5A7E4273")
60 HI=0
```

```

70 CALL CLEAR
80 CALL SCREEN(5)
90 FOR N=1 TO 12
100 CALL COLOR(N,16,1)
110 NEXT N
120 CALL COLOR(13,10,1)
130 CALL COLOR(14,4,1)
140 RESTORE
150 CALL HCHAR(11,10,136,15)
160 CALL HCHAR(22,1,129,1)
170 X=2
180 FOR N=1 TO 15
190 READ A,C
200 IF A>X THEN 230
210 B=1
220 GOTO 240
230 B=-1
240 GOSUB 1170
250 X=A
260 GOSUB 1220
270 CALL SOUND(100,100+N*10,0)
280 CALL HCHAR(11,X,C,1)
290 NEXT N
300 B=-1
310 A=32
320 GOSUB 1170
330 CALL HCHAR(22,32,32,1)
340 A=16
350 B=3
360 R=21
370 SC=0
380 FR=0
390 A$="SCORE =0      HIGH SCORE ="&STR$(
HI)
400 J=24
410 K=3
420 GOSUB 1710
430 CALL HCHAR(11,10,32,15)
440 CALL HCHAR(22,A,129,1)
450 FOR N=1 TO 8

```

```

460 X=1+RND*2
470 FOR M=1 TO X
480 GOSUB 1290
490 FOR Z=1 TO 50
500 NEXT Z
510 NEXT M
520 P=1+INT(RND*20)
530 Q=137
540 IF RND>.5 THEN 700
550 CALL HCHAR(P,1,136,1)
560 M=2
570 CALL GCHAR(P,M,W)
580 SS=-110*(Q=136)-117*(Q=137)
590 CALL SOUND(150,SS,0)
600 CALL HCHAR(P,M-1,32,1)
610 CALL HCHAR(P,M,Q,1)
620 IF W=128 THEN 1610
630 Q=-136*(Q=137)-137*(Q=136)
640 GOSUB 1290
650 IF W=1 THEN 1610
660 M=M+1
670 IF M<33 THEN 570
680 CALL HCHAR(P,32,32,1)
690 GOTO 840
700 CALL HCHAR(P,32,136,1)
710 M=31
720 CALL GCHAR(P,M,W)
730 SS=-110*(Q=136)-117*(Q=137)
740 CALL SOUND(150,SS,0)
750 CALL HCHAR(P,M+1,32,1)
760 CALL HCHAR(P,M,Q,1)
770 IF W=128 THEN 1610
780 Q=-136*(Q=137)-137*(Q=136)
790 GOSUB 1290
800 IF W=1 THEN 1610
810 M=M-1
820 IF M>0 THEN 720
830 CALL HCHAR(P,1,32,1)
840 NEXT N
850 FOR N=1 TO 24
860 CALL HCHAR(N,N,136,33-N)

```

```

870 CALL VCHAR(N,N,136,25-N)
880 CALL SOUND(-300,-3,24-N)
890 NEXT N
900 FOR N=1 TO 24
910 CALL HCHAR(N,N,32,33-N)
920 CALL VCHAR(N,N,32,25-N)
930 CALL SOUND(-300,-7,24-N)
940 NEXT N
950 IF SC<=HI THEN 1050
960 A$="NEW HIGH SCORE"
970 J=8
980 K=9
990 GOSUB 1710
1000 A$=" "&STR$(SC)
1010 J=10
1020 K=14
1030 GOSUB 1710
1040 HI=SC
1050 A$="PRESS THE SPACE BAR"
1060 J=16
1070 K=6
1080 GOSUB 1710
1090 A$="TO PLAY AGAIN"
1100 J=18
1110 K=9
1120 GOSUB 1710
1130 CALL KEY(3,J,K)
1140 IF J<>32 THEN 1130
1150 CALL CLEAR
1160 GOTO 140
1170 FOR Z=X TO A STEP -B
1180 CALL HCHAR(22,Z+B,32,1)
1190 CALL HCHAR(22,Z,129,1)
1200 NEXT Z
1210 RETURN
1220 CALL SOUND(150,-5,0)
1230 CALL HCHAR(21,X,128,1)
1240 FOR Z=20 TO 11 STEP -1
1250 CALL HCHAR(Z+1,X,32,1)
1260 CALL HCHAR(Z,X,128,1)

```

```

1270 NEXT Z
1280 RETURN
1290 FOR T=1 TO 2
1300 D=A
1310 CALL KEY(3,J,K)
1320 A=A+(J=83)-(J=68)
1330 A=A+(A>30)-(A<3)
1340 CALL HCHAR(22,D,32,1)
1350 CALL HCHAR(22,A,129,1)
1360 IF (J<>32)+(FR=1) THEN 1390
1370 FR=1
1380 CALL SOUND(150,-5,0)
1390 IF FR=1 THEN 1430
1400 FOR NN=1 TO 60
1410 NEXT NN
1420 GOTO 1590
1430 IF R<21 THEN 1480
1440 R=20
1450 E=A
1460 CALL HCHAR(21,E,128,1)
1470 GOTO 1590
1480 CALL GCHAR(R,E,V)
1490 CALL HCHAR(R+1,E,32,1)
1500 CALL HCHAR(R,E,128,1)
1510 R=R-1
1520 IF (R=0)+(R=P-4)=0 THEN 1560
1530 CALL HCHAR(R+1,E,32,1)
1540 R=21
1550 FR=0
1560 IF V=32 THEN 1590
1570 T=2
1580 W=1
1590 NEXT T
1600 RETURN
1610 CALL SOUND(100,523,0)
1620 CALL HCHAR(P,M,32,1)
1630 SC=SC+(21-P)
1640 A$=STR$(SC)
1650 J=24
1660 K=10

```

```
1670 GOSUB 1710
1680 FR=0
1690 R=21
1700 GOTO 840
1710 FOR NN=1 TO LEN(A$)
1720 CALL HCHAR(J,K+NN,ASC(SEG$(A$,NN,1)
),1)
1730 NEXT NN
1740 RETURN
1750 DATA 12,73,22,84,18,67,24,82,16,69,
13,69,10,65,20,85,17,78,21,78,11,76,19,7
9,14,78,23,69,15,32
```

ASTORNIA

You are the administrator of a mining colony which has been established on an asteroid named Astornia 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. By mining the optimum amount of ore you should be able to buy sufficient food and oxygen for your colony to survive. Sometime during each year there may be a rare raid by asteroid bandits.

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.

The variables are set up in lines 100 to 220. GOSUB 750 displays the computer's report. Lines 270 to 360 allow you to mine the copper ore and then deduct certain amounts from your capital and oxygen supply. Lines 390 to 530 allow you to buy food, adding the food bought to the present stock and subtracting the cost from your capital. Lines 560 to 640 allow you to buy oxygen and tell you how many years the current

stock will last for. Lines 660 to 740 subtract the annual maintenance from your capital, the food needs from your food stocks and the oxygen needs from your oxygen supply. The program then jumps back to line 240 where the next year begins and the whole process starts again.

The program is ended one way or another by lines 1050 to 1330. The last section of the program, lines 1340 to 1710 is a subroutine which decides what has attacked your colony and what damage has been caused. The attackers can range from a fleet of Syrian ships to Vyrilliex outworlders and as well as stealing food, oxygen and money they will usually kill a few Astornians.

```
10 RANDOMIZE
20 CALL CLEAR
30 PRINT TAB(8);"A S T O R N I A"
40 PRINT TAB(8);"===== "
50 PRINT :::::
60 FOR N=1 TO 800
70 NEXT N
80 CALL SCREEN(11)
90 CALL SOUND(500,-1,0)
100 YR=0
110 A$="THE COLONY HAS PERISHED"
120 FK=80+INT(RND*40)
130 CA=INT(7*(700+RND*800)/1+INT(RND*3))
140 FC=1+INT(RND*7)
150 AC=2+INT(RND*3)
160 FD=2000+INT(RND*500)
170 OX=2000-INT(RND*1000)
180 OC=1+INT(RND*7)
190 AP=AC*OC*2-INT(RND*OC)
200 RP=200+INT(RND*400)
210 FE=2+INT(RND*5)
220 OD=3+INT(RND*3)
230 CALL CLEAR
240 YR=YR+1
250 FK=FK+FK/(3+INT(RND*18))-FK/(4+INT(RND*15))
```

```

260 GOSUB 750
270 PRINT :TAB(3);"COPPER ORE - HOW MAN
Y TONS"
280 PRINT :TAB(4);"WILL YOU MINE AND SEL
L?"
290 PRINT :TAB(5);"THEY USE UP";AC;"UNIT
S OF"
300 PRINT :TAB(4);"OXYGEN AND SELL FOR $
";STR$(AP)
310 INPUT B
320 IF B*AC<=OX THEN 350
330 PRINT :TAB(7);"NOT ENOUGH OXYGEN."
340 GOTO 310
350 CA=CA+B*AP
360 OX=OX-B*AC
370 CALL CLEAR
380 GOSUB 750
390 PRINT :TAB(5);"FOOD COSTS $";STR$(F
C);" PER UNIT."
400 PRINT :TAB(6);"EACH PERSON NEEDS";FE
410 PRINT :TAB(10);"FOOD UNITS."
420 PRINT :TAB(5);"$";STR$(INT(FC*FE));"
EACH, $";STR$(INT(FC*FE*FK));" FOR"
430 PRINT :TAB(12);"STATION."
440 PRINT :TAB(5);"THIS WILL LAST";INT(FD/(FE*F
K));"YEARS WITH"
450 PRINT :TAB(4);"THE PRESENT POPULATIO
N."
460 PRINT :TAB(6);"HOW MANY FOOD UNITS"
470 PRINT :TAB(9);"WILL YOU BUY?"
480 INPUT C
490 IF C*FC<=CA THEN 520
500 PRINT :TAB(7);"NOT ENOUGH MONEY."
510 GOTO 480
520 FD=FD+C
530 CA=CA-C*FC
540 CALL CLEAR
550 GOSUB 750
560 PRINT :TAB(8);"HOW MUCH OXYGEN"
570 PRINT :TAB(9);"WILL YOU BUY?"

```

```

580 PRINT : "CURRENT STOCKS WILL LAST FOR
"
590 PRINT : TAB(3); INT(OX/(OD*FK)); " YEAR
S WITH THE PRESENT"
600 PRINT : TAB(10); "POPULATION."
610 INPUT D
620 IF D*OC<=CA THEN 650
630 PRINT : TAB(7); "NOT ENOUGH MONEY."
640 GOTO 610
650 CALL CLEAR
660 IF RND<.8 THEN 680
670 GOSUB 1340
680 FD=FD-INT(FE*FK)
690 CA=CA-RP-D*OC
700 OX=OX+D-OD*FK
710 CALL SOUND(200,230,10)
720 CALL SOUND(200,240,5)
730 CALL SOUND(200,250,0)
740 GOTO 240
750 PRINT TAB(7); "COMPUTERS REPORT."
760 PRINT TAB(7); "-----"
770 PRINT ::
780 IF OX<OD*FK THEN 1050
790 IF FD<FE*FK THEN 1090
800 IF CA<1 THEN 1130
810 IF FK<10 THEN 1170
820 IF FK>20 THEN 850
830 PRINT TAB(4); "WARNING - POPULATION I
S"
840 PRINT : TAB(6); "NEARING EXTINCTION."
850 IF OX>2*OD*FK THEN 880
860 PRINT : TAB(4); "WARNING - OXYGEN SUPP
LY"
870 PRINT : TAB(12); "IS LOW."
880 IF FD>2*FE*FK THEN 900
890 PRINT : TAB(3); "WARNING - FOOD STOCKS
LOW."
900 IF CA>2000 THEN 920
910 PRINT : "WARNING - MONEY RUNNING LOW.
"

```

```

920 PRINT :TAB(4); "THERE ARE"; INT(FK); "P
EOPLE ON"
930 PRINT :TAB(6); "ASTORNIA IN YEAR"; YR
940 PRINT :TAB(5); "MONEY CREDIT IS $"; ST
R$(INT(CA))
950 PRINT :TAB(4); "ANNUAL MAINTENANCE $"
; STR$(RP)
960 PRINT :TAB(7); "OXYGEN TANKS HOLD"
970 PRINT :TAB(11); INT(OX); "UNITS."
980 PRINT :TAB(2); "OXYGEN COSTS $"; STR$(
OC); " PER UNIT."
990 PRINT :TAB(3); "OXYGEN NEED PER PERSO
N: "; OD
1000 PRINT :TAB(4); "FOOD STOCKS STAND AT
"; FD
1010 PRINT ::TAB(9); "PRESS ANY KEY."
1020 CALL KEY(3,X,Y)
1030 IF Y=0 THEN 1020
1040 RETURN
1050 PRINT TAB(4); A$
1060 PRINT :TAB(5); "YOU RAN OUT OF OXYGE
N"
1070 PRINT :TAB(11); "IN YEAR"; YR
1080 GOTO 1280
1090 PRINT TAB(4); A$
1100 PRINT : "FOOD SUPPLIES WERE EXHAUSTE
D"
1110 PRINT :TAB(11); "IN YEAR"; YR
1120 GOTO 1280
1130 PRINT TAB(4); A$
1140 PRINT :TAB(2); "THE TREASURY RAN DRY
DURING"
1150 PRINT :TAB(13); "YEAR"; YR
1160 GOTO 1280
1170 PRINT :TAB(3); "YOUR POPULATION HAS
FALLEN"
1180 PRINT :TAB(5); "TO "; STR$(INT(FK)); "
. DO YOU WANT TO"
1190 PRINT :TAB(3); "COMMIT SUICIDE PAINL
ESSLY"

```

```

1200 PRINT :TAB(3);"NOW (1) OR AWAIT A S
AD AND"
1210 PRINT :TAB(3);"LINGERING DEATH (2)?
"
1220 INPUT B
1230 CALL CLEAR
1240 IF B<>2 THEN 1270
1250 PRINT "I HOPE YOU HAVE CHOSEN WELL.
"
1260 GOTO 270
1270 PRINT :TAB(12);"GOODBYE."
1280 FOR P=1 TO 2
1290 FOR N=0 TO 500 STEP 10
1300 CALL SOUND(300-N/2,110+N,0)
1310 NEXT N
1320 NEXT P
1330 STOP
1340 J=1+INT(RND*6)
1350 PRINT TAB(2);"THE COLONY WAS ATTACK
ED BY"
1360 PRINT :TAB(2);
1370 ON J GOTO 1380,1400,1420,1440,1460,
1490
1380 PRINT "A FLEET OF SYRIAN SHIPS."
1390 GOTO 1500
1400 PRINT "RENEGADE EARTHLINGS."
1410 GOTO 1500
1420 PRINT "MARTIAN SPACE PILOTS."
1430 GOTO 1500
1440 PRINT "VYRILLIEX OUTWORLDERS."
1450 GOTO 1500
1460 PRINT "A LONE SHIP, APPARENTLY"
1470 PRINT :TAB(3);"UNDER ROBOT CONTROL.
"
1480 GOTO 1500
1490 PRINT "A PARRALEXIAN ESCORT VESSEL"
1500 CALL SOUND(4000,200,0,202,0)
1510 Z=1+INT(FK/(2+INT(RND*15)))
1520 PRINT ::"THERE WERE";Z;"PEOPLE KILL
ED"

```

```
1530 X=250+INT(RND*250)
1540 PRINT :TAB(6);"#";STR$(X);" WORTH O
F DAMAGE"
1550 Y=1+INT(RND*300)
1560 W=1+INT(RND*300)
1570 PRINT :TAB(2);"AND FOOD STOCKS HAVE
FALLEN"
1580 PRINT :TAB(13);"BY";W
1590 FD=FD-W
1600 IF FD>0 THEN 1620
1610 FD=1+INT(RND*50)
1620 FK=FK-Z
1630 OX=OX-Y
1640 IF OX>0 THEN 1660
1650 OX=1+INT(RND*50)
1660 CA=CA-X
1670 PRINT ::TAB(9);"PRESS ANY KEY."
1680 CALL KEY(3,X,Y)
1690 IF Y=0 THEN 1680
1700 CALL CLEAR
1710 RETURN
```

ROULETTE

At last you have the opportunity to play one of the traditional casino games where you can gamble as much as you like and not lose a penny of your life savings. Roulette uses a wheel numbered from one to thirty-six plus zero and a table divided into squares which are also marked with these numbers, used for placing bets. The numbers are arranged in three rows of twelve. Some are red numbers and some are black. The red numbers are 1, 3, 5, 7, 9, 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.

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 of 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 rows
- 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.

Lines 30 to 200 provide the title and short display at the beginning of the game. Lines 260 to 410 input the required information and call the desired routine using ON-GOSUB. Lines 540 to 720 print the wheel on the screen and spin it. You are told whether you have won or lost by lines 730 to 830 and a jump is then made back to line 250 which lets you try again. The sub-routines (lines 890 to 2000) work out which numbers you are betting on and set the payout odds.

```

10 RANDOMIZE
20 CALL CHAR(128,"FAFFFAFFFFF5FFFF5F")
30 CALL CLEAR
40 CALL SCREEN(16)
50 PRINT TAB(8);"R O U L E T T E"
60 FOR N=1 TO 13
70 PRINT
80 NEXT N
90 V=11
100 W=27
110 FOR P=1 TO 2
120 FOR N=0 TO 6 STEP .4
130 X=10*SIN(N)+11
140 Y=10*COS(N)+17
150 CALL HCHAR(V,W,32,1)
160 CALL HCHAR(X,Y,42,1)
170 V=X
180 W=Y

```

```

190 NEXT N
200 NEXT P
210 Z$=CHR$(128)
220 W$="      "
230 DIM B(24)
240 CH=100
250 CALL CLEAR
260 PRINT TAB(6);"YOU HAVE";CH;"CHIPS."
270 PRINT :TAB(5);"LADIES AND GENTLEMEN,"
"
280 PRINT :TAB(4);"PLEASE PLACE YOUR BET
S."
290 CALL KEY(3,J,K)
300 IF (J<65)+(J>81)<>0 THEN 290
310 IF J=81 THEN 2010
320 A=J-64
330 FOR Q=1 TO 24
340 B(Q)=-99
350 NEXT Q
360 PRINT ::
370 INPUT "      HOW MANY CHIPS? ":CP
380 IF CP>CH THEN 370
390 CH=CH-CP
400 PRINT ::
410 ON A GOSUB 890,930,1000,1080,1150,12
30,1290,1350,1410,1470,1530,1620,1720,17
90,1890,1950
420 PRINT ::TAB(6);"YOU HAVE BET ON THE"
430 PRINT :TAB(7);"FOLLOWING NUMBERS:"
440 PRINT
450 FOR P=1 TO N
460 IF (P-1)/5<>INT((P-1)/5) THEN 480
470 PRINT :TAB(3);
480 PRINT B(P);
490 IF P=N THEN 510
500 PRINT ", ";
510 NEXT P
520 FOR N=1 TO 600
530 NEXT N
540 PRINT ::TAB(2);"** THE WHEEL IS SPIN
NING **"

```

```

550 PRINT :::
560 PRINT :TAB(12);Z#;Z#;Z#;Z#;Z#;Z#;Z#;
Z#
570 FOR N=1 TO 5
580 PRINT TAB(12);Z#;W#;Z#
590 NEXT N
600 PRINT TAB(12);Z#;Z#;Z#;Z#;Z#;Z#;Z#;Z
#
610 PRINT ::
620 FOR N=1 TO 50
630 C=INT(RND*37)
640 X=INT(C/10)
650 CALL HCHAR(18,17,X+48,1)
660 CALL HCHAR(18,18,C-X*10+48,1)
670 CALL SOUND(200,-5,0)
680 FOR P=1 TO N*2
690 NEXT P
700 NEXT N
710 PRINT :TAB(4);"THE BALL HAS STOPPED
ON"
720 PRINT :TAB(15);C
730 E=1
740 IF B(E)=C THEN 790
750 E=E+1
760 IF E<25 THEN 740
770 PRINT :TAB(9);"YOU HAVE LOST"
780 GOTO 840
790 WI=INT(OD*CP)
800 PRINT :TAB(3);"CONGRATULATIONS, YOU
HAVE"
810 PRINT :TAB(5);"WON";WI;"CHIPS PLUS Y
OUR"
820 PRINT :TAB(8);"BET OF";CP;"CHIPS"
830 CH=CH+WI+CP
840 FOR N=1 TO 600
850 NEXT N
860 IF CH=0 THEN 2110
870 IF CH>1100 THEN 2190
880 GOTO 250
890 INPUT " WHICH NUMBER? ":B(1)

```

```

900 OD=35
910 N=1
920 RETURN
930 INPUT "      FIRST NUMBER? ":B(1)
940 PRINT
950 INPUT "      SECOND NUMBER? ":B(2)
960 IF (ABS(B(1)-B(2))=1)+(ABS(B(1)-B(2)
)=3)=0 THEN 940
970 OD=17
980 N=2
990 RETURN
1000 PRINT "      WHICH NUMBER IN"
1010 INPUT "      LEFT COLUMN? ":D
1020 FOR E=0 TO 2
1030 B(E+1)=D+E
1040 NEXT E
1050 OD=11
1060 N=3
1070 RETURN
1080 INPUT "      FIRST NUMBER OF SIX? ":D
1090 FOR E=0 TO 5
1100 B(E+1)=D+E
1110 NEXT E
1120 OD=5
1130 N=6
1140 RETURN
1150 INPUT "      FIRST NUMBER IN SQUARE? ":D
1160 B(1)=D
1170 B(2)=D+1
1180 B(3)=D+3
1190 B(4)=D+4
1200 OD=8
1210 N=4
1220 RETURN
1230 FOR E=1 TO 12
1240 B(E)=E
1250 NEXT E
1260 OD=2
1270 N=12
1280 RETURN

```

```

1290 FOR E=1 TO 12
1300 B(E)=E+12
1310 NEXT E
1320 OD=2
1330 N=12
1340 RETURN
1350 FOR E=1 TO 12
1360 B(E)=E+24
1370 NEXT E
1380 OD=2
1390 N=12
1400 RETURN
1410 FOR E=1 TO 18
1420 B(E)=E
1430 NEXT E
1440 OD=1
1450 N=18
1460 RETURN
1470 FOR E=1 TO 18
1480 B(E)=E+18
1490 NEXT E
1500 OD=1
1510 N=18
1520 RETURN
1530 PRINT "    LOW NUMBER AT END OF"
1540 INPUT "    LINE? (1,2 OR 3) ":D
1550 IF (D=1)+(D=2)+(D=3)=0 THEN 1530
1560 FOR E=0 TO 11
1570 B(E+1)=D+E*3
1580 NEXT E
1590 OD=2
1600 N=12
1610 RETURN
1620 PRINT "    LOW NUMBER AT END OF"
1630 INPUT "    FIRST ROW? (1 OR 2) ":D
1640 IF (D=1)+(D=2)=0 THEN 1620
1650 FOR E=0 TO 11
1660 B(E+1)=D+3*E
1670 B(E+13)=D+3*E+1
1680 NEXT E

```

```

1690 OD=.5
1700 N=24
1710 RETURN
1720 RESTORE
1730 FOR E=1 TO 18
1740 READ B(E)
1750 NEXT E
1760 OD=1
1770 N=18
1780 RETURN
1790 RESTORE
1800 FOR E=1 TO 18
1810 READ Z
1820 NEXT E
1830 FOR E=1 TO 18
1840 READ B(E)
1850 NEXT E
1860 OD=1
1870 N=18
1880 RETURN
1890 FOR E=2 TO 36 STEP 2
1900 B(E/2)=E
1910 NEXT E
1920 OD=1
1930 N=18
1940 RETURN
1950 FOR E=1 TO 35 STEP 2
1960 B((E+1)/2)=E
1970 NEXT E
1980 OD=1
1990 N=18
2000 RETURN
2010 CALL CLEAR
2020 PRINT :TAB(4);"YOU ARE WITHDRAWING
FROM"
2030 PRINT :TAB(5);"THE GAME WITH";CH;"C
HIPS"
2040 FOR S=110 TO 1010 STEP 20
2050 CALL SOUND(-500,S,0,S+20,0)
2060 NEXT S
2070 FOR S=1010 TO 110 STEP -20

```

```
2080 CALL SOUND(-500,S,0,S+20,0)
2090 NEXT S
2100 STOP
2110 FOR N=1 TO 24
2120 PRINT "YOU HAVE RUN OUT OF CHIPS..."
"
2130 CALL SOUND(-300,110+RND*1000,0)
2140 NEXT N
2150 FOR N=1 TO 50
2160 CALL SOUND(-300,110+RND*1000,0)
2170 NEXT N
2180 STOP
2190 FOR N=1 TO 10
2200 CALL SOUND(-300,262,0,294,0)
2210 PRINT TAB(5);"YOU"
2220 NEXT N
2230 FOR N=1 TO 10
2240 CALL SOUND(-300,294,0,330,0)
2250 PRINT TAB(9);"HAVE"
2260 NEXT N
2270 FOR N=1 TO 10
2280 CALL SOUND(-300,330,0,349,0)
2290 PRINT TAB(14);"BROKEN"
2300 NEXT N
2310 FOR N=1 TO 10
2320 CALL SOUND(-300,349,0,392,0)
2330 PRINT TAB(21);"THE"
2340 NEXT N
2350 FOR N=1 TO 10
2360 CALL SOUND(-300,392,0,440,0)
2370 PRINT TAB(25);"BANK"
2380 NEXT N
2390 STOP
2400 DATA 1,3,5,7,9,12,14,16,18,19,21,23
,25,27,30,32,34,36
2410 DATA 2,4,6,8,10,11,13,15,17,20,22,2
4,26,28,29,31,33,35
```

AIR RAID

A squadron of enemy bombers are approaching the capital and they will turn the buildings into rubble if you do not intercept them soon. Sitting in the cockpit of your fighter you must line the aircraft up in your sights and then watch them explode when they are hit by your machine gun fire.

To pilot your fighter use keys S to bank left, D to bank right, U to climb and N to dive. The machine gun will fire automatically when the aircraft is dead in the centre of your sights. The game may seem a little confusing at first as the enemy aircraft moves left when you move right, up when you move down and so on. It helps if you remember that the picture you are seeing on the television screen is the actual view out of the cockpit window.

You must shoot down as many enemy aircraft as you can before the time runs out. The score attained for shooting an aircraft down depends on the time it takes you to do it. Obviously, the longer you take, the less points you score when you destroy it. The game features on screen scoring and also has a high score routine which keeps track of the highest number of points scored in a single game.

Lines 10 to 380 initiate the game, display the title and set the screen up. Lines 390 to 770 move the aircraft around the screen and subtract 10 points from the score if the aircraft is allowed to go off of the screen. Lines 780 to 1060 display the machine gun fire and increase the score accordingly.

```

10 RANDOMIZE
20 CALL CHAR(128,"02868CFF18306000")
30 CALL CLEAR
40 CALL SCREEN(8)
50 CALL COLOR(1,8,8)
60 CALL COLOR(2,2,1)
70 FOR N=3 TO 7
80 CALL COLOR(N,14,1)
90 NEXT N
100 CALL COLOR(8,2,1)
110 CALL COLOR(12,2,1)
120 HI=0
130 A#="AIR RAID"
140 CALL CLEAR
150 FOR N=1 TO 8
160 CALL HCHAR(11,N+12,43,1)
170 CALL VCHAR(6,N+12,124,5)
180 CALL VCHAR(12,N+12,124,5)
190 CALL SOUND(-50,-5,0)
200 CALL SOUND(200,-5,10)
210 CALL VCHAR(6,N+12,32,5)
220 CALL VCHAR(12,N+12,32,5)
230 CALL HCHAR(11,N+12,ASC(SEG$(A#,N,1)),
,1)
240 NEXT N
250 FOR N=1 TO 400
260 NEXT N
270 FOR N=1 TO 11
280 PRINT
290 CALL SOUND(-50,-5,0)
300 CALL SOUND(200,-5,10)
310 NEXT N
320 TI=99
330 TJ=100
340 SC=0
350 A#="SCORE=0      TIME=99      HI="&STR$(
HI)%"  "
360 FOR N=5 TO 32
370 CALL HCHAR(24,N,ASC(SEG$(A#,N-4,1)),
1)

```

```

380 NEXT N
390 CALL HCHAR(12,16,91,1)
400 CALL HCHAR(12,17,43,1)
410 CALL HCHAR(12,18,93,1)
420 Z=RND
430 CL=5-2*(Z<.75)-6*(Z<.5)-(Z<.25)
440 CALL COLOR(13,CL,1)
450 X=5+INT(RND*16)
460 Y=6+INT(RND*22)
470 C=32
480 V=X
490 W=Y
500 TI=TI-1
510 Z$=STR$(TI)&" "
520 CALL HCHAR(24,22,ASC(SEG$(Z$,1,1)),1)
)
530 CALL HCHAR(24,23,ASC(SEG$(Z$,2,1)),1)
)
540 CALL SOUND(100,131,0,156,0)
550 IF TI=0 THEN 1080
560 CALL KEY(3,J,K)
570 A=(J=78)-(J=85)
580 B=(J=68)-(J=83)
590 Z=RND
600 X=X+(Z<.4)-(Z>.6)+A
610 Z=RND
620 Y=Y+(Z<.4)-(Z>.6)+B
630 IF (X<1)+(X>22)<>0 THEN 700
640 IF (Y<3)+(Y>30)<>0 THEN 700
650 CALL HCHAR(V,W,C,1)
660 CALL GCHAR(X,Y,C)
670 CALL HCHAR(X,Y,128,1)
680 IF C=43 THEN 780
690 GOTO 480
700 CALL HCHAR(V,W,32,1)
710 SC=SC-10
720 SC=-SC*(SC>0)
730 A$=STR$(SC)&" "
740 CALL HCHAR(24,11,ASC(SEG$(A$,1,1)),1)
)

```

```

750 CALL HCHAR(24,12,ASC(SEG$(A$,2,1)),1
)
760 CALL HCHAR(24,13,ASC(SEG$(A$,3,1)),1
)
770 GOTO 420
780 FOR N=22 TO 18 STEP -1
790 CALL HCHAR(N,28-N,47,1)
800 CALL HCHAR(N,N+6,92,1)
810 CALL SOUND(-50,-5,0)
820 CALL SOUND(200,-5,10)
830 NEXT N
840 FOR N=22 TO 18 STEP -1
850 CALL HCHAR(N,28-N,32,1)
860 CALL HCHAR(N-5,33-N,47,1)
870 CALL HCHAR(N,N+6,32,1)
880 CALL HCHAR(N-5,N+1,92,1)
890 CALL SOUND(-500,-5,35-N)
900 NEXT N
910 FOR N=17 TO 13 STEP -1
920 CALL HCHAR(N,28-N,32,1)
930 CALL HCHAR(N,N+6,32,1)
940 CALL SOUND(-300,-5,35-N)
950 NEXT N
960 FOR N=0 TO 30
970 CALL COLOR(13,CL,1)
980 CALL COLOR(13,B,8)
990 CALL SOUND(-500,-7,N)
1000 NEXT N
1010 SC=SC+(100-(TJ-TI))
1020 TJ=TI
1030 A$=STR$(SC)&"  "
1040 CALL HCHAR(24,11,ASC(SEG$(A$,1,1)),
1)
1050 CALL HCHAR(24,12,ASC(SEG$(A$,2,1)),
1)
1060 CALL HCHAR(24,13,ASC(SEG$(A$,3,1)),
1)
1070 GOTO 390
1080 A$="OUT OF TIME"
1090 CALL HCHAR(V,W,32,1)

```

```
1100 CALL HCHAR(12,16,32,3)
1110 FOR N=12 TO 22
1120 CALL HCHAR(11,N,ASC(SEG#(A#,N-11,1)
),1)
1130 NEXT N
1140 CALL COLOR(13,11,11)
1150 CALL HCHAR(9,11,128,13)
1160 CALL HCHAR(13,11,128,13)
1170 CALL VCHAR(9,10,128,5)
1180 CALL VCHAR(9,24,128,5)
1190 FOR N=1 TO 30
1200 CALL COLOR(13,11,11)
1210 CALL COLOR(13,8,8)
1220 NEXT N
1230 FOR N=1 TO 14
1240 PRINT
1250 NEXT N
1260 IF SC<HI THEN 1280
1270 HI=SC
1280 PRINT TAB(10);"ANOTHER GO ?"
1290 PRINT ::
1300 CALL KEY(3,J,K)
1310 IF J=89 THEN 1300
1320 IF J<>78 THEN 1300
1330 STOP
```

DIGIT BRAIN

Digit Brain will test your skills of deduction. The computer generates a four digit 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 and give you coded clues to its accuracy. Black means you have correctly guessed a number and it is in the right place. White means you have the number correct but in the wrong place.

After you have had some practice at Digit Brain you may wish to increase the difficulty to make it a little more challenging. This is done quite simply by altering the number in line 540. The number is always one more than the number of guesses that you are allowed to have before the computer tells you the correct code.

The four digit code is selected in lines 170 to 240. Lines 260 to 310 input your guess and split it up into single numbers. Lines 330 to 520 work out how many blacks and whites you have scored.

```
10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(8)
40 A$="* DIGIT BRAIN *"
50 FOR N=1 TO 5
60 FOR M=7 TO 13
70 PRINT TAB(M):A$
80 NEXT M
90 FOR M=12 TO 6 STEP -1
```

```

100 PRINT TAB(M);A#
110 NEXT M
120 NEXT N
130 FOR N=1 TO 1000
140 NEXT N
150 DIM A(4),B(4)
160 C=1
170 A(1)=1+INT(RND*9)
180 A(2)=1+INT(RND*9)
190 IF A(2)=A(1) THEN 180
200 A(3)=1+INT(RND*9)
210 IF (A(3)=A(1))+ (A(3)=A(2))<>0 THEN 200
220 A(4)=1+INT(RND*9)
230 IF (A(4)=A(1))+ (A(4)=A(2))+ (A(4)=A(3))<>0 THEN 220
240 D=1000*A(1)+100*A(2)+10*A(3)+A(4)
250 CALL CLEAR
260 PRINT :;TAB(12);"GUESS";C
270 INPUT X
280 B(1)=INT(X/1000)
290 B(2)=INT((X-1000*B(1))/100)
300 B(3)=INT((X-1000*B(1)-100*B(2))/10)
310 B(4)=X-1000*B(1)-100*B(2)-10*B(3)
320 IF D=X THEN 570
330 N=0
340 W=0
350 FOR E=1 TO 4
360 IF A(E)<>B(E) THEN 390
370 N=N+1
380 A(E)=0
390 NEXT E
400 FOR F=1 TO 4
410 IF A(F)=0 THEN 460
420 FOR E=1 TO 4
430 IF B(F)<>A(E) THEN 450
440 W=W+1
450 NEXT E
460 NEXT F
470 A(1)=INT(D/1000)

```

```
480 A(2)=INT((D-1000*A(1))/100)
490 A(3)=INT((D-1000*A(1)-100*A(2))/10)
500 A(4)=D-1000*A(1)-100*A(2)-10*A(3)
510 PRINT :TAB(11);"YOU SCORED"
520 PRINT :TAB(5);N;"BLACKS AND";W;"WHIT
ES."
530 C=C+1
540 IF C<16 THEN 260
550 PRINT :TAB(3);"MY NUMBER WAS ";A(1)
;A(2);A(3);A(4)
560 STOP
570 PRINT :TAB(8);"CONGRATULATIONS"
580 PRINT :TAB(8);"YOU GOT IT IN";C
590 FOR N=1 TO 20
600 X=110+RND*400
610 Y=510+RND*400
620 Z=910+RND*400
630 CALL SOUND(-500,X,0,Y,0,Z,0)
640 NEXT N
650 END
```

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 (light blue). 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 they change colour and 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 go first. 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 number along the bottom. If you cannot move enter 0 for the first number. The computer will keep track of the score and will 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 hard to beat.

Lines 620 to 940 contain the logic for the computer's moves. Lines 1010 to 1250 control the player's moves. Lines 1280 to 1340 use CALL HCHAR to print up the grid pattern and keep it in place during the course of the game. Lines 1420 to 1580 work out which pieces have been captured and change their colour accordingly.

```

10 RANDOMIZE
20 CALL CLEAR
30 CALL CHAR(128,"3C7EFFFFFFFF7E3C")
40 CALL CHAR(129,"0000001818000000")
50 CALL CHAR(136,"3C7EFFFFFFFF7E3C")
60 CALL CHAR(144,"FF87BBBB87AFB7BB")
70 CALL CHAR(145,"FFFFFFFFFFFFFFFF")
80 CALL SCREEN(16)
90 CALL COLOR(13,16,2)
100 CALL COLOR(14,6,2)
110 CALL COLOR(15,2,1)
120 PRINT TAB(12);CHR$(144);"EVERSI"
130 FOR N=1 TO 12
140 PRINT
150 CALL SOUND(100,250,0)
160 NEXT N
170 FOR N=1 TO 4
180 CALL COLOR(15,16,2)
190 FOR P=1 TO 150
200 NEXT P
210 CALL COLOR(15,2,1)
220 FOR P=1 TO 150
230 NEXT P
240 NEXT N
250 R=1
260 DIM A(10,10)
270 FOR B=2 TO 9
280 FOR C=2 TO 9
290 A(B,C)=129
300 NEXT C
310 NEXT B
320 A(5,5)=128
330 A(6,6)=128
340 A(5,6)=136
350 A(6,5)=136
360 CALL CLEAR
370 PRINT TAB(4);"DO YOU WISH TO GO FIRS
T?"
380 PRINT ::
390 CALL KEY(3,KY,KK)

```

```

400 IF (KY=78)+(KY=89)=0 THEN 390
410 CALL CLEAR
420 PRINT TAB(12);"I HAVE 2"
430 PRINT :TAB(11);"YOU HAVE 2"
440 FOR N=1 TO 5
450 PRINT
460 NEXT N
470 CALL HCHAR(3,13,145,10)
480 CALL HCHAR(12,13,145,10)
490 CALL VCHAR(4,13,145,8)
500 CALL VCHAR(4,22,145,8)
510 FOR N=1 TO 8
520 CALL HCHAR(3+N,12,N+48,1)
530 CALL HCHAR(13,13+N,N+48,1)
540 NEXT N
550 GOSUB 1260
560 IF KY=89 THEN 1020
570 CALL SOUND(200,120,0)
580 A$=" MY MOVE "
590 GOSUB 1820
600 S=128
610 T=136
620 H=0
630 FOR B=2 TO 9
640 FOR C=2 TO 9
650 IF A(B,C)<>129 THEN 930
660 Q=0
670 FOR D=-1 TO 1
680 FOR E=-1 TO 1
690 K=0
700 F=B
710 G=C
720 IF A(F+D,G+E)<>128 THEN 770
730 K=K+1
740 F=F+D
750 G=G+E
760 GOTO 720
770 IF A(F+D,G+E)<>136 THEN 790
780 Q=Q+K
790 NEXT E

```

```

800 NEXT D
810 IF (B=2)+(B=9)+(C=2)+(C=9)=0 THEN 83
0
820 Q=Q*2
830 IF (B=3)+(B=8)+(C=3)+(C=8)=0 THEN 85
0
840 Q=Q/2
850 IF ((B=2)+(B=9)<>0)+((C=3)+(C=8)<>0)
<>-2 THEN 870
860 Q=Q/2
870 IF ((B=3)+(B=8)<>0)+((C=2)+(C=9)<>0)
<>-2 THEN 890
880 Q=Q/2
890 IF (Q<H)+(Q=0)+((RND>.3)+(Q=H)=-2)<>
0 THEN 930
900 H=Q
910 M=B
920 N=C
930 NEXT C
940 NEXT B
950 IF (H=0)+(R=0)=-2 THEN 1590
960 IF H=0 THEN 1010
970 CALL HCHAR(24,16,M+47,1)
980 CALL HCHAR(24,17,N+47,1)
990 GOSUB 1420
1000 GOSUB 1260
1010 CALL SOUND(200,150,0)
1020 A$="YOUR MOVE"
1030 GOSUB 1820
1040 CALL KEY(3,KY,KK)
1050 IF (KY<48)+(KY>56)<>0 THEN 1040
1060 CALL SOUND(100,600,0)
1070 CALL HCHAR(24,16,KY,1)
1080 IF KY<>48 THEN 1110
1090 R=0
1100 GOTO 570
1110 R=1
1120 M=KY-47
1130 CALL KEY(3,KY,KK)
1140 IF KK<>0 THEN 1130

```

```

1150 CALL KEY(3,KY,KK)
1160 IF (KY<49)+(KY>56)<>0 THEN 1150
1170 CALL SOUND(100,600,0)
1180 CALL HCHAR(24,17,KY,1)
1190 N=KY-47
1200 IF A(M,N)<>129 THEN 1010
1210 S=136
1220 T=128
1230 GOSUB 1420
1240 GOSUB 1260
1250 GOTO 570
1260 C=0
1270 H=0
1280 FOR B=2 TO 9
1290 FOR D=2 TO 9
1300 CALL HCHAR(B+2,D+12,A(B,D),1)
1310 C=C-(A(B,D)=136)
1320 H=H-(A(B,D)=128)
1330 NEXT D
1340 NEXT B
1350 A#=STR$(C)&" "
1360 CALL HCHAR(16,21,ASC(SEG$(A#,1,1)),
1)
1370 CALL HCHAR(16,22,ASC(SEG$(A#,2,1)),
1)
1380 A#=STR$(H)&" "
1390 CALL HCHAR(18,22,ASC(SEG$(A#,1,1)),
1)
1400 CALL HCHAR(18,23,ASC(SEG$(A#,2,1)),
1)
1410 RETURN
1420 FOR C=-1 TO 1
1430 FOR D=-1 TO 1
1440 F=M
1450 G=N
1460 IF A(F+C,G+D)<>S THEN 1500
1470 F=F+C
1480 G=G+D
1490 GOTO 1460
1500 IF A(F+C,G+D)<>T THEN 1560

```

```

1510 A(F,G)=T
1520 IF (M=F)+(N=G)=-2 THEN 1560
1530 F=F-C
1540 G=G-D
1550 GOTO 1510
1560 NEXT D
1570 NEXT C
1580 RETURN
1590 C=0
1600 H=0
1610 FOR B=2 TO 9
1620 FOR D=2 TO 9
1630 C=C-(A(B,D)=136)
1640 H=H-(A(B,D)=128)
1650 NEXT D
1660 NEXT B
1670 CALL CLEAR
1680 FOR F=110 TO 210 STEP 5
1690 CALL SOUND(-400,F,0)
1700 NEXT F
1710 IF H=C THEN 1770
1720 IF H>C THEN 1750
1730 PRINT TAB(10);"I WON:";C;H
1740 GOTO 1780
1750 PRINT TAB(9);"YOU WON:";H;C
1760 GOTO 1780
1770 PRINT TAB(6);"IT IS A DRAW:";C;H
1780 FOR N=1 TO 5
1790 PRINT
1800 NEXT N
1810 STOP
1820 CALL HCHAR(22,13,32,9)
1830 CALL HCHAR(24,16,32,2)
1840 FOR Q=1 TO 9
1850 CALL HCHAR(22,12+Q,ASC(SEG$(A$,Q,1)
),1)
1860 NEXT Q
1870 RETURN

```

SPACE SURVIVAL

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 kilometre 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 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 which area of the cube you wish to search. Enter three numbers from one to eight in turn. The computer will look at this area and tell you that you have found the astronaut or it will give you clues to help you look again.

Lines 220 to 310 accept your co-ordinates (D,E,F) and then line 340 checks these against the location of the astronaut (A,B,C). If they are not the same lines 370 to 490 see where the astronaut is in relation to you and provide the clues. Variable J counts the number of turns and keeps track of the astronaut's oxygen supply by using a loop.

```
10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(10)
40 PRINT TAB(9); "SPACE SURVIVAL"
50 PRINT ::
60 CALL SOUND(1000,122,0)
70 FOR N=1 TO 600
```

```

80 NEXT N
90 CALL SOUND(500,130,0)
100 PRINT :TAB(3);"YOU HAVE 10 HOURS TO
FIND"
110 PRINT :TAB(4);"AN ASTRONAUT LOST IN
AN"
120 PRINT :TAB(3);"8 KILOMETRE CUBE OF S
PACE."
130 A=1+INT(RND*8)
140 B=1+INT(RND*8)
150 C=1+INT(RND*8)
160 PRINT ::
170 FOR N=1 TO 1000
180 NEXT N
190 FOR J=1 TO 10
200 CALL SOUND(200,140,0)
210 CALL CLEAR
220 PRINT "WHERE DO YOU WISH TO SEARCH?"
230 PRINT :TAB(8);"X CO-ORDINATE.."
240 INPUT D
250 IF (D<1)+(D>8)<>0 THEN 240
260 PRINT :TAB(8);"Y CO-ORDINATE.."
270 INPUT E
280 IF (E<1)+(E>8)<>0 THEN 270
290 PRINT :TAB(8);"Z CO-ORDINATE.."
300 INPUT F
310 IF (F<1)+(F>8)<>0 THEN 300
320 CALL CLEAR
330 CALL SOUND(200,300,0)
340 IF (A=D)+(B=E)+(C=F)=-3 THEN 640
350 PRINT TAB(6);"ASTRONAUT NOT THERE."
360 PRINT :TAB(3);"HOURS OF AIR REMAININ
G:";10-J
370 PRINT ::TAB(13);"MOVE:"
380 IF A>D THEN 420
390 IF A=D THEN 430
400 PRINT :TAB(15);"UP"
410 GOTO 430
420 PRINT :TAB(14);"DOWN"
430 IF B=E THEN 450

```

```

440 PRINT :TAB(13);"ACROSS"
450 IF C<F THEN 490
460 IF C=F THEN 500
470 PRINT :TAB(12);"FORWARDS"
480 GOTO 500
490 PRINT :TAB(11);"BACKWARDS"
500 IF J<9 THEN 530
510 PRINT ::TAB(3);"DANGER...ASTRONAUTS
DEATH"
520 PRINT :TAB(11);"IMMINENT."
530 FOR N=1 TO 800
540 NEXT N
550 NEXT J
560 CALL CLEAR
570 PRINT TAB(7);"YOU HAVE FAILED!!"
580 PRINT :TAB(5);"ASTRONAUT OUT OF AIR.
"
590 PRINT :TAB(8);"HE WAS AT";A;B;C
600 FOR N=1 TO 800
610 NEXT N
620 CALL SOUND(1000,110,0,120,0)
630 STOP
640 FOR N=1 TO 5
650 FOR S=110 TO 510 STEP 20
660 CALL SOUND(-400,S,0,620-S,0)
670 NEXT S
680 NEXT N
690 PRINT TAB(7);"YOU HAVE FOUND THE"
700 PRINT :TAB(7);"ASTRONAUT. HE HAD"
710 PRINT :TAB(6);10-J;"HOURS OF AIR LEF
T."
720 FOR N=1 TO 800
730 NEXT N
740 STOP

```

CUBIK'S RUBE

This is a puzzle in which there is a four by four grid made up of sixteen coloured blocks. The completed puzzle should have four of the same colour blocks in each corner of the grid. This is how it is displayed at the beginning of the game but you are then asked to wait while the computer shuffles the blocks about.

A prompt will then appear asking you which line you wish to move. Entering a number from one to four will cause the relevant vertical line to scroll one space downwards. The block that goes off of the bottom of the grid will reappear at the top. Similarly, entering a number from five to eight will cause the relevant horizontal line to scroll one space to the right with the rightmost block reappearing on the left.

Although the computer only makes twelve random moves of the cube to shuffle it, lines 950 to 990, you will probably have to make many times more moves than this to get it back again. When you think you have solved the puzzle press S and the computer will check it for you. If the puzzle is incorrect you will be told so and you will have to carry on shuffling the blocks until you get it right. This puzzle is very challenging and you may become frustrated at first but if you keep trying you will soon get the hang of it.

Lines 490 to 900 set up the screen and tell you the correct position for each colour on the grid. Lines 1050 to 1240 ask you which line you wish to move, call the subroutine at lines 2000 to 2140 which scrolls the relevant line, check to see if you have pressed key S and increment the number of moves you have had so far (variable MV).

```

10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(16)
40 CALL COLOR(9,13,13)
50 CALL COLOR(10,5,5)
60 CALL COLOR(11,10,10)
70 CALL COLOR(12,14,14)
80 CALL COLOR(13,6,1)
90 CALL CHAR(128,"FF9999FFFF9999FF")
100 CALL CHAR(129,"0000000000000000FF")
110 CALL CHAR(130,"FF00000000000000")
120 CALL CHAR(131,"0101010101010101")
130 CALL CHAR(132,"8080808080808080")
140 BS=500
150 RESTORE
160 FOR N=1 TO 27
170 READ D,E,C
180 CALL HCHAR(D,E,128,C)
190 NEXT N
200 FOR N=1 TO 21
210 READ D,E,C
220 CALL VCHAR(D,E,128,C)
230 NEXT N
240 FOR M=1 TO 6
250 FOR N=2 TO 16
260 CALL COLOR(13,N,1)
270 CALL SOUND(100,M*100+N*10,0)
280 NEXT N
290 NEXT M
300 CALL COLOR(13,6,1)
310 MV=0
320 DIM A(4,4)
330 FOR X=1 TO 2
340 FOR Y=1 TO 2
350 A(X,Y)=96
360 NEXT Y
370 FOR Y=3 TO 4
380 A(X,Y)=112
390 NEXT Y
400 NEXT X

```

```

410 FOR X=3 TO 4
420 FOR Y=1 TO 2
430 A(X,Y)=104
440 NEXT Y
450 FOR Y=3 TO 4
460 A(X,Y)=120
470 NEXT Y
480 NEXT X
490 CALL CLEAR
500 FOR X=1 TO 4
510 FOR Y=1 TO 4
520 CALL HCHAR(8+X,15+Y,A(X,Y),1)
530 NEXT Y
540 NEXT X
550 FOR N=5 TO 7
560 CALL HCHAR(N,N+8,129,20-N*2)
570 CALL HCHAR(N+9,20-N,130,N*2-4)
580 NEXT N
590 FOR N=12 TO 14
600 CALL VCHAR(N-6,N,131,34-N*2)
610 CALL VCHAR(20-N,N+9,132,N*2-18)
620 NEXT N
630 FOR N=49 TO 52
640 CALL HCHAR(8,N-33,N,1)
650 NEXT N
660 FOR N=53 TO 56
670 CALL HCHAR(N-44,20,N,1)
680 NEXT N
690 A$="CUBIKS RUBE"
700 J=1
710 K=11
720 GOSUB 1250
730 A$="PRESS S WHEN PUZZLE SOLVED"
740 J=3
750 K=4
760 GOSUB 1250
770 A$="COLORS ARE: TOP LEFT      "&CHR$(
96)
780 J=17
790 K=3

```

```

800 GOSUB 1250
810 A$="TOP RIGHT      "&CHR$(112)
820 J=18
830 K=15
840 GOSUB 1250
850 A$="BOTTOM LEFT   "&CHR$(104)
860 J=19
870 GOSUB 1250
880 A$="BOTTOM RIGHT "&CHR$(120)
890 J=20
900 GOSUB 1250
910 A$="PLEASE WAIT"
920 J=22
930 K=11
940 GOSUB 1250
950 FOR M=1 TO 12
960 Z=1+INT(RND*8)
970 CALL SOUND(250,110+Z*50,0)
980 GOSUB 2000
990 NEXT M
1000 A$="MOVES = 0      BEST = "&STR$(BS)
    &" "
1010 J=24
1020 K=5
1030 GOSUB 1250
1040 CALL HCHAR(22,12,32,11)
1050 A$="WHICH LINE ?:"
1060 J=22
1070 K=10
1080 GOSUB 1250
1090 CALL KEY(3,Z,K)
1100 IF Z=83 THEN 1120
1110 IF (Z<49)+(Z>56)<>0 THEN 1090
1120 CALL HCHAR(22,24,Z,1)
1130 IF Z<>83 THEN 1150
1140 GOTO 1370
1150 Z=Z-48
1160 CALL SOUND(250,110+Z*50,0)
1170 MV=MV+1

```

```

1180 A$=STR$(MV)&"  "
1190 J=24
1200 K=13
1210 GOSUB 1250
1220 CALL HCHAR(22,24,32,1)
1230 GOSUB 2000
1240 GOTO 1090
1250 FOR N=1 TO LEN(A$)
1260 CALL HCHAR(J,K+N,ASC(SEG$(A$,N,1)),
1)
1270 NEXT N
1280 RETURN
1290 DATA 3,4,3,10,4,3,10,9,2,3,14,2,6,1
4,2,10,14,2,3,18
1300 DATA 3,10,18,3,5,24,1,8,24,1,4,25,1
,3,28,2,6,28,2,10,28,2
1310 DATA 9,27,1,4,30,1,15,9,2,18,9,2,19
,9,1,20,10,1,22,14
1320 DATA 2,15,19,2,18,19,2,22,19,2,15,2
4,3,18,24,2,22,24
1330 DATA 3,4,3,6,3,8,7,3,11,7,3,13,8,4,
16,2,7,16,3,4,19
1340 DATA 6,3,22,8,6,23,2,9,25,2,4,27,2,
7,30,3,16,8,7,16
1350 DATA 11,2,21,11,2,15,13,7,15,16,7,1
5,18,8,16,21,2,19
1360 DATA 21,3,15,23,8
1370 A$="CHECKING"
1380 J=22
1390 K=13
1400 CALL HCHAR(22,11,32,15)
1410 GOSUB 1250
1420 CALL SOUND(500,131,0,262,0,523,0)
1430 CK=1
1440 FOR X=1 TO 4
1450 FOR Y=1 TO 4
1460 CALL GCHAR(8+X,15+Y,Q)
1470 IF Q=A(X,Y)THEN 1490
1480 CK=0
1490 NEXT Y

```

```
1500 NEXT X
1510 IF CK=1 THEN 1640
1520 A$="INCORRECT, TRY AGAIN"
1530 J=22
1540 K=7
1550 CALL HCHAR(22,14,32,8)
1560 GOSUB 1250
1570 FOR M=1 TO 4
1580 FOR N=110 TO 156 STEP 46
1590 CALL SOUND(500,N,0)
1600 NEXT N
1610 NEXT M
1620 CALL HCHAR(22,8,32,20)
1630 GOTO 1050
1640 CALL CLEAR
1650 A$="CORRECT"
1660 J=6
1670 K=13
1680 GOSUB 1250
1690 A$="WELL DONE, YOU DID"
1700 J=8
1710 K=8
1720 GOSUB 1250
1730 A$="IT IN "&STR$(MV)&" MOVES"
1740 J=10
1750 K=10
1760 GOSUB 1250
1770 IF MV>BS THEN 1840
1780 A$="THAT IS THE BEST SO FAR"
1790 J=14
1800 K=5
1810 GOSUB 1250
1820 BS=MV
1830 GOTO 1920
1840 A$="BUT IT HAS BEEN DONE"
1850 J=14
1860 K=7
1870 GOSUB 1250
1880 A$="IN "&STR$(BS)&" MOVES!"
1890 J=16
```

```
1900 K=11
1910 GOSUB 1250
1920 A$="PRESS THE SPACE BAR"
1930 J=20
1940 K=7
1950 GOSUB 1250
1960 CALL KEY(3,J,K)
1970 IF J<>32 THEN 1960
1980 CALL CLEAR
1990 GOTO 150
2000 IF Z>4 THEN 2080
2010 CALL GCHAR(12,15+Z,0)
2020 FOR N=11 TO 9 STEP -1
2030 CALL GCHAR(N,15+Z,P)
2040 CALL HCHAR(N+1,15+Z,P,1)
2050 NEXT N
2060 CALL HCHAR(9,15+Z,0,1)
2070 RETURN
2080 CALL GCHAR(4+Z,19,0)
2090 FOR N=18 TO 16 STEP -1
2100 CALL GCHAR(4+Z,N,P)
2110 CALL HCHAR(4+Z,N+1,P,1)
2120 NEXT N
2130 CALL HCHAR(4+Z,16,0,1)
2140 RETURN
```

LIFE SIMULATION

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.

At the start of the game you must create the first generation by placing cells anywhere you wish on the grid. It is a 10 by 10 grid and there will be a cursor in the top left hand corner. If you press A a cell will appear where the cursor was and the cursor will move one space to the right. If you press the space bar the cursor will move but it will not place a cell in its previous position. Continue to do this for the rest of the grid. When the cursor reaches the right hand edge of the grid it will move down a line and reappear on the left. After the cursor has covered all ten rows of the grid it will disappear. You will then have to wait some time while the computer uses the laws to create the next generation. It will then be displayed on the screen. The computer will continue to create generation after generation until you press CLEAR. You may then run the program again if you wish to try a new pattern of cells. The idea for the program comes from John Conway's LIFE.

```

10 G=0
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 CALL CHAR(129,"FFB199999999B1FF")
40 CALL CLEAR
50 DIM M(12,12),N(12,12)
60 PRINT TAB(10);"GENERATION 0"
70 PRINT
80 GOSUB 410
90 K=12
100 Z=13
110 CALL HCHAR(K,Z,62,1)
120 CALL KEY(3,J,JJ)
130 IF (J=32)+(J=65)=0 THEN 120
140 IF J=32 THEN 190
150 CALL HCHAR(K,Z,129,1)
160 M(K-10,Z-11)=1
170 N(K-10,Z-11)=1
180 GOTO 200
190 CALL HCHAR(K,Z,128,1)
200 Z=Z+1
210 IF Z<23 THEN 110
220 Z=13
230 K=K+1
240 IF K=22 THEN 260
250 GOTO 110
260 G=G+1
270 FOR K=2 TO 11
280 FOR Z=2 TO 11
290 C=M(K-1,Z-1)+M(K-1,Z)+M(K-1,Z+1)
300 D=C+M(K,Z-1)+M(K,Z+1)
310 E=C+M(K+1,Z-1)+M(K+1,Z)+M(K+1,Z+1)
320 IF (M(K,Z)=0)+(C=3)+(D=2)<>0 THEN 340
330 N(K,Z)=0
340 IF (M(K,Z)=1)+(C<>3)<>0 THEN 360
350 N(K,Z)=1
360 NEXT Z
370 NEXT K
380 PRINT :TAB(10);"GENERATION";G
390 GOSUB 410
400 GOTO 260

```

```

410 FOR K=1 TO 12
420 PRINT :TAB(10);
430 FOR Z=1 TO 12
440 M(K,Z)=N(K,Z)
450 IF M(K,Z)=1 THEN 480
460 PRINT CHR$(128);
470 GOTO 490
480 PRINT CHR$(129);
490 NEXT Z
500 NEXT K
510 PRINT
520 PRINT
530 RETURN

```

The following printouts give a graphic illustration of the development of one colony over ten generations. You can see how the colony gradually splits into two separate groups, which become quite distinct by the 2nd generation. Then, from generation 3 to generation 4 the lower group dies out and after this the remaining group expands and forms quite a stable pattern. You can experiment with different patterns and see which ones remain unchanged for an infinite number of generations, such as four cells in a square, and which ones die out after only a short time, such as a few isolated cells.

GENERATION 0

```

      00
     00
    0 0
   0 0 0
  0 00
 0 00
00 0 0
0

```

GENERATION 1

```

      00
     00 0
    0 00
      0 00
       0 00
      00
      00

```

GENERATION 2

```
000
  0
 00
00 0
```

```
00
 0
00
```

GENERATION 3

```
0
 00
 0 0
00 0
00
```

```
0
 0
0
```

GENERATION 4

```
00
000
 0 0
0
000
```

```
00
```

GENERATION 5

```
0 0
 0 0
00 0
0 0
00
 0
```

GENERATION 6

```
0
 0 00
00 00
 0 0
000
00
```

GENERATION 7

```
00
00 0
00
0
0 0
 0 0
```

GENERATION 8

```

      000
     0  0
    0
   00
  000
   0
  
```

GENERATION 9

```

      000
     00 0
    0
   0
  00 0
   0
  
```

The following program is an adaptation of the first and it outputs all of the information to a printer, if you have one attached, rather than the screen. The grid size has been enlarged to 15 by 25 to print out a bigger colony and also the first generation of cells is created randomly by the computer. The program is very slow but it does make a nice demonstration program for a printer.

In order to output information to the printer a file must be opened using the OPEN command as in line 10. If your printer is attached to the computer via an RS232 interface change line 10 to, OPEN #1:"RS232" otherwise the computer will give you an I/O ERROR. When you want to terminate the program press any key. This closes the file, line 300, and executes a STOP statement.

After the program listing there is a sample printout of this version of the program.

```

10 OPEN #1:"PIO"
20 RANDOMIZE
30 DIM M(17,27),N(17,27)
40 G=0
50 FOR K=2 TO 16
60 FOR Z=2 TO 26
70 IF RND>.2 THEN 90
80 N(K,Z)=1
90 NEXT Z
100 NEXT K
110 PRINT #1:"          GENERATION 0"
  
```

```

120 GOSUB 320
130 FOR K=2 TO 16
140 FOR Z=2 TO 26
150 C=M(K-1,Z-1)+M(K-1,Z)+M(K-1,Z+1)
160 C=C+M(K,Z-1)+M(K,Z+1)
170 C=C+M(K+1,Z-1)+M(K+1,Z)+M(K+1,Z+1)
180 IF (M(K,Z)=0)+(C=3)+(C=2)<>0 THEN 200
190 N(K,Z)=0
200 IF (M(K,Z)=1)+(C<>3)<>0 THEN 220
210 N(K,Z)=1
220 NEXT Z
230 NEXT K
240 G=G+1
250 PRINT #1:" "
260 PRINT #1:"          GENERATION";G
270 GOSUB 320
280 CALL KEY(3,J,JJ)
290 IF JJ=0 THEN 130
300 CLOSE #1
310 STOP
320 FOR K=1 TO 16
330 PRINT #1:" "
340 FOR Z=1 TO 26
350 M(K,Z)=N(K,Z)
360 IF M(K,Z)=1 THEN 390
370 PRINT #1:" ";
380 GOTO 400
390 PRINT #1:"0";
400 NEXT Z
410 NEXT K
420 PRINT #1:" "
430 PRINT #1:" "
440 RETURN

```

GENERATION 0

```

      0 0 0 0   0   0 0
    0   000   0   0
0   0   00  00  00
0   0   0   0   0
0   0 00 0   0 000
0   00 00   0 0 0 00
    0 00   0 0
0 0 0   0   0 0
    0   000
  00 0   0 0 0   0 0
    0   000 0   00 0
    0   000   00 0
  0 0 0   0   0
    00   0 0 0   0
      0 0   0 00 0 00 0
  
```

GENERATION 1

```

      00 00
      0 0 0   0   000
      00000   00
00   0 00  00 0 0
00   00 00 0   0 00
    0 0 0000   000 00
      0 000   000 0 000
00   0 0
    0   0 0000
  000 0 0 0 0 00
00   000 0   000 0
000  000 0   0 0
0 0   0
  000 00   00 00  0 0
      000  00
  
```

SLIMY SLITHER SUPPLIES

This program demonstrates just how effectively your computer can handle simulation information, manipulating it according to ground rules laid down by the programmer. It 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 current selling price of your product, the current cost of production and the strength of your work force and its production potential. By the way, the product that you are selling is slimy slithers which are merely trick toys.

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 slimy slithers 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.

All of the variables used in the program are set up in lines 260 to 350. The main loop from lines 360 to 460 calls the various subroutines in turn that make up the program. GOSUB 1370 provides the printout of the factory's status. GOSUB 1880 controls the hiring and firing of staff, plus any union reaction to firing staff. GOSUB 1620 controls the manufacture of your product. It allows you to set a production target and tells you the production results. GOSUB 1150 is your sales team. It provides the good news or the bad news about the current week's sales. The sales figures can include slimy slithers from your warehouse as well as the current week's production. Any unsold slimy slithers are automatically placed in storage. GOSUB 470 controls the "unpredictables". This includes events such as union demands for pay increases, rising production costs and the occasional warehouse fire.

```
10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(2)
40 PRINT TAB(5):"SLIMY SLITHER SUPPLIES"
50 FOR N=1 TO 12
60 PRINT
70 NEXT N
80 FOR M=1 TO 5
90 FOR N=1 TO 2
100 CALL SCREEN(18-N*2)
110 FOR P=5 TO 8
120 CALL COLOR(P,12+N*2,1)
130 NEXT P
140 CALL SOUND(300,-N,0)
150 FOR L=1 TO 100
160 NEXT L
170 NEXT N
180 NEXT M
190 FOR N=1 TO 8
200 CALL COLOR(N,16,1)
210 NEXT N
220 CALL COLOR(12,16,1)
230 FOR N=1 TO 500
240 NEXT N
```

```

250 REM  **VARIABLES**
260 CA=500+INT(RND*500)
270 ST=100+INT(RND*500)
280 SP=10+INT(RND*6)
290 CT=2+INT(RND*5)
300 IF CT>SP THEN 280
310 WF=7+INT(RND*10)
320 WG=12+INT(RND*SP*5)
330 PD=5+INT(RND*5)
340 WK=0
350 Z=1
360 REM  **MAIN LOOP**
370 WK=WK+1
380 GOSUB 1370
390 GOSUB 1880
400 GOSUB 1620
410 GOSUB 1150
420 GOSUB 470
430 CA=CA-INT(WG*WF)
440 IF CA+ST<1 THEN 2070
450 IF CA+ST>999999 THEN 2210
460 GOTO 370
470 REM  **UNPREDICTABLES**
480 IF RND<.45 THEN 580
490 A=INT(RND*10)+1
500 CALL SOUND(500,110,0,117,0)
510 PRINT :TAB(8);"UNIONS DEMAND A"
520 PRINT :TAB(8);STR$(A);"% PAY INCREAS
E."
530 WG=INT(100*(WG+A*WG/100))/100
540 PRINT :TAB(8);"PAY PER EMPLOYEE"
550 PRINT :TAB(9);"IS NOW $";STR$(WG)
560 FOR N=1 TO 800
570 NEXT N
580 IF RND<.85 THEN 750
590 CALL SOUND(500,-7,0)
600 PRINT :TAB(4);"WAREHOUSE FIRE DESTROYS"
610 PRINT :TAB(5);"SOME OF YOUR STOCK..."
"

```

```

620 PRINT :TAB(3); "STANDBY FOR DAMAGE RE
PORT: "
630 FOR N=1 TO 600
640 NEXT N
650 A=INT(RND*ST/2)+1
660 ST=ST-A
670 PRINT :TAB(3); "TOTAL STOCK DESTROYED
WAS"
680 PRINT :TAB(5); A; "SLIMY SLITHERS WORT
H"
690 PRINT :TAB(10); "$"; STR$(A*SP); " RETA
IL. "
700 FOR N=1 TO 300
710 NEXT N
720 PRINT :TAB(4); "STOCK ON HAND IS NOW"
;ST
730 FOR N=1 TO 800
740 NEXT N
750 IF RND>.3 THEN 970
760 CALL SOUND(500,156,0,-5,0)
770 PRINT ::TAB(7); "SUPPLIER ANNOUNCES"
780 PRINT :TAB(4); "DRAMATIC PRICE RISE !
!!"
790 FOR N=1 TO 500
800 NEXT N
810 A=.03+INT(RND*100*CT/6)/100
820 A$=STR$(A)
830 IF A>=1 THEN 850
840 A$="0"&A$
850 IF A/.1<>INT(A/.1) THEN 870
860 A$=A$&"0"
870 PRINT :TAB(4); "THE COST OF MAKING EA
CH"
880 PRINT :TAB(3); "SLIMY SLITHER HAS GON
E UP"
890 PRINT :TAB(9); "BY $"; A$; " EACH. "
900 FOR N=1 TO 300
910 NEXT N
920 CT=CT+A
930 PRINT :TAB(5); "IT NOW COSTS $"; STR$(
CT); " TO"

```

```

940 PRINT :TAB(9);"MAKE EACH ONE."
950 FOR N=1 TO 800
960 NEXT N
970 IF RND<.65 THEN 1140
980 CALL SOUND(500,523,0,659,0,784,0)
990 PRINT :TAB(4);"YOU HAVE AN OPPORTUN
ITY"
1000 PRINT :TAB(6);"TO RAISE YOUR PRICE.
"
1010 PRINT :TAB(4);"SLIMY SLITHERS NOW S
ELL"
1020 PRINT :TAB(12);"FOR $";STR$(SP)
1030 PRINT :TAB(5);"STATE YOUR PERCENTAG
E"
1040 PRINT :TAB(9);"PRICE INCREASE"
1050 INPUT A
1060 Z=Z+A
1070 SP=INT(100*(SP+A*SP/100))/100
1080 FOR N=1 TO 300
1090 NEXT N
1100 PRINT :TAB(4);"SLIMY SLITHERS NOW S
ELL"
1110 PRINT :TAB(12);"FOR $";STR$(SP)
1120 FOR N=1 TO 800
1130 NEXT N
1140 RETURN
1150 REM **SALE**
1160 PRINT :TAB(6);"STOCK ON HAND IS";S
T
1170 FOR N=1 TO 8
1180 CALL SOUND(100,440,0)
1190 FOR P=1 TO 40
1200 NEXT P
1210 NEXT N
1220 PRINT : "STAND BY FOR SALES REPORT..
."
1230 FOR N=1 TO 400
1240 NEXT N
1250 A=INT(RND*ST/ (.3+Z/10))+5+5*(SP>500
)

```

```

1260 IF A<ST THEN 1280
1270 A=ST
1280 CALL SOUND(200,1047,0)
1290 PRINT :TAB(6);"TOTAL SLIMY SLITHERS
"
1300 PRINT :TAB(10);"SOLD....";A
1310 ST=ST-A
1320 PRINT :TAB(1);"INCOME FROM SALES: $
";STR$(A*SP)
1330 CA=CA+INT(A*SP)
1340 FOR N=1 TO 800
1350 NEXT N
1360 RETURN
1370 REM **PRINT OUT**
1380 CALL SOUND(1000,131,0,262,0,523,0,-
1,0)
1390 PRINT ::TAB(5);"FACTORY REPORT: WEE
K";WK
1400 PRINT TAB(5);"~~~~~ ~~~~~ ~~~
~~~"
1410 PRINT :TAB(3);"CAPITAL ON HAND IS $
";STR$(CA)
1420 PRINT :TAB(4);"YOUR WAREHOUSE HOLDS
";ST
1430 PRINT :TAB(6);"SLIMY SLITHERS WORTH
"
1440 PRINT TAB(12);"$";STR$(ST*SP)
1450 PRINT :TAB(4);"THEY SELL FOR $";STR
$(SP);" EACH"
1460 PRINT : "AND COST $";STR$(CT);" EACH
TO MAKE."
1470 PRINT :TAB(4);"WORKFORCE IS";WF;"PE
OPLE."
1480 PRINT : "THEIR WAGES ARE $";STR$(WG)
;" EACH."
1490 PRINT :TAB(6);"THIS WEEKS WAGE BILL
"
1500 PRINT :TAB(12);"IS $";STR$(WG*WF)
1510 PRINT :TAB(9);"PRESS ANY KEY."
1520 CALL KEY(3,KY,KK)

```

```

1530 IF KK=0 THEN 1520
1540 PRINT :TAB(6);"EACH PERSON CAN MAKE
"
1550 PRINT :TAB(3);PD;"SLIMY SLITHERS A
WEEK."
1560 PRINT :TAB(7);"A TOTAL OUTPUT OF"
1570 PRINT :TAB(6);PD*WF;"SLIMY SLITHERS
"
1580 PRINT :TAB(9);"PRESS ANY KEY."
1590 CALL KEY(3,KY,KK)
1600 IF KK=0 THEN 1590
1610 RETURN
1620 PRINT ::TAB(3);"HOW MANY SLIMY SLIT
HERS DO"
1630 PRINT :TAB(7);"YOU WISH TO MAKE?"
1640 INPUT MK
1650 IF MK=0 THEN 1870
1660 IF MK*CT<=CA THEN 1700
1670 PRINT :TAB(6);"NOT ENOUGH MONEY TO"
1680 PRINT :TAB(8);"MAKE THAT MANY."
1690 GOTO 1620
1700 IF MK<=PD*WF THEN 1740
1710 PRINT :TAB(6);"NOT ENOUGH EMPLOYEES
"
1720 PRINT :TAB(7);"TO MAKE THAT MANY."
1730 GOTO 1620
1740 PRINT :TAB(8);"TARGET WEEK";WK;"IS"
1750 PRINT :TAB(7);MK;"SLIMY SLITHERS."
1760 MK=MK-INT(RND*MK/8)
1770 IF MK>0 THEN 1790
1780 MK=0
1790 FOR N=1 TO 400
1800 NEXT N
1810 PRINT :TAB(6);"TOTAL MADE IN WEEK";
WK
1820 PRINT :TAB(12);"WAS";MK
1830 ST=ST+MK
1840 CA=INT(CA-CT*MK)
1850 FOR N=1 TO 800
1860 NEXT N

```

```

1870 RETURN
1880 REM **PEOPLE**
1890 PRINT ::TAB(5);"HOW MANY PEOPLE DO
YOU"
1900 PRINT :TAB(9);"WANT TO HIRE?"
1910 INPUT A
1920 WF=WF+A
1930 PRINT :TAB(5);"TOTAL WORKFORCE IS";
WF
1940 IF A<>0 THEN 2040
1950 PRINT ::TAB(5);"HOW MANY PEOPLE DO
YOU"
1960 PRINT :TAB(9);"WANT TO FIRE?"
1970 INPUT A
1980 IF A=0 THEN 2060
1990 IF A>WF THEN 1950
2000 A=INT(RND*A)+1
2010 PRINT :TAB(5);"UNIONS WILL ONLY ALL
OW"
2020 PRINT :TAB(9);"YOU TO FIRE";A
2030 WF=WF-A
2040 FOR N=1 TO 800
2050 NEXT N
2060 RETURN
2070 FOR N=1 TO 24
2080 PRINT "#####BANKRUPT#####
"
2090 CALL SOUND(100,-7,0)
2100 NEXT N
2110 PRINT ::TAB(5);"YOU KEPT SLIMY SLIT
HER"
2120 PRINT :TAB(6);"SUPPLIES RUNNING FOR
"
2130 PRINT :TAB(11);WK;"WEEKS."
2140 PRINT :::
2150 PRINT :TAB(3);"ENTER Y FOR ANOTHER
TRY OR"
2160 PRINT :TAB(6);"N TO END THE ORDEAL.
"
2170 CALL KEY(3,KY,KK)

```

```

2180 IF KY=89 THEN 20
2190 IF KY<>78 THEN 2170
2200 STOP
2210 A$="#####"
2220 B$="#"
2230 CALL CLEAR
2240 PRINT A$
2250 PRINT B$:B$
2260 PRINT "# YOU'VE MADE A MILLION #
"
2270 PRINT B$:B$
2280 PRINT A$
2290 FOR N=1 TO 8
2300 PRINT
2310 NEXT N
2320 FOR N=1 TO 800
2330 NEXT N
2340 STOP

```

GOLD MINER

You are an old miner who is searching for a large store of gold worth millions of pounds. The complex maze of narrow passageways makes it impossible for you to find the gold without your latest invention, a gold detector. Unfortunately the readout has not yet been perfected and may be hard to interpret.

When the program is RUN the title will be displayed for some time while the maze, which is partly random and partly fixed, is set up in an array. The maze will then be displayed briefly before the game begins. You always start in the bottom left hand corner and the gold may be in any of the other corners.

Press N, S, E or W to move around the maze. The computer will only respond if the direction is clear. You may also press H at any time for help. This gives you another brief look at the maze and shows you your current position but costs you a penalty of 15 moves.

```
10 RESTORE
20 RANDOMIZE
30 DIM A(15,15)
40 CALL CHAR(136,"FFFFFFFFFFFFFFFF")
50 CALL CHAR(128,"182418083E081422")
60 CALL CHAR(129,"1824182A1C083C04")
70 CALL SCREEN(13)
80 CALL CLEAR
90 FOR N=1 TO 12
100 CALL COLOR(N,16,1)
110 NEXT N
```

```

120 CALL COLOR(13,16,1)
130 CALL COLOR(14,11,1)
140 GOSUB 1380
150 GOSUB 850
160 GOSUB 580
170 M=M+1
180 CALL CLEAR
190 PRINT : :TAB(7); "MOVE NUMBER ";M
200 PRINT : :TAB(9); "NORTH: ";
210 IF A(D+1,E)=S THEN 240
220 PRINT "WALL"
230 GOTO 250
240 PRINT "CLEAR"
250 PRINT : :TAB(9); "SOUTH: ";
260 IF A(D-1,E)=S THEN 290
270 PRINT "WALL"
280 GOTO 300
290 PRINT "CLEAR"
300 PRINT : :TAB(9); "EAST: ";
310 IF A(D,E+1)=S THEN 340
320 PRINT "WALL"
330 GOTO 350
340 PRINT "CLEAR"
350 PRINT : :TAB(9); "WEST: ";
360 IF A(D,E-1)=S THEN 390
370 PRINT "WALL"
380 GOTO 400
390 PRINT "CLEAR"
400 PRINT : :TAB(4); "GOLD DETECTOR READS
";100*(ABS(Z-D)+ABS(Y-E))+Y-E
410 PRINT : :TAB(8); "WHICH DIRECTION?"
420 PRINT : :TAB(6); "NORTH, SOUTH, EAST,":
TAB(9); "WEST, HELP."
430 CALL KEY(3,K,J)
440 IF (K=78)+(K=83)+(K=69)+(K=87)+(K=72)
)=0 THEN 430
450 IF (K=78)+(A(D+1,E)=X)=-2 THEN 430
460 IF (K=83)+(A(D-1,E)=X)=-2 THEN 430
470 IF (K=69)+(A(D,E+1)=X)=-2 THEN 430
480 IF (K=87)+(A(D,E-1)=X)=-2 THEN 430

```

```

490 IF K<>72 THEN 520
500 GOSUB 580
510 GOTO 560
520 D=D+(K=83)-(K=78)
530 E=E+(K=87)-(K=69)
540 SND=262*(K=83)+330*(K=78)+392*(K=87)
+440*(K=69)
550 CALL SOUND(-100,-SND,0)
560 IF (Z=D)+(Y=E)=-2 THEN 1210
570 GOTO 170
580 CALL SCREEN(2)
590 CALL COLOR(1,16,2)
600 CALL CLEAR
610 FOR B=15 TO 1 STEP -1
620 PRINT TAB(8);
630 FOR C=1 TO 15
640 IF A(B,C)=S THEN 670
650 PRINT CHR$(136);
660 GOTO 730
670 IF (B=D)+(C=E)=-2 THEN 700
680 PRINT " ";
690 GOTO 730
700 PRINT CHR$(128);
710 CALL SOUND(-100,262,0)
720 GOTO 740
730 CALL SOUND(-500,-5,30-C*2)
740 NEXT C
750 PRINT
760 NEXT B
770 M=M+15
780 FOR J=1 TO 1000
790 NEXT J
800 CALL SCREEN(13)
810 CALL COLOR(1,16,13)
820 CALL CLEAR
830 A(D,E)=S
840 RETURN
850 B=INT(RND*3)+1
860 Z=14
870 Y=14

```

```

880 IF B<>2 THEN 900
890 Y=2
900 IF B<>3 THEN 920
910 Z=2
920 X=1
930 S=2
940 FOR B=1 TO 15
950 FOR C=1 TO 15
960 A(B,C)=X
970 IF RND<.9 THEN 990
980 A(B,C)=S
990 IF (C<2)+(C>14)+(B<2)+(B>14)=0 THEN
1010
1000 A(B,C)=X
1010 NEXT C
1020 NEXT B
1030 D=2
1040 E=2
1050 FOR F=1 TO 69
1060 READ B,C
1070 A(B,C)=S
1080 NEXT F
1090 M=-15
1100 RETURN
1110 DATA 2,2,2,3,2,4,2,5,2,6,2,7
1120 DATA 3,7,4,7,5,7,5,6,5,5,5,4,5,3,6,
3
1130 DATA 7,3,7,4,7,5,7,6,7,7,7,8,7,9,9,
8
1140 DATA 9,9,10,8,10,7,10,6,10,5,10,4,8
,8
1150 DATA 10,3,11,3,12,3,13,3,14,3,14,2,
7,10
1160 DATA 6,10,5,10,4,10,3,10,2,10,2,11,
2,12
1170 DATA 2,13,2,14,6,11,6,12,6,13,6,14,
7,12
1180 DATA 14,12,8,12,8,14,9,12,9,13,9,14
,10,12
1190 DATA 11,9,11,10,11,11,11,12,12,9,13

```

```

,9,13,10
1200 DATA 13,11,13,12,13,13,13,14,14,14
1210 CALL CLEAR
1220 A$="***** YOU FOUND IT *****"
1230 FOR N=4 TO 29
1240 CALL HCHAR(12,N,ASC(SEG$(A$,N-3,1))
,1)
1250 NEXT N
1260 FOR N=110 TO 320 STEP 30
1270 CALL SOUND(250,N,0)
1280 CALL SOUND(250,N,0,N+30,0)
1290 CALL SOUND(250,N,0,N+30,0,N+60,0)
1300 NEXT N
1310 PRINT TAB(6);"YOU TOOK ";M;" MOVES"
1320 FOR N=1 TO 500
1330 NEXT N
1340 FOR N=1 TO 25
1350 PRINT
1360 NEXT N
1370 GOTO 10
1380 A$="GOLD MINER"
1390 FOR N=2 TO 32
1400 CC=129
1410 IF N/2=INT(N/2) THEN 1430
1420 CC=128
1430 CALL HCHAR(12,N-1,32,1)
1440 CALL HCHAR(12,N,CC,1)
1450 IF (N<13)+(N>22)<>0 THEN 1470
1460 CALL HCHAR(12,N-1,ASC(SEG$(A$,N-12,
1)),1)
1470 CALL SOUND(-200,N*55,0)
1480 NEXT N
1490 CALL HCHAR(12,32,32,1)
1500 RETURN

```

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 + signs at the top of the screen, and you are the * signs at the bottom.

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 top of it, 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 will ask you "MOVE FROM ?? TO ??". You then enter the co-ordinates of the piece you wish to move, starting with the number down the side of the board, followed by the number along the bottom. Do exactly the same for the square you wish to move to.

The program uses the DATA statements in lines 1290 to 1380 to set up the board. The lower case letters in these lines (a and h) are entered from the keyboard after releasing the ALPHA LOCK key. The variable CS is the computer's score and the variable HS is the human player's score.

```

10 RANDOMIZE
20 CALL CLEAR
30 CALL COLOR(2,5,16)
40 CALL COLOR(9,2,2)
50 CALL COLOR(10,16,16)
60 DIM S$(12,13)
70 PRINT TAB(11);"HAMADRYAD"
80 PRINT :::
90 FOR A=1 TO 10
100 PRINT TAB(10);
110 READ B$
120 FOR B=1 TO 11
130 S$(A,B)=SEG$(B$,B,1)
140 PRINT S$(A,B);
150 NEXT B
160 PRINT
170 NEXT A
180 PRINT
190 CS=0
200 HS=0
210 PRINT :TAB(7);"MOVE FROM      TO"
220 PRINT ::
230 PRINT :TAB(10);"COMPUTER: 0"
240 PRINT :TAB(12);"HUMAN: 0"
250 GOTO 320
260 GOSUB 1070
270 IF CS<>6 THEN 320
280 CALL CLEAR
290 PRINT TAB(11);">>>I WIN<<<"
300 PRINT :::
310 STOP
320 CALL HCHAR(17,19,63,2)
330 CALL HCHAR(17,25,63,2)
340 K=19
350 GOSUB 1190
360 A=X-48
370 K=20
380 GOSUB 1190
390 B=X-48
400 K=25

```

```

410 GOSUB 1190
420 C=X-48
430 K=26
440 GOSUB 1190
450 D=X-48
460 IF (ABS(A-C)=1)+(ABS(B-D)=1)<>-2 THE
N 320
470 IF (S$(A+1,B+1)<>"*")+ (S$(C+1,D+1)="
*)<>0 THEN 320
480 HS=HS-(S$(C+1,D+1)="+")
490 S$(A+1,B+1)="h"
500 S$(C+1,D+1)="*"
510 GOSUB 1070
520 IF HS<>6 THEN 570
530 CALL CLEAR
540 PRINT TAB(9); ">>>YOU WIN<<<"
550 PRINT :::
560 STOP
570 A$="*"
580 GOSUB 790
590 IF FL=1 THEN 630
600 A$="h"
610 GOSUB 790
620 IF G=0 THEN 610
630 S$(E,F)="h"
640 CS=CS-(S$(E+G,F+H)="*")
650 S$(E+G,F+H)="+"
660 X=E+47
670 K=19
680 GOSUB 1240
690 X=F+47
700 K=20
710 GOSUB 1240
720 X=E+G+47
730 K=25
740 GOSUB 1240
750 X=F+H+47
760 K=26
770 GOSUB 1240
780 GOTO 260

```

```

790 E=2
800 F=2
810 G=0
820 H=0
830 FL=0
840 IF S$(E,F)<>"+" THEN 1010
850 IF S$(E+1,F+1)<>A$ THEN 880
860 G=1
870 H=1
880 IF S$(E+1,F-1)<>A$ THEN 910
890 G=1
900 H=-1
910 IF S$(E-1,F+1)<>A$ THEN 940
920 G=-1
930 H=1
940 IF S$(E-1,F-1)<>A$ THEN 970
950 G=-1
960 H=-1
970 IF G=0 THEN 1010
980 FL=1
990 IF (A$="*")+ (RND<.1)<>0 THEN 1060
1000 G=0
1010 E=E+1
1020 IF E<11 THEN 1050
1030 E=2
1040 F=F+1
1050 IF F<12 THEN 830
1060 RETURN
1070 CALL SOUND(100,440,0)
1080 CALL SOUND(100,294,0)
1090 CALL SOUND(100,494,0)
1100 CALL SOUND(100,330,0)
1110 FOR A=2 TO 9
1120 FOR B=2 TO 10
1130 CALL HCHAR(A+4,11+B,ASC(S$(A,B)),1)
1140 NEXT B
1150 NEXT A
1160 CALL HCHAR(21,22,48+CS,1)

```

```
1170 CALL HCHAR(23,21,48+HS,1)
1180 RETURN
1190 CALL KEY(3,X,KK)
1200 IF (X<49)+(X>57)<>0 THEN 1190
1210 IF X<>57 THEN 1240
1220 IF K=19 THEN 1190
1230 IF K=25 THEN 1190
1240 CALL SOUND(100,784,0)
1250 CALL HCHAR(17,K,X,1)
1260 CALL KEY(3,KY,KK)
1270 IF KK<>0 THEN 1260
1280 RETURN
1290 DATA " 123456789 "
1300 DATA "1+a+a+a+a+1"
1310 DATA "2a+a+a+a+a2"
1320 DATA "3+a+a+a+a+3"
1330 DATA "4ahahahaha4"
1340 DATA "5hahahahah5"
1350 DATA "6a*a*a*a*a6"
1360 DATA "7*a*a*a*a*a7"
1370 DATA "8a*a*a*a*a8"
1380 DATA " 123456789 "
```

LUNAR DESCENT

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 onboard 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 towards the surface. It is possible to make your craft ascend again if you apply enough negative thrust.

The main loop from lines 180 to 570 displays the various information on the screen and inputs the amount of thrust you wish to apply and the amount of time which you wish to apply it for. The crash is performed by lines 580 to 660 and the landing from lines 670 to 720. Lines 730 to 1110 tell you if there is a malfunction and, when you enter the access code, tell you exactly what the problem is.

```
10 RANDOMIZE
20 M=0
30 T=0
40 S=0
50 H=3000
60 F=600/(1+INT(RND*3))
```

```

70 Q=-17
80 B=1
90 CALL CLEAR
100 PRINT TAB(3);"WELCOME TO LUNAR DESCE
NT."
110 PRINT :TAB(3);"YOUR MISSION: TO PILO
T THE"
120 PRINT :TAB(3);"FIRST MARS LANDING CR
AFT."
130 PRINT :TAB(11);"GOOD LUCK."
140 FOR N=1 TO 1500
150 NEXT N
160 CALL CLEAR
170 GOTO 420
180 PRINT :TAB(4);"POSITIVE NUMBERS TOWA
RDS"
190 PRINT :TAB(3);"SURFACE, NEGATIVE NUM
BERS"
200 PRINT :TAB(7);"AWAY FROM SURFACE."
210 INPUT Z
220 IF (Z<-50)+(Z>50)<>0 THEN 560
230 PRINT :TAB(4);"FOR HOW LONG? (SECON
D)"
240 INPUT E
250 IF E<1 THEN 240
260 CALL CLEAR
270 PRINT :TAB(11);"THRUST";Z
280 PRINT :TAB(9);"FOR";E;"SECONDS."
290 FOR N=1 TO 700
300 NEXT N
310 CALL CLEAR
320 T=T+E
330 S=S+10+3*E*INT((Z+1)/B)
340 IF Z=0 THEN 360
350 F=F-3*E*ABS(Z*(1+INT(RND*2))+1)
360 H=H-E*(S+1+INT(RND*5))
370 IF F<0 THEN 580
380 IF (H>100)+(H<0)+(S>40)=0 THEN 670
390 IF H<=0 THEN 580
400 IF (RND<.1)+(M<>2)<>-2 THEN 420

```

```

410 GOSUB 730
420 CALL SOUND(100,260,0)
430 CALL SOUND(100,300,0)
440 CALL SOUND(400,280,0)
450 PRINT ::TAB(3);"HEIGHT ABOVE SURFACE
:";H
460 IF Q=-17 THEN 500
470 Q=Q-1-INT(RND*16)
480 IF Q<0 THEN 580
490 PRINT :TAB(10);"AIR LEFT:";Q
500 PRINT :TAB(8);"RATE OF FALL:";S
510 IF B=1 THEN 540
520 PRINT :TAB(9);"W A R N I N G"
530 PRINT :TAB(9);"THRUST ERRATIC"
540 PRINT :TAB(8);"ENERGY LEFT:";F
550 PRINT :TAB(6);"TIME IN FLIGHT:";T
560 PRINT ::TAB(6);"THRUST? (-50 TO 50)
"
570 GOTO 180
580 CALL CLEAR
590 PRINT TAB(7);"*** C R A S H ***"
600 CALL SOUND(1000,-7,0)
610 IF (Q>0)+(Q=-17)<>0 THEN 630
620 PRINT ::TAB(5);"AIR SUPPLY EXHAUSTED
!!"
630 IF F>0 THEN 650
640 PRINT ::TAB(7);"ENERGY EXHAUSTED!!"
650 PRINT ::TAB(7);"HIT SURFACE AT";ABS(
S)
660 STOP
670 CALL CLEAR
680 PRINT TAB(4);"<<<< PERFECT LANDING >>
>"
690 PRINT ::TAB(7);"FINAL VELOCITY";ABS(
S)
700 CALL SOUND(1000,400,0)
710 CALL SOUND(1000,300,0,350,0)
720 STOP
730 CALL CLEAR
740 M=M+1

```

```

750 U=1+INT(RND*10)
760 FOR V=1 TO 4
770 PRINT :TAB(6);"### D A N G E R ###"
780 NEXT V
790 PRINT :TAB(8);"MISSION CONTROL,"
800 PRINT :TAB(7);"WE HAVE A PROBLEM."
810 FOR N=1 TO 800
820 NEXT N
830 CALL CLEAR
840 FOR NN=1 TO 10
850 CALL SOUND(100,110+RND*400,0)
860 PRINT :TAB(1+RND*16);"MALFUNCTION"
870 CALL SOUND(100,510+RND*400,0)
880 NEXT NN
890 CALL CLEAR
900 PRINT TAB(7);"USE ACCESS CODE";U
910 PRINT :TAB(10);"FOR DETAILS."
920 INPUT V
930 CALL CLEAR
940 IF U<>V THEN 580
950 A=1+INT(RND*2)
960 IF Q=-17 THEN 980
970 A=2
980 IF B=1 THEN 1000
990 A=1
1000 ON A GOSUB 1060,1090
1010 PRINT :TAB(5);"PRESS ENTER TO RETU
RN"
1020 PRINT :TAB(11);"TO FLIGHT."
1030 INPUT V$
1040 CALL CLEAR
1050 RETURN
1060 Q=INT(RND*19)+101
1070 PRINT :TAB(4);"OXYGEN METER UNRELIA
BLE."
1080 RETURN
1090 B=B+2+INT(RND*3)
1100 PRINT :TAB(4);"THRUST CONTROL ERRAT
IC."
1110 RETURN

```

FLIPPA

This 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. 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 flips the four squares in the middle of the outside rows.

Lines 220 to 280 randomly make the squares either black or white. Lines 300 to 340 check to see if the puzzle is correct. Lines 350 to 430 display the number of the move that you are on using CALL HCHAR so that the display remains static. When you press a key from one to nine line 590 goes to the corresponding lines in the program that flip the correct squares.

```
10 RANDOMIZE
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 CALL CHAR(136,"FFFFFFFFFFFFFFFF")
40 CALL SCREEN(14)
50 FOR N=1 TO 8
60 CALL COLOR(N,12,1)
70 NEXT N
80 CALL COLOR(13,2,1)
90 CALL COLOR(14,16,1)
100 CALL CLEAR
```

```

110 PRINT TAB(9); "MOVE NUMBER: "
120 FOR N=1 TO 5
130 PRINT
140 NEXT N
150 FOR N=4 TO 10 STEP 3
160 FOR P=1 TO 3
170 CALL HCHAR(N,5+P*3,44+N+P,1)
180 NEXT P
190 NEXT N
200 DIM A(10),F(4)
210 M=0
220 Q=128
230 X=136
240 FOR C=1 TO 9
250 A(C)=0
260 IF RND<.5 THEN 280
270 A(C)=X
280 NEXT C
290 GOSUB 500
300 N=0
310 FOR C=1 TO 9
320 N=N-(A(C)=X)
330 NEXT C
340 IF (N=1)+(A(5)=X)=-2 THEN 1000
350 M=M+1
360 A$=STR$(M)&"  "
370 C=ASC(SEG$(A$,1,1))
380 D=ASC(SEG$(A$,2,1))
390 E=ASC(SEG$(A$,3,1))
400 CALL HCHAR(18,23,C,1)
410 CALL HCHAR(18,24,D,1)
420 CALL HCHAR(18,25,E,1)
430 CALL SOUND(100,500,0)
440 CALL KEY(3,KY,KK)
450 IF (KY<49)+(KY>57)<>0 THEN 440
460 CALL SOUND(100,110,0)
470 N=KY-48
480 GOSUB 560
490 GOTO 290
500 FOR N=4 TO 10 STEP 3

```

```

510 FOR P=1 TO 3
520 CALL HCHAR(N, 17+P*3, A(P+(N-4)), 1)
530 NEXT P
540 NEXT N
550 RETURN
560 IF A(N)=X THEN 990
570 F(3)=10
580 F(4)=10
590 ON N GOTO 600,640,670,710,740,790,82
0,860,890
600 F(1)=2
610 F(2)=4
620 F(3)=5
630 GOTO 920
640 F(1)=1
650 F(2)=3
660 GOTO 920
670 F(1)=2
680 F(2)=5
690 F(3)=6
700 GOTO 920
710 F(1)=1
720 F(2)=7
730 GOTO 920
740 F(1)=2
750 F(2)=4
760 F(3)=6
770 F(4)=8
780 GOTO 920
790 F(1)=3
800 F(2)=9
810 GOTO 920
820 F(1)=4
830 F(2)=5
840 F(3)=8
850 GOTO 920
860 F(1)=7
870 F(2)=9
880 GOTO 920
890 F(1)=5

```

```
900 F(2)=6
910 F(3)=8
920 FOR G=1 TO 4
930 IF A(F(G))<>X THEN 960
940 A(F(G))=0
950 GOTO 970
960 A(F(G))=X
970 NEXT G
980 A(N)=X
990 RETURN
1000 CALL CLEAR
1010 PRINT TAB(6);"YOU SOLVED IT IN";M
1020 FOR N=1 TO 11
1030 PRINT
1040 NEXT N
1050 CALL SOUND(400,262,0)
1060 CALL SOUND(400,247,0)
1070 CALL SOUND(400,294,0)
1080 CALL SOUND(1000,262,0)
1090 END
```

STAR PATROL

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 and destroy them.

You have a limited amount of energy available and your computer will keep you informed of your energy level. Your space ship is equipped 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 which direction. The long range scanner will give you a positive or negative reading for a square two squares away from you in a specified direction. Both types of scanners use up precious energy. Long range scanning uses more energy than short range scanning.

An alien ship will only show up on the screen when it has been destroyed. If you hit a ship with your fire it will appear as a white square. If you fire your gun and miss an X will appear in the square that you aimed at. Your ship is shown as a white triangle. Any attempt to move your ship through a square occupied by an alien ship which has not yet been destroyed will result in a collision.

Your computer will give you a list of instructions telling you how to move your ship, use your scanners and fire your guns. Short range scanners become inoperable in the outside edges of the grid and long range scanners become inoperable from a square just before the edge.

Lines 250 to 300 set up the galaxy and place the alien ships in random positions. GOSUB 1690 in line 410 draws the galaxy on the screen. The main loop is from lines 420 to 490 and this displays the energy you have left and inputs your command.

One of three subroutines is then called which will either scan, move or fire. The scan routine is from lines 500 to 880 and this will perform a long or short scan. The move routine is from lines 890 to 1050 and this will move the ship in the direction of your choice. It will then call the routine starting at line 1690 which redraws the galaxy on the screen. If your ship has collided with an enemy ship there will be a spectacular display and the game will end. The fire routine is from lines 1060 to 1370 and again you must enter a specified direction in which you wish to fire. You will be told if you hit or miss and you will also be told if the enemy decides to fire back. Taking a direct hit from the enemy could use up a lot of your energy. Lines 1380 to 1500 end the program if you have run out of energy and give you your command rating for the mission. Your position is worked out and displayed by lines 1840 to 2010 and you may be at any one of seven stars in the galaxy.

```

10 RANDOMIZE
20 CALL CLEAR
30 CALL CHAR(128,"000000C0E0F8FEFF")
40 CALL CHAR(129,"007E424242427E00")
50 CALL CHAR(130,"0000001818000000")
60 CALL COLOR(13,16,2)
70 CALL SCREEN(8)
80 A$="STAR PATROL"
90 FOR N=2 TO 10 STEP 2
100 PRINT :TAB(3+N);SEG$(A$,N/2,1);TAB(2
7-N);SEG$(A$,12-N/2,1)
110 NEXT N
120 PRINT :TAB(15);"P"
130 FOR N=2 TO 10 STEP 2
140 PRINT :TAB(15-N);SEG$(A$,6-N/2,1);TA
B(15+N);SEG$(A$,6+N/2,1)
150 NEXT N
160 PRINT ::
170 FOR N=1 TO 600
180 NEXT N
190 CALL SCREEN(5)
200 FOR N=1 TO 8

```

```

210 CALL COLOR(N,8,1)
220 NEXT N
230 FOR N=1 TO 600
240 NEXT N
250 DIM A(10,10),B(10,10)
260 FOR N=1 TO 15
270 X=1+INT(RND*10)
280 Y=1+INT(RND*10)
290 A(X,Y)=1
300 NEXT N
310 C=5
320 D=5
330 A(C,D)=0
340 B(C,D)=2
350 H=0
360 AL=0
370 Z$="RUSDORAN"
380 E=2500
390 CALL SOUND(500,250,0)
400 CALL CLEAR
410 GOSUB 1690
420 PRINT :TAB(6);"REMAINING ENERGY";IN
T(E)
430 IF E<1 THEN 1380
440 IF AL=0 THEN 460
450 PRINT :TAB(12);"TALLY";AL
460 PRINT :TAB(6);"1-SCAN 2-MOVE 3-FIRE"
470 INPUT K
480 ON K GOSUB 500,890,1060
490 GOTO 420
500 PRINT : "SCANNER:SHORT(1) OR LONG(2)?
"
510 INPUT K
520 IF (K=1)+(K=2)=0 THEN 510
530 IF (K=1)+((C=10)+(D=10)<>0)<>-2 THEN
570
540 PRINT :TAB(6);"SHORT RANGE SCAN NOT"

```

```

550 PRINT :TAB(4);"POSSIBLE IN THIS SECT
OR"
560 GOTO 860
570 IF (K=2)+(C>8)+(D>8)<>0<>-2 THEN 6
10
580 PRINT : "LONG RANGE SCAN NOT POSSIBLE
"
590 PRINT :TAB(9);"IN THIS SECTOR"
600 GOTO 860
610 E=E-10*K
620 IF K=2 THEN 700
630 F=(A(C+1,D)=1)+(A(C+1,D+1)=1)+(A(C+1
,D-1)=1)+(A(C,D-1)=1)
640 F=F+(A(C,D+1)=1)+(A(C-1,D)=1)+(A(C-1
,D+1)=1)+(A(C-1,D-1)=1)
650 IF F=0 THEN 680
660 PRINT :TAB(9);Z$;" NEAR"
670 GOTO 860
680 PRINT :TAB(9);"SCAN NEGATIVE"
690 GOTO 860
700 PRINT :TAB(6);"DIRECTION:N,S,E,W ?"
710 INPUT A$
720 Z=0
730 IF (A$="N")+ (A(C-2,D)=1)<>-2 THEN 75
0
740 Z=1
750 IF (A$="S")+ (A(C+2,D)=1)<>-2 THEN 77
0
760 Z=1
770 IF (A$="E")+ (A(C,D+2)=1)<>-2 THEN 79
0
780 Z=1
790 IF (A$="W")+ (A(C,D-2)=1)<>-2 THEN 81
0
800 Z=1
810 PRINT :TAB(6);"SCANNER IS:";
820 IF Z=0 THEN 840
830 PRINT "POSITIVE"
840 IF Z=1 THEN 860
850 PRINT "NEGATIVE"
860 FOR N=1 TO 800

```

```

870 NEXT N
880 RETURN
890 E=E-50
900 B(C,D)=H
910 PRINT :TAB(6);"DIRECTION? (N,S,E,W)"
920 INPUT A$
930 C=C+(A$="N")-(A$="S")
940 C=C+(C=11)-(C=0)
950 D=D+(A$="W")-(A$="E")
960 D=D+(D=11)-(D=0)
970 CALL SOUND(300,250,0)
980 PRINT :TAB(9);"LOCATION";C;D
990 FOR N=1 TO 800
1000 NEXT N
1010 IF A(C,D)=1 THEN 1510
1020 H=B(C,D)
1030 B(C,D)=2
1040 GOSUB 1690
1050 RETURN
1060 G=C
1070 F=D
1080 PRINT :TAB(3);"DIRECTION OF FIRE N,
S,E,W?"
1090 INPUT A$
1100 G=G+(A$="N")-(A$="S")
1110 G=G+(G=11)-(G=0)
1120 F=F+(A$="W")-(A$="E")
1130 F=F+(F=11)-(F=0)
1140 E=E-100
1150 IF A(G,F)<>1 THEN 1210
1160 PRINT :TAB(3);"YOU HIT THE ";Z$;" S
HIP."
1170 AL=AL+1
1180 A(G,F)=4
1190 B(G,F)=4
1200 GOTO 1340
1210 B(G,F)=3
1220 PRINT :TAB(11);"YOU MISSED"
1230 FOR N=1 TO 800
1240 NEXT N

```

```

1250 PRINT :TAB(7);"THE ";Z$;"S ARE"
1260 PRINT :TAB(9);"SHOOTING BACK!"
1270 FOR N=1 TO 600
1280 NEXT N
1290 IF RND>.5 THEN 1330
1300 PRINT :TAB(5);"THE ENEMY HAS HIT US
!!"
1310 E=E-100*RND
1320 GOTO 1340
1330 PRINT :TAB(9);"THEY MISSED US"
1340 FOR N=1 TO 500
1350 NEXT N
1360 GOSUB 1690
1370 RETURN
1380 CALL CLEAR
1390 PRINT :TAB(5);"ENERGY BANKS EXHAUST
ED"
1400 PRINT :TAB(4);"YOU DESTROYED";AL;Z$
1410 PRINT :TAB(13);"SHIPS."
1420 PRINT ::"ON THIS MISSION YOUR COMMA
ND"
1430 PRINT :TAB(10);"RATING IS";1000*AL/
8
1440 FOR N=110 TO 200
1450 CALL SOUND(-300,N,0)
1460 NEXT N
1470 FOR N=200 TO 110 STEP -10
1480 CALL SOUND(-300,N,0)
1490 NEXT N
1500 STOP
1510 FOR P=1 TO 2
1520 K=-130*(P=1)-32*(P=2)
1530 FOR N=1 TO 12
1540 CALL HCHAR(13-N,17-N,K,N*2)
1550 CALL VCHAR(13-N,16+N,K,N*2)
1560 CALL HCHAR(12+N,17-N,K,N*2)
1570 CALL VCHAR(13-N,17-N,K,N*2)
1580 NEXT N
1590 FOR N=1 TO 4
1600 CALL VCHAR(1,5-N,K,24)

```

```

1610 CALL VCHAR(1,28+N,K,24)
1620 NEXT N
1630 NEXT P
1640 PRINT "YOUR SHIP HAS COLLIDED WITH"
1650 PRINT :TAB(8);"A ";Z$;" SHIP."
1660 FOR N=1 TO 800
1670 NEXT N
1680 STOP
1690 PRINT ::
1700 FOR Q=1 TO 10
1710 PRINT TAB(11);
1720 FOR P=1 TO 10
1730 IF B(Q,P)<>0 THEN 1750
1740 PRINT CHR$(130);
1750 IF B(Q,P)<>2 THEN 1770
1760 PRINT CHR$(128);
1770 IF B(Q,P)<>3 THEN 1790
1780 PRINT "X";
1790 IF B(Q,P)<>4 THEN 1810
1800 PRINT CHR$(129);
1810 NEXT P
1820 PRINT
1830 NEXT Q
1840 Q=C*D
1850 PRINT :TAB(6);"POSITION: ";
1860 A$="RIGEL"
1870 IF Q>90 THEN 1990
1880 A$="SIRIUS"
1890 IF Q>80 THEN 1990
1900 A$="ALTAIR"
1910 IF Q>70 THEN 1990
1920 A$="CANOPUS"
1930 IF Q>50 THEN 1990
1940 A$="VEGA"
1950 IF Q>30 THEN 1990
1960 A$="PROCYON"
1970 IF Q>10 THEN 1990
1980 A$="ALDERBARAN"
1990 PRINT A$
2000 PRINT :TAB(1);"SECTOR";C;D
2010 RETURN

```

TREASURE TRAIL

This game puts you deep underground in a dark cave system. The object of the game is to gain as much wealth as possible whilst avoiding hazards such as pools of quicksand and fearsome monsters. There are also magic caves which will transport you to a different part of the cave system if you visit them.

You have a bow and arrows to protect you from the flesh eating monsters which you may use when you are in a cave adjoining an inhabited cave. If you are unfortunate enough to land up in the same cave as a monster it will probably eat you but occasionally it may become frightened and run away from you.

You will be shown a map of the cave system at the beginning of the game and at various other times but for most of the time you will be in the dark which makes the game a great deal harder. When the prompt appears on the screen enter N, S, E or W to move from cave to cave or press F to fire your bow and arrow. The computer will tell you if something is nearby but it will not say which direction it is in.

Treasure is displayed on the map as a \$ sign, quicksand is shown with a Q, you are shown as a small face and an empty space is shown as a dot. A letter M can either mean a monster or magic so beware.

The game ends when you run out of time, or even sooner if you don't avoid the monsters and quicksand.

```

10 RANDOMIZE
20 CALL CHAR(128,"E7E7E7001818423C")
30 CALL CHAR(129,"007E7E7E7E7E7E00")
40 CALL CLEAR
50 CALL SCREEN(2)
60 FOR T=1 TO 13
70 CALL COLOR(T,16,2)
80 NEXT T
90 GT=21+INT(RND*10)
100 CALL CLEAR
110 PRINT TAB(3);"#### TREASURE TRAIL
####"
120 FOR N=1 TO 12
130 PRINT
140 NEXT N
150 DIM A(100)
160 H=1
170 L=0
180 G=0
190 AR=6
200 FOR B=1 TO 100
210 A(B)=46
220 IF (B<12)+(B>90)+(10*INT(B/10)=B)+(1
0*INT(B/10)=B-1)=0 THEN 240
230 A(B)=129
240 NEXT B
250 FOR B=1 TO 5
260 RESTORE
270 FOR D=1 TO 5
280 Z=INT(RND*76)+12
290 IF A(Z)=129 THEN 280
300 READ C
310 A(Z)=C
320 NEXT D
330 NEXT B
340 FOR B=1 TO 8
350 READ P(B)
360 NEXT B
370 E=55
380 A(E)=128

```

```

390 CALL CLEAR
400 GOSUB 2120
410 CALL CLEAR
420 PRINT :TAB(12);"CAVE";E
430 IF G<=0 THEN 460
440 PRINT :TAB(6);"YOU HAVE $";G;"WORTH"
450 PRINT :TAB(12);"OF GOLD"
460 GOSUB 1350
470 IF E=55 THEN 490
480 GOSUB 2100
490 CALL SOUND(500,131,0)
500 PRINT :TAB(7);"N, S, E, W OR F ?"
510 CALL KEY(3,J,JJ)
520 IF (J=78)+(J=83)+(J=69)+(J=87)+(J=70)
)=0 THEN 510
530 IF ((J=78)+(A(E-10)=129)=-2)+((J=83)
)+(A(E+10)=129)=-2)<>0 THEN 550
540 IF ((J=69)+(A(E+1)=129)=-2)+((J=87)+
(A(E-1)=129)=-2)=0 THEN 590
550 PRINT :TAB(10);"BLOCKED CAVE"
560 FOR T=1 TO 300
570 NEXT T
580 GOTO 480
590 A(E)=46
600 E=E+10*(J=78)+(J=87)-10*(J=83)-(J=69
)
610 IF J<>70 THEN 630
620 GOSUB 1560
630 IF A(E)<>109 THEN 650
640 GOSUB 740
650 IF A(E)<>77 THEN 670
660 GOSUB 870
670 IF A(E)<>81 THEN 690
680 GOSUB 1120
690 IF A(E)<>36 THEN 710
700 GOSUB 1210
710 H=H+1
720 IF H>GT THEN 1960
730 GOTO 410
740 PRINT :TAB(9);"////////-\\\\"

```

```

750 PRINT TAB(9);"\\ \\ MAGIC ///"
760 PRINT TAB(10);"\\ \\ \\ _///// "
770 A(E)=46
780 E=INT(RND*76)+12
790 IF A(E)=129 THEN 780
800 FOR T=3 TO 16
810 FOR TT=1 TO 50
820 NEXT TT
830 CALL SCREEN(T)
840 NEXT T
850 CALL SCREEN(2)
860 RETURN
870 CALL CLEAR
880 PRINT TAB(5);"?? MONSTER HERE! ??"
890 FOR T=1 TO 4
900 FOR TT=30 TO 0 STEP -1
910 CALL SOUND(-200,-5,TT)
920 NEXT TT
930 NEXT T
940 M=RND
950 IF M>.2 THEN 980
960 PRINT :TAB(7);"IT IS RUNNING AWAY"
970 GOTO 1060
980 PRINT :TAB(6);"IT HAS SEEN YOU...."
990 FOR T=1 TO 4
1000 FOR TT=30 TO 0 STEP -1
1010 CALL SOUND(-200,349+(90-TT*3),TT)
1020 NEXT TT
1030 NEXT T
1040 IF M<.85 THEN 1090
1050 PRINT :TAB(11);"AND FLEES"
1060 FOR T=1 TO 400
1070 NEXT T
1080 RETURN
1090 PRINT :TAB(8);"AND EATS YOU !!!"
1100 CALL SOUND(2000,110,0,-7,0)
1110 GOTO 1980
1120 CALL CLEAR
1130 FOR J=1 TO 20
1140 PRINT TAB(5*J);"HORRORS...";

```

```

1150 PRINT "QUICKSAND"
1160 CALL SOUND(-200,210-J*5,0)
1170 NEXT J
1180 FOR T=1 TO 600
1190 NEXT T
1200 GOTO 1980
1210 CALL CLEAR
1220 FOR J=1 TO 20
1230 PRINT TAB(5*J);"* WEALTH *"
1240 CALL SOUND(-200,210+5*J,0)
1250 NEXT J
1260 CALL CLEAR
1270 K=INT(RND*100)+100
1280 PRINT :TAB(6);"YOU HAVE FOUND GOLD"
1290 PRINT :TAB(10);"WORTH $";K
1300 G=G+K
1310 CALL SOUND(3000,131,0,165,0,196,0)
1320 FOR T=1 TO 400
1330 NEXT T
1340 RETURN
1350 Y=1
1360 L=A(E+P(Y))
1370 IF (L=46)+(L=32)=0 THEN 1420
1380 IF Y=8 THEN 1410
1390 Y=Y+1
1400 GOTO 1360
1410 RETURN
1420 PRINT :TAB(10);"NEARBY IS..."
1430 IF L<>129 THEN 1450
1440 PRINT :TAB(12);"NO PATH"
1450 IF L<>109 THEN 1470
1460 PRINT :TAB(13);"MAGIC"
1470 IF L<>77 THEN 1490
1480 PRINT :TAB(12);"MONSTER"
1490 IF L<>81 THEN 1510
1500 PRINT :TAB(11);"QUICKSAND"
1510 IF L<>36 THEN 1530
1520 PRINT :TAB(14);"GOLD"
1530 FOR T=1 TO 400
1540 NEXT T

```

```

1550 RETURN
1560 AR=AR-1
1570 IF AR<>0 THEN 1600
1580 PRINT :TAB(9); "NO ARROWS LEFT"
1590 RETURN
1600 PRINT :TAB(9); AR; "ARROWS LEFT"
1610 SS=0
1620 PRINT :TAB(6); "FIRE ARROW IN WHICH"
1630 PRINT :TAB(11); "DIRECTION?"
1640 CALL KEY(3,J,JJ)
1650 IF (J=78)+(J=83)+(J=69)+(J=87)=0 TH
EN 1640
1660 CALL CLEAR
1670 IF (J=78)+(A(E-10)=77)<>-2 THEN 170
0
1680 SS=1
1690 YT=E-10
1700 IF (J=83)+(A(E+10)=77)<>-2 THEN 173
0
1710 SS=1
1720 YT=E+10
1730 IF (J=69)+(A(E+1)=77)<>-2 THEN 1760
1740 SS=1
1750 YT=E+1
1760 IF (J=87)+(A(E-1)=77)<>-2 THEN 1790
1770 SS=1
1780 YT=E-1
1790 IF SS=1 THEN 1820
1800 PRINT :TAB(9); "NOTHING THERE"
1810 GOTO 1930
1820 PRINT :TAB(10); "** A HIT **"
1830 FOR T=250 TO 150 STEP -10
1840 CALL SOUND(200,T,0)
1850 CALL SOUND(100,T,0)
1860 NEXT T
1870 IF RND>.4 THEN 1920
1880 PRINT :TAB(6); "THE MONSTER IS DEAD"
1890 A(YT)=46
1900 G=G+INT(RND*100)
1910 GOTO 1930

```

```

1920 PRINT :TAB(5);"THE MONSTER IS WOUND
ED"
1930 FOR T=1 TO 600
1940 NEXT T
1950 RETURN
1960 PRINT ::
1970 PRINT TAB(4);"YOU HAVE RUN OUT OF A
IR"
1980 PRINT :TAB(2);"YOU SURVIVED FOR";H;
"MINUTES"
1990 PRINT :TAB(4);"AND FOUND $";G;"OF G
OLD"
2000 CALL SOUND(250,262,0)
2010 CALL SOUND(125,262,0)
2020 CALL SOUND(125,262,0)
2030 CALL SOUND(250,294,0)
2040 CALL SOUND(250,262,0)
2050 CALL SOUND(250,294,0)
2060 CALL SOUND(2000,220,0,262,0,330,0)
2070 FOR T=1 TO 800
2080 NEXT T
2090 STOP
2100 A(E)=128
2110 IF RND>.3 THEN 2240
2120 PRINT ::
2130 FOR J=1 TO 91 STEP 10
2140 PRINT TAB(11);
2150 FOR JJ=J TO J+9
2160 PRINT CHR$(A(JJ));
2170 NEXT JJ
2180 PRINT
2190 NEXT J
2200 WA=INT(RND*400)
2210 FOR J=1 TO WA
2220 NEXT J
2230 CALL CLEAR
2240 RETURN
2250 DATA 129,109,77,81,36
2260 DATA -11,-10,-9,-1,1,9,10,11

```

SPIN A WIN

You have probably guessed by the title that this program simulates a one armed bandit. You will be given twenty computer dollars to start with and each spin will cost you one dollar. Unfortunately computer dollars can only be spent inside a computer.

When START appears on the screen press S and you will actually be able to see the reels spinning thanks to the clever part of the program from lines 900 to 1190 and also the routine from lines 2150 to 2180. There are three reels and each one is made up of three large rectangles, each one being a different colour. When the reels stop spinning there will be three coloured rectangles beside each other displayed on the screen. If all three rectangles are the same colour then you win \$5, and if all three colours are displayed on the screen you win \$3. Any other combination of colours means that you lose your money. If you do have a win you may collect or gamble the money by pressing either C or G. Pressing G will cause the words NOTHING and DOUBLE to appear on the screen and they will flash on and off alternately until, after a while, the light stops on one or the other of them. You may gamble the winnings as many times as you wish but you will usually lose if you gamble more than twice on the same go. If you do gamble you have a 40 percent chance of winning.

The game ends when you either win more than \$50 in total and break the bank, or when you lose all of your computer dollars and are bankrupt.

The characters used in drawing the display are defined in lines 20 to 130 using the CALL CHAR command. Lines 140 to 640 set up the screen and display the title. The main part of the program is from lines 730 to 2010 and lines 2020 to 2140 end the game. Variable S is the amount of money that you have left.

```

10 RANDOMIZE
20 CALL CHAR(128,"0000000000000000FF")
30 CALL CHAR(129,"FF00000000000000")
40 CALL CHAR(130,"0101010101010101")
50 CALL CHAR(131,"B0B0B0B0B0B0B0B0")
60 CALL CHAR(132,"000000000000FFFF")
70 CALL CHAR(133,"FFFF000000000000")
80 CALL CHAR(134,"0303030303030303")
90 CALL CHAR(135,"C0C0C0C0C0C0C0C0")
100 CALL CHAR(136,"0000181818180000")
110 CALL CHAR(137,"0000003C3C000000")
120 CALL CHAR(144,"0000181818180000")
130 CALL CHAR(145,"0000003C3C000000")
140 CALL CLEAR
150 CALL SCREEN(12)
160 CALL COLOR(9,5,5)
170 CALL COLOR(10,9,9)
180 CALL COLOR(11,3,3)
190 CALL COLOR(13,2,1)
200 CALL HCHAR(2,3,132,29)
210 CALL HCHAR(11,3,133,29)
220 CALL VCHAR(3,2,134,8)
230 CALL VCHAR(3,32,135,8)
240 FOR N=3 TO 25 STEP 11
250 CALL VCHAR(5,N,130,4)
260 CALL VCHAR(5,N+6,131,4)
270 NEXT N
280 FOR N=4 TO 26 STEP 11
290 CALL HCHAR(4,N,128,5)
300 FOR P=5 TO 8
310 CALL HCHAR(P,N,104,5)
320 NEXT P
330 CALL HCHAR(9,N,129,5)
340 NEXT N
350 A=104
360 B=104
370 C=104
380 S=20
390 A$=" SPIN"
400 J=6

```

```

410 K=3
420 GOSUB 2230
430 CALL SOUND(500,262,0,330,0,392,0)
440 A$=" A "
450 K=14
460 GOSUB 2230
470 CALL SOUND(500,220,0,262,0,330,0)
480 A$=" WIN "
490 K=25
500 GOSUB 2230
510 CALL SOUND(500,175,0,220,0,262,0)
520 A$="THREE OF A KIND PAYS....$5"
530 J=22
540 K=4
550 GOSUB 2230
560 A$="ONE OF EACH PAYS.....$3"
570 J=24
580 GOSUB 2230
590 A$="HIT ANY KEY TO PLAY"

600 J=16
610 K=7
620 GOSUB 2230
630 CALL KEY(3,J,K)
640 IF K=0 THEN 630
650 CALL HCHAR(16,8,32,19)
660 CALL HCHAR(6,4,104,5)
670 CALL HCHAR(6,15,104,5)
680 CALL HCHAR(6,26,104,5)
690 A$="YOU HAVE $20"
700 J=13
710 K=11
720 GOSUB 2230
730 IF S=0 THEN 2020
740 IF S>50 THEN 2070
750 CALL COLOR(14,12,12)
760 CALL COLOR(15,12,12)
770 A$="START"
780 J=17
790 K=14
800 GOSUB 2230

```

```

810 CALL SOUND(300,-1,0)
820 CALL KEY(3,J,K)
830 IF J<>83 THEN 820
840 CALL HCHAR(17,15,32,5)
850 S=S-1
860 A$=STR$(S)&" "
870 J=13
880 K=21
890 GOSUB 2230
900 E=INT(RND*8)+8
910 K=4
920 FOR P=1 TO E
930 A=A+8
940 IF A<120 THEN 960
950 A=96
960 Q=A
970 GOSUB 2150
980 NEXT P
990 CALL SOUND(150,523,0)
1000 E=INT(RND*8)+8
1010 K=15
1020 FOR P=1 TO E
1030 B=B+8
1040 IF B<120 THEN 1060
1050 B=96
1060 Q=B
1070 GOSUB 2150
1080 NEXT P
1090 CALL SOUND(150,262,0)
1100 E=INT(RND*8)+8
1110 K=26
1120 FOR P=1 TO E
1130 C=C+8
1140 IF C<120 THEN 1160
1150 C=96
1160 Q=C
1170 GOSUB 2150
1180 NEXT P
1190 CALL SOUND(150,131,0)

```

```
1200 W=0
1210 IF (A=B)+(A=C)<>-2 THEN 1230
1220 W=5
1230 IF (A=B)+(A=C)+(B=C)<>0 THEN 1250
1240 W=3
1250 IF W=0 THEN 1650
1260 A$="YOU WIN $"&STR$(W)
1270 J=15
1280 K=12
1290 GOSUB 2230
1300 CALL HCHAR(17,10,136,1)
1310 CALL HCHAR(19,10,136,1)
1320 CALL HCHAR(17,26,144,1)
1330 CALL HCHAR(19,26,144,1)
1340 A$=CHR$(137)&"COLLECT"&CHR$(137)
1350 J=18
1360 K=5
1370 GOSUB 2230
1380 A$=CHR$(145)&"GAMBLE"&CHR$(145)
1390 K=21
1400 GOSUB 2230
1410 CALL COLOR(15,12,12)
1420 CALL COLOR(14,6,1)
1430 N=1
1440 CALL SOUND(200,185,0)
1450 CALL KEY(3,J,K)
1460 IF J=67 THEN 1580
1470 N=N+1
1480 IF N<5 THEN 1450
1490 CALL COLOR(14,12,12)
1500 CALL COLOR(15,6,1)
1510 N=1
1520 CALL SOUND(200,208,0)
1530 CALL KEY(3,J,K)
1540 IF J=71 THEN 1660
1550 N=N+1
1560 IF N<5 THEN 1530
1570 GOTO 1410
1580 S=S+W
1590 A$=STR$(S)
1600 J=13
```

```
1610 K=21
1620 GOSUB 2230
1630 CALL HCHAR(15,13,32,11)
1640 GOSUB 2190
1650 GOTO 730
1660 CALL COLOR(14,12,12)
1670 CALL COLOR(15,12,12)
1680 A$="NOTHING"
1690 J=18
1700 K=6
1710 GOSUB 2230
1720 A$="DOUBLE"
1730 K=22
1740 GOSUB 2230
1750 N=1
1760 Z=RND
1770 CALL COLOR(15,12,12)
1780 CALL COLOR(14,6,1)
1790 CALL SOUND(200,370,0)
1800 FOR P=1 TO 40
1810 NEXT P
1820 IF (N=5)+(Z<.6)=-2 THEN 1970
1830 CALL COLOR(14,12,12)
1840 CALL COLOR(15,6,1)
1850 CALL SOUND(200,415,0)
1860 FOR P=1 TO 40
1870 NEXT P
1880 IF (N=5)+(Z>=.6)=-2 THEN 1910
1890 N=N+1
1900 GOTO 1770
1910 W=W*2
1920 FOR N=1 TO 400
1930 NEXT N
1940 CALL COLOR(14,12,12)
1950 CALL COLOR(15,12,12)
1960 GOTO 1260
1970 FOR N=1 TO 400
1980 NEXT N
1990 CALL HCHAR(15,13,32,11)
2000 GOSUB 2190
```

```
2010 GOTO 730
2020 A$="YOU HAVE RUN OUT OF MONEY!"
2030 J=16
2040 K=4
2050 GOSUB 2230
2060 GOTO 2110
2070 A$="YOU HAVE BROKEN THE BANK!"
2080 J=16
2090 K=4
2100 GOSUB 2230
2110 FOR N=1 TO 500
2120 NEXT N
2130 CALL CLEAR
2140 END
2150 FOR N=5 TO 8
2160 CALL HCHAR(N,K,0,5)
2170 NEXT N
2180 RETURN
2190 CALL HCHAR(17,10,32,17)
2200 CALL HCHAR(19,10,32,17)
2210 CALL HCHAR(18,6,32,24)
2220 RETURN
2230 FOR N=1 TO LEN(A$)
2240 CALL HCHAR(J,K+N,ASC(SEG$(A$,N,1)),
1)
2250 NEXT N
2260 RETURN
```

FOXES AND CHICKENS

This is your big opportunity to be an ecologist. The game Foxes and Chickens requires you to set up a population of these animals in such a way that it becomes self-supporting. Too many foxes and the chickens won't be able to breed fast enough to feed them. Too many chickens and they will overrun the farmyard and crowd out the foxes.

The computer will give you a month by month read-out of the farmyard's population. Each time the program is run the number of chickens needed to feed each fox is randomly selected.

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.

Lines 290 to 360 input the number foxes and chickens you wish to start with. Lines 380 to 770 calculate and display the number of foxes and chickens that are left from month to month. Lines 780 to 1200 tell you which animals died out and ask you if you wish to play again. The subroutine from lines 1210 to 1260 uses CALL HCHAR to position words on the screen where it wants to rather than having them appearing at the bottom of the screen and then scrolling up.

```
10 RANDOMIZE
20 CALL CLEAR
30 A$="WELCOME TO FOXES AND"
40 J=6
50 GOSUB 1210
60 A$="CHICKENS. THE OBJECT OF THE"
70 J=8
```

```

80 GOSUB 1210
90 A$="GAME IS TO CREATE A"
100 J=10
110 GOSUB 1210
120 A$="POPULATION OF FOXES AND"
130 J=12
140 GOSUB 1210
150 A$="CHICKENS WHICH WILL SURVIVE"
160 J=14
170 GOSUB 1210
180 A$="FOR AS LONG AS POSSIBLE."
190 J=16
200 GOSUB 1210
210 A$="PRESS ANY KEY TO CONTINUE."
220 J=21
230 GOSUB 1210
240 CALL KEY(3,J,K)
250 IF K=0 THEN 240
260 CALL CLEAR
270 HI=0
280 FD=RND
290 PRINT TAB(4);"HOW MANY FOXES WILL"::
300 INPUT " YOU START WITH (1-99)? ":FP
310 IF FP<100 THEN 330
320 FP=99
330 PRINT ::TAB(3);"HOW MANY CHICKENS AR
E THERE"::
340 INPUT " IN THE FARMYARD (1-99)? ":CP
350 IF CP<100 THEN 370
360 CP=99
370 MO=0
380 CALL CLEAR
390 A$="MONTH 0"
400 J=8
410 GOSUB 1210
420 A$="FOXES          CHICKENS"
430 J=12
440 GOSUB 1210
450 A$=STR$(FP)
460 J=14
470 K=8

```

```

480 GOSUB 1220
490 A$=STR$(CP)
500 K=21
510 GOSUB 1220
520 FOR N=1 TO 600
530 NEXT N
540 FP=FP/3
550 CP=CP/3
560 MO=MO+1
570 A$=STR$(MO)
580 J=8
590 K=19
600 GOSUB 1220
610 IF FP<=CP/FD THEN 630
620 FP=CP/FD
630 FP=ABS(FP+((8*FP-FP*CP/3)*FD))
640 CP=ABS(CP+((4*CP-CP*FP)*.01))
650 A$=STR$(INT(FP))&"    "
660 J=14
670 K=8
680 GOSUB 1220
690 A$=STR$(INT(CP))&"    "
700 K=21
710 GOSUB 1220
720 IF (FP<2)+(CP<2)=-2 THEN 780
730 IF (FP<2)+(CP>2)=-2 THEN 850
740 IF (FP>2)+(CP<2)=-2 THEN 890
750 FOR N=1 TO 500
760 NEXT N
770 GOTO 560
780 A$="WE HAVE RUN OUT OF FOXES"
790 J=18
800 GOSUB 1210
810 A$="AND CHICKENS!"
820 J=20
830 GOSUB 1210
840 GOTO 920
850 A$="WE HAVE RUN OUT OF FOXES!"
860 J=18
870 GOSUB 1210

```

```

880 GOTO 920
890 A$="WE HAVE RUN OUT OF CHICKENS!"
900 J=18
910 GOSUB 1210
920 CALL SOUND(1000,262,0,330,0)
930 CALL SOUND(1000,262,0,392,0)
940 CALL SOUND(1500,262,0,330,0)
950 FOR N=1 TO 800
960 NEXT N
970 CALL CLEAR
980 A$="THE POPULATION OF FOXES AND"
990 J=8
1000 GOSUB 1210
1010 A$="CHICKENS SURVIVED FOR"
1020 J=10
1030 GOSUB 1210
1040 A$=STR$(MO)&" MONTHS."
1050 J=12
1060 GOSUB 1210
1070 IF MO<HI THEN 1090
1080 HI=MO
1090 A$="THE LONGEST SO FAR IS "&STR$(HI
)
1100 J=14
1110 GOSUB 1210
1120 A$="DO YOU WISH TO PLAY AGAIN?"
1130 J=20
1140 GOSUB 1210
1150 CALL KEY(3,J,K)
1160 IF J<>89 THEN 1190
1170 CALL CLEAR
1180 GOTO 280
1190 IF J<>78 THEN 1150
1200 STOP
1210 K=INT((33-LEN(A$))/2+.5)
1220 FOR Q=1 TO LEN(A$)
1230 CALL HCHAR(J,K+Q,ASC(SEG$(A$,Q,1)),
1)
1240 CALL SOUND(100,100*Q+2*J,0)
1250 NEXT Q
1260 RETURN

```

ALLEY SCARE

Most adventure-type games are based on logic and are cleverly planned, with many challenges for the player. Alley Scare is nothing like that. It is more of an ordeal than an adventure.

The instructions are displayed at the beginning of the game by lines 150 to 180 and you are told what to do at various stages in the game. Lines 270 to 350 display various pieces of information on the screen. Lines 400 to 600 tell you what is attacking you and with what, and lines 610 to 710 give you a choice of defensive weapons, decide whether or not you have defeated the attacker and, depending on this result call one of two routines which tell you if you have won the fight or if you have lost. Lines 760 to 1000 hold the information for the situations that may occur at any time during the game.

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

```
10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(2)
40 FOR N=1 TO 8
50 CALL COLOR(N,16,1)
60 NEXT N
70 A$="** ALLEY SCARE **"
80 FOR N=1 TO 17
90 PRINT :TAB(N+7);SEG$(A$,N,1)
100 NEXT N
110 PRINT ::
```

```

120 X=0
130 S=30
140 W=1
150 PRINT " YOU ARE IN A DARK, SLIMY":
"OPENING SOMEWHERE IN A"
160 PRINT "NETWORK OF ALLEYS. YOU":"H
AVE ONLY THIRTY GOLD PIECES":"IN YOUR PO
CKET AND YOU MUST"
170 PRINT "FIGHT YOUR WAY OUT BEFORE":"Y
OU DIE OF STARVATION.":" MANY ALLEY CR
EATURES"
180 PRINT "BLOCK THE WAY AND THEY WILL":
"STEAL YOUR GOLD IF YOU DO":"NOT GUARD I
T WELL."
190 FOR N=1 TO 10
200 PRINT
210 NEXT N
220 INPUT "PRESS ENTER":A$
230 CALL CLEAR
240 IF W>=1 THEN 260
250 W=1
260 IF W>=10 THEN 1010
270 PRINT :TAB(6);"THIS IS ALLEY NO. ";W
280 PRINT :TAB(4);"ALLEY NO. 10 IS THE E
XIT"
290 X=X+1
300 PRINT :TAB(4);"THIS IS CHALLENGE NO.
 ";X
310 IF S>=1 THEN 330
320 S=3
330 PRINT :TAB(4);"YOU HAVE";S;"GOLD PIE
CES"
340 K=2+INT(RND*4)
350 PRINT :TAB(2);"YOU ARE NOW FACING";K
;"ALLEYS"
360 PRINT
370 INPUT "WHICH ONE WILL YOU TRY? ":A
380 IF RND<.2 THEN 720
390 IF A>K THEN 370
400 K=INT(RND*4)

```

```

410 IF K<>0 THEN 430
420 E$="GIANT RAT"
430 IF K<>1 THEN 450
440 E$="WEREWOLF"
450 IF K<>2 THEN 470
460 E$="TROLL"
470 IF K<>3 THEN 490
480 E$="ZOMBIE"
490 PRINT :TAB(3);"THE ALLEY IS BLOCKED
BY A"
500 E=INT(RND*4)
510 IF E<>0 THEN 530
520 F$="POISONED SPEAR"
530 IF E<>1 THEN 550
540 F$="FLAMING SWORD"
550 IF E<>2 THEN 570
560 F$="BOW AND ARROW"
570 IF E<>3 THEN 590
580 F$="CROSSBOW"
590 PRINT :TAB(7);E$;" ARMED WITH"
600 PRINT :TAB(10);"A ";F$
610 PRINT :TAB(2);"WHICH WEAPON DO YOU C
HOOSE?"
620 PRINT :TAB(6);"A POINTED STICK (1)"
630 PRINT :TAB(9);"BARE HANDS (2)"
640 PRINT :TAB(8);"A SMALL ROCK (3)"
650 INPUT B
660 C=INT(RND*3)+1
670 IF B=C THEN 700
680 GOSUB 1250
690 GOTO 740
700 GOSUB 1180
710 GOTO 740
720 K=1+INT(RND*4)
730 ON K GOSUB 760,810,860,910
740 PRINT
750 GOTO 220
760 PRINT :TAB(4);"YOU HAVE FALLEN THROU
GH"
770 PRINT :TAB(11);"A TRAPDOOR"

```

```

780 W=W-1
790 S=S-INT(RND*2)-1
800 RETURN
810 PRINT :TAB(4);"A FLOOD WASHES YOU DO
WN"
820 PRINT :TAB(10);"A SIDE ALLEY"
830 W=W+1
840 S=S-INT(RND*2)-1
850 RETURN
860 PRINT :TAB(4);"YOU HAVE BEEN HELPED
BY"
870 PRINT :TAB(8);"A PASSING GNOME"
880 S=S+INT(RND*5)+1
890 W=W+INT(RND*3)+1
900 RETURN
910 PRINT :TAB(5);"WHOOPEE!! A HOARD OF
"
920 PRINT :TAB(2);"GOLD, CHOOSE UP TO 5
PIECES"
930 PRINT :TAB(4);"BUT BE CAREFUL, THE M
ORE"
940 PRINT :TAB(3);"YOU TAKE, THE MORE IT
WILL"
950 PRINT :TAB(2);"COST YOU. HOW MANY TO
TAKE?"
960 INPUT D
970 IF D>5 THEN 960
980 S=S+D
990 W=W-INT(D/2)
1000 RETURN
1010 PRINT :TAB(6);"YOU ARE AT THE EXIT"
1020 E=20+INT(RND*5)
1030 PRINT :TAB(3);"THERE IS A LARGE OGR
E HERE"
1040 PRINT :TAB(3);"DEMANDING";E;"GOLD P
IECES."
1050 PRINT :TAB(5);"PRESS ENTER TO PAY H
IM"
1060 INPUT A$
1070 S=S-E

```

```

1080 IF S<0 THEN 1150
1090 PRINT :TAB(6);"YES, YOU CAN PAY HIM
"
1100 PRINT :TAB(5);"YOU HAVE WON THE GAM
E"
1110 IF S=0 THEN 1170
1120 PRINT :TAB(6);"AND YOU MAY KEEP THE
"
1130 PRINT :TAB(4);"REMAINING";S;"GOLD P
IECES"
1140 GOTO 1170
1150 PRINT :TAB(4);"THE OGRE HAS KILLED
YOU"
1160 PRINT :TAB(6);"YOU COULD NOT PAY HI
M"
1170 END
1180 PRINT :TAB(4);"YOU HAVE FOUGHT YOUR
WAY"
1190 PRINT :TAB(8);"PAST THE ";E$
1200 S=S+INT(RND*3)+1
1210 PRINT :TAB(4);"AND HAVE";S;"GOLD PI
ECES"
1220 W=W+INT(RND*3)+1
1230 PRINT :TAB(2);"YOU ARE APPROACHING
ALLEY";W
1240 RETURN
1250 PRINT :TAB(2);"THE ";E$;" BEAT YOU
AND"
1260 S=S-INT(RND*4)-1
1270 PRINT :TAB(8);"LEFT YOU WITH";S
1280 PRINT :TAB(10);"GOLD PIECES"
1290 W=W+1-INT(RND*3)
1300 IF W>0 THEN 1320
1310 W=1
1320 PRINT :TAB(4);"AND SENT YOU TO ALLE
Y";W
1330 RETURN

```

SKITTLES

With this program you can try your luck at bowling. The computer will show you a view of the bowling alley from above, with ten skittles set out in the usual way at the top of the screen. You will then be able to play a game of ten frames with two balls to each frame.

The computer will tell you which frame and ball you are currently playing, your score for that frame, your total score and also the highest score achieved over ten frames so far. It will also reset the skittles for you after each frame. When the screen has been set up by lines 20 to 610 your ball will appear at the bottom end of the alley. You may then move it left or right using keys S and D, and when it is in the correct position press B to bowl. The ball will then roll up the alley and knock over any skittles in its path. When the ball hits a skittle it will be deflected in any one of three directions. A skittle which is knocked over by the ball may knock over other skittles beside or behind it making the game much more realistic.

You score one point for each skittle that is knocked down and if you successfully knock all ten skittles down in the same frame then you are given a bonus of five points.

Lines 620 to 690 allow you to line up the ball before you bowl it. After you press B the ball is moved up the screen by lines 700 to 740 and lines 750 to 1030 work out which skittles have been knocked over. Then the score is adjusted accordingly by lines 1060 to 1190 and it is displayed on the screen. After all ten frames have been played lines 1280 to 1340 end the game and amend the high score if it has been beaten before going back to the beginning of the program for a new game.

```

10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(7)
40 FOR N=1 TO 8
50 CALL COLOR(N,16,1)
60 NEXT N
70 CALL CHAR(128,"00183C3C3C3C1800")
80 CALL CHAR(136,"183C3C1818181818")
90 CALL COLOR(9,16,16)
100 CALL COLOR(10,15,15)
110 CALL COLOR(12,2,16)
120 CALL COLOR(13,2,16)
130 CALL COLOR(14,5,16)
140 A$="SKITTLES"
150 PRINT TAB(14);"/\"
160 PRINT ::::
170 PRINT TAB(11);A$
180 PRINT ::::
190 A$=A$&" "&A$
200 PRINT TAB(6);A$
210 PRINT ::::
220 A$=A$&" "&SEG$(A$,1,8)
230 PRINT A$
240 PRINT ::::
250 H=0
260 FOR N=1 TO 1000
270 NEXT N
280 CALL CLEAR
290 PRINT TAB(24);"FRAME"
300 PRINT : "SCORE FOR";TAB(25);"=1"
310 PRINT "THIS FRAME"
320 PRINT :TAB(4);"=0"
330 PRINT TAB(24);"BALLS"
340 PRINT TAB(24);"LEFT"
350 PRINT :TAB(25);"=2"
360 PRINT TAB(3);"TOTAL"
370 PRINT TAB(3);"SCORE"
380 PRINT :TAB(4);"=0";TAB(24);"HIGH"
390 PRINT TAB(24);"SCORE"
400 PRINT :TAB(24);"=";H

```

```

410 PRINT ::
420 FOR N=13 TO 21
430 CALL VCHAR(1,N,96,24)
440 NEXT N
450 F=0
460 A$=CHR$(136)&CHR$(96)
470 A$=A$&A$&A$&A$
480 FOR A=1 TO 10
490 S=0
500 B$=STR$(A)&" "
510 CALL HCHAR(7,28,ASC(SEG$(B$,1,1)),1)
520 CALL HCHAR(7,29,ASC(SEG$(B$,2,1)),1)
530 CALL SOUND(500,131,0,165,0)
540 FOR J=2 TO 8 STEP 2
550 B$=SEG$(A$,1,10-J)
560 GOSUB 1360
570 NEXT J
580 B=1
590 CALL HCHAR(10,7,48,1)
600 CALL HCHAR(10,8,32,1)
610 CALL HCHAR(14,28,51-B,1)
620 Y=17
630 CALL HCHAR(24,Y,96,1)
640 CALL KEY(3,J,K)
650 Y=Y+(J=83)-(J=68)
660 Y=Y+(Y=22)-(Y=12)
670 IF J=66 THEN 700
680 CALL HCHAR(24,Y,128,1)
690 GOTO 630
700 FOR N=23 TO 9 STEP -1
710 CALL SOUND(-1000,-5,10)
720 CALL HCHAR(N+1,Y,96,1)
730 CALL HCHAR(N,Y,128,1)
740 NEXT N
750 V=0
760 W=Y
770 FOR N=8 TO 2 STEP -2
780 Y=Y+V
790 CALL SOUND(-2000,-5,10)
800 CALL GCHAR(N,Y,Q)

```

```

810 CALL HCHAR(N+1,W,96,1)
820 CALL HCHAR(N,Y,128,1)
830 W=Y
840 IF Q<>136 THEN 970
850 CALL SOUND(-100,262,0,-5,10)
860 V=INT(RND*3)-1
870 IF (RND<.4)+(Y>14)<>-2 THEN 890
880 CALL HCHAR(N,Y-2,96,1)
890 IF (RND<.4)+(Y<20)<>-2 THEN 910
900 CALL HCHAR(N,Y+2,96,1)
910 IF N=2 THEN 960
920 IF RND<.6 THEN 940
930 CALL HCHAR(N-2,Y-1,96,1)
940 IF RND<.6 THEN 990
950 CALL HCHAR(N-2,Y+1,96,1)
960 GOTO 990
970 FOR P=1 TO 40
980 NEXT P
990 CALL HCHAR(N,Y,96,1)
1000 CALL HCHAR(N-1,Y,128,1)
1010 FOR P=1 TO 40
1020 NEXT P
1030 NEXT N
1040 CALL HCHAR(1,Y,96,1)
1050 CALL SOUND(-1,-7,0)
1060 Z=10
1070 FOR N=1 TO 4
1080 FOR P=13+N TO 21-N
1090 CALL GCHAR(N*2,P,Q)
1100 IF Q<>136 THEN 1120
1110 Z=Z-1
1120 NEXT P
1130 NEXT N
1140 S=Z
1150 IF Z<>10 THEN 1170
1160 S=S+5
1170 B$=STR$(S)&" "
1180 CALL HCHAR(10,7,ASC(SEG$(B$,1,1)),1)
)
1190 CALL HCHAR(10,8,ASC(SEG$(B$,2,1)),1)
)

```

```

1200 B=B+1
1210 IF B=2 THEN 610
1220 F=F+S
1230 B#=STR$(F)&"  "
1240 CALL HCHAR(18,7,ASC(SEG$(B#,1,1)),1
)
1250 CALL HCHAR(18,8,ASC(SEG$(B#,2,1)),1
)
1260 CALL HCHAR(18,9,ASC(SEG$(B#,3,1)),1
)
1270 NEXT A
1280 CALL HCHAR(14,28,48,1)
1290 B#="END OF GAME"
1300 J=11
1310 GOSUB 1360
1320 IF F<H THEN 260
1330 H=F
1340 GOTO 260
1350 CALL HCHAR(14,28,48,1)
1360 K=INT((32-LEN(B#))/2)+1
1370 FOR N=1 TO LEN(B#)
1380 CALL HCHAR(J,K+N,ASC(SEG$(B#,N,1)),
1)
1390 NEXT N
1400 RETURN

```

NUMBER JUGGLE

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 0123456789 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 NUMBER?". Entering 1 would then reverse the whole 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.

Lines 70 to 160 draw a sine wave at the beginning of the game. Lines 200 to 340 generate the ten digits and juggle them around. Lines 360 to 530 allow you to continue reversing parts of the sequence until you get it in order. When you have done this lines 540 to 630 tell you how many moves it took and play a short tune.

```
10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(12)
40 PRINT TAB(9); "NUMBER JUGGLE"
50 PRINT :::
60 Z=48
70 FOR N=0 TO 6.2 STEP 0.2
80 Y=N*5+1
90 X=9+SIN(N)*8
100 S=110+RND*2000
110 CALL HCHAR(X,Y,Z,1)
120 CALL SOUND(100,S,0)
130 Z=Z+1
140 IF Z<58 THEN 160
```

```

150 Z=48
160 NEXT N
170 FOR N=1 TO 800
180 NEXT N
190 M=1
200 FOR N=1 TO 10
210 A(N)=N+47
220 NEXT N
230 FOR T=1 TO 20
240 X=1+INT(RND*10)
250 Y=1+INT(RND*10)
260 B=A(X)
270 C=A(Y)
280 A(X)=C
290 A(Y)=B
300 NEXT T
310 A$=""
320 FOR N=1 TO 10
330 A$=A$&CHR$(A(N))
340 NEXT N
350 CALL CLEAR
360 CALL SOUND(200,262,0)
370 CALL SOUND(200,523,0)
380 CALL SOUND(200,262,0)
390 PRINT : :TAB(3);"MOVE NO. ";M;": ";A$
400 PRINT : :TAB(8);"REVERSE NUMBER?"
410 INPUT R
420 IF (R<1)+(R>9)<>0 THEN 410
430 B$=""
440 FOR T=10 TO R STEP -1
450 B$=B$&SEG$(A$,T,1)
460 NEXT T
470 IF R=1 THEN 500
480 A$=SEG$(A$,1,R-1)&B$
490 GOTO 510
500 A$=B$
510 IF A$="0123456789" THEN 540
520 M=M+1
530 GOTO 360

```

```
540 PRINT :TAB(18);A#
550 PRINT :::TAB(10);"YOU DID IT."
560 PRINT :TAB(8);"IT TOOK";M;"MOVES."
570 PRINT :::
580 RESTORE
590 FOR N=1 TO 7
600 READ S
610 CALL SOUND(100,S,0)
620 NEXT N
630 DATA 131,165,196,247,196,165,131
```

TWENTY ONE

This program plays a dice version of the card game Pontoon. The object of the game is to roll the dice until the total is either 21, or as close to it as you dare 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 that the computer plays this game quite well and will win more than fifty percent of the games 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 times 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.

Lines 110 to 190 draw the circle of numbers at the beginning of the game using sine (SIN) and cosine (COS). The loop has to go from 0 to 2π as the computer works in radians rather than degrees. There are 2π radians in one complete revolution. π is defined in line 110 using the DEF command. The player's go is controlled by lines 270 to 420 and the computer's go is controlled by lines 430 to 540. Lines 550 to 640 work out who has won and call either the routine from lines 980 to 1010 or the routine from lines 1020 to 1050. If it is a draw then the routine from lines 920 to 970 is called. The dice is rolled by the routine from lines 650 to 710 which also provides the sound effects.

```

10 RANDOMIZE
20 CALL CLEAR
30 CALL SCREEN(8)
40 CALL COLOR(3,7,1)
50 CALL COLOR(4,7,1)
60 PRINT TAB(13);"TWENTY"
70 PRINT :TAB(14);"ONE"
80 FOR N=1 TO 10
90 PRINT
100 NEXT N
110 DEF PI=3.1416
120 A=50
130 FOR N=0 TO 2*PI STEP .4
140 X=10*SIN(N)
150 Y=10*COS(N)
160 CALL HCHAR(X+12,Y+17,A,1)
170 A=-49*(A=50)-50*(A=49)
180 CALL SOUND(100,A*4,0,A-51,0)
190 NEXT N
200 HH=0
210 CC=0
220 FOR N=1 TO 1000
230 NEXT N
240 CALL CLEAR
250 H=0
260 C=0
270 PRINT :TAB(5);"1 TO ROLL, 2 TO STAND
"
280 PRINT :TAB(12);"3 TO END"
290 CALL KEY(3,J,K)
300 IF (J=49)+(J=50)+(J=51)=0 THEN 290
310 IF J<>51 THEN 330
320 STOP
330 IF J=50 THEN 430
340 GOSUB 650
350 H=H+D
360 CALL SOUND(300,156,0,622,0)
370 PRINT :TAB(8);"YOUR SCORE IS";H
380 IF H<21 THEN 290
390 IF H=21 THEN 420

```

```

400 GOSUB 720
410 GOTO 430
420 GOSUB 810
430 PRINT : : TAB(11); "MY GO NOW"
440 GOSUB 650
450 C=C+D
460 CALL SOUND(300,233,0,932,0)
470 PRINT : TAB(9); "MY SCORE IS"; C
480 IF (C<21)+(H>C)=-2 THEN 440
490 IF (C<21)+(H=C)+(C<19)=-3 THEN 440
500 IF C<21 THEN 550
510 IF C=21 THEN 540
520 GOSUB 720
530 GOTO 550
540 GOSUB 810
550 IF H>C THEN 610
560 IF H<C THEN 590
570 GOSUB 920
580 GOTO 620
590 GOSUB 980

600 GOTO 620
610 GOSUB 1020
620 PRINT : : TAB(9); HH; "GAMES TO YOU"
630 PRINT : : TAB(9); CC; "GAMES TO ME"
640 GOTO 220
650 D=1+INT(RND*6)
660 FOR N=1 TO 20
670 CALL SOUND(10,-1,0)
680 CALL SOUND(10,-7,0)
690 NEXT N
700 PRINT : TAB(7); "THE DICE SHOWS A"; D
710 RETURN
720 PRINT : : TAB(12); "B U S T"
730 FOR N=1 TO 20
740 CALL SOUND(100,110,0,-7,0)
750 NEXT N
760 IF H>21 THEN 790
770 C=0
780 RETURN
790 H=0

```

```
800 RETURN
810 PRINT :TAB(6);"T W E N T Y   O N E"
820 FOR N=3 TO 16
830 CALL COLOR(1,1,N)
840 CALL SCREEN(N)
850 FOR F=1 TO 5
860 CALL SOUND(5,N*60,0)
870 NEXT F
880 NEXT N
890 CALL SCREEN(8)
900 CALL COLOR(1,1,8)
910 RETURN
920 PRINT :TAB(11);"DEAD HEAT"
930 CALL SOUND(500,-3,0)
940 CALL SOUND(10,-1,0)
950 HH=HH+1
960 CC=CC+1
970 RETURN
980 PRINT :TAB(12);"I   W   I   N"
990 CALL SOUND(300,131,0)
1000 CC=CC+1
1010 RETURN
1020 PRINT :TAB(9);"Y O U   W I N"
1030 CALL SOUND(300,523,0)
1040 HH=HH+1
1050 RETURN
```

ORCHARD NIM

This program will print a random number of apples on the screen. You and the computer then take turns removing apples from the board. The one who is forced to take the last apple is the loser.

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

The computer has been programmed not to play perfectly. Line 580 puts a random factor in the computer's moves if the condition is met in line 570. Lines 320 to 380 print the apples on the screen. Variable Z is the number of apples that are left. This is initially a random number between 21 and 36. Z is modified by your input (C) in line 400 and by the computer's move (D) in line 620. Lines 530 to 610 are the lines the computer uses to make its moves.

```
10 RANDOMIZE
20 CALL CLEAR
30 CALL CHAR(128,"@C1054FEFEFE7C38")
40 CALL CHAR(136,"FFFFFFFFFFFFFFFF")
50 CALL SCREEN(16)
60 CALL COLOR(13,3,1)
70 CALL COLOR(14,11,1)
80 PRINT TAB(10);"ORCHARD NIM"
90 FOR N=1 TO 12
100 PRINT
110 NEXT N
```

```

120 FOR N=1 TO 40
130 CALL HCHAR(1+RND*23,1+RND*31,128,1)
140 CALL SOUND(100,110+RND*1000,0)
150 NEXT N
160 FOR N=1 TO 24
170 CALL HCHAR(N,1,32,32)
180 CALL HCHAR(N,3,136,1)
190 CALL SOUND(200,110+N*10,0)
200 CALL HCHAR(N,3,32,1)
210 NEXT N
220 C=0
230 D=0
240 Z=21+INT(RND*16)
250 F=3+INT(RND*3)
260 DIM A(36)
270 GOTO 730
280 PRINT TAB(6);"PIECES ON BOARD:";Z
290 PRINT :TAB(2);"MAXIMUM NUMBER TO REM
OVE:";F
300 PRINT :TAB(3);"YOU TOOK";C;TAB(20);"
I TOOK";D
310 PRINT ::
320 FOR N=1 TO 31 STEP 6
330 PRINT :TAB(10);
340 FOR P=0 TO 5
350 PRINT CHR$(A(N+P));" ";
360 NEXT P
370 PRINT
380 NEXT N
390 PRINT ::TAB(4);"HOW MANY WILL YOU TA
KE?"
400 INPUT C
410 IF (C<1)+(C>F)<>0 THEN 400
420 Z=Z-C
430 IF Z>0 THEN 530
440 CALL CLEAR
450 CALL SOUND(1000,262,0,330,0,392,0)
460 PRINT :TAB(6);"# # # I WIN # # #"
470 FOR N=1 TO 8
480 PRINT

```

```

490 NEXT N
500 FOR N=1 TO 800
510 NEXT N
520 GOTO 810
530 D=Z-1-INT((Z-1)/(F+1))*(F+1)
540 IF (D=0)+(Z<>1)<>-2 THEN 560
550 D=1+INT(RND*F)
560 IF D>=Z THEN 550
570 IF RND<.8 THEN 590
580 D=D+INT(RND*2)-INT(RND*2)
590 IF D>F THEN 580
600 IF D>0 THEN 620
610 D=1
620 Z=Z-D
630 IF Z>0 THEN 730
640 CALL CLEAR
650 CALL SOUND(1000,392,0,494,0,587,0)
660 PRINT TAB(5);"* * * YOU WIN * * *"
670 FOR N=1 TO 8
680 PRINT
690 NEXT N
700 FOR N=1 TO 800
710 NEXT N
720 GOTO 810
730 FOR N=1 TO Z
740 A(N)=128
750 NEXT N
760 FOR N=Z+1 TO 36
770 A(N)=32
780 NEXT N
790 CALL CLEAR
800 GOTO 280
810 PRINT TAB(6);"ANOTHER GAME? (Y/N)"
820 PRINT ::
830 INPUT A$
840 IF A$="Y" THEN 220
850 IF A$<>"N" THEN 830
860 END

```

SPACE RACE

Five martians are stranded in space and they must all be rescued as quickly as possible. The only problem is that a vast asteroid belt separates them from your small ship and your fuel tanks can only hold enough fuel for you to rescue one martian before returning to your mothership to refuel. The mothership however is so large that it can supply you with an almost unlimited amount of fuel which means that you can make as many trips through the asteroid belt as you wish.

You start off with three ships but you destroy one each time you hit an asteroid. Your ship starts at the bottom of the screen and moves upwards towards the martians automatically. Press the S key to move left and the D key to move right. You may also put the ship into reverse by pressing the space bar but this wastes valuable time. If you manage to reach the top of the screen without being destroyed you are given a few seconds to pick up a man before the ship turns around and starts to move back down the screen. If you hit an asteroid on the return journey and a martian is onboard he will be killed. If you save all five martians or all of your ships are destroyed the game ends and you are asked if you wish to play again.

This is quite a fast moving game as the number of objects that move on the screen at the same time have been kept to a minimum. If I had wanted the asteroids to move as well as the ship I would have had to write the game in a low level language such as machine code which I am afraid it is not possible to do using the standard machine.

```
10 RANDOMIZE
20 CALL CHAR(128,"00183C7E7E3C3000")
30 CALL CHAR(136,"08081C3E3E7F4949")
40 CALL CHAR(137,"49497F3E3E1C0808")
50 CALL CHAR(144,"386C3810FE102844")
60 CALL CLEAR
70 CALL SCREEN(2)
80 CALL COLOR(1,2,2)
```

```

90 FOR N=3 TO 8
100 CALL COLOR(N,16,1)
110 NEXT N
120 CALL COLOR(13,2,1)
130 CALL COLOR(14,8,1)
140 CALL COLOR(15,13,1)
150 PRINT TAB(4);"SKILL LEVEL ? (1 TO 9)
"
160 FOR N=1 TO 10
170 PRINT
180 NEXT N
190 CALL KEY(3,J,K)
200 IF (J<49)+(J>57)<>0 THEN 190
210 NN=(J-48)/15
220 CALL CLEAR
230 TI=0
240 SP=3
250 MN=0
260 PRINT TAB(11);"SPACE RACE"
270 PRINT ::::
280 FOR N=1 TO 384
290 IF RND>NN THEN 320
300 PRINT CHR$(128);
310 GOTO 330
320 PRINT " ";
330 NEXT N
340 CALL COLOR(13,16,1)
350 PRINT :::::
360 CALL HCHAR(1,13,32,10)
370 CALL HCHAR(1,3,144,1)
380 CALL HCHAR(1,9,144,1)
390 CALL HCHAR(1,17,144,1)
400 CALL HCHAR(1,24,144,1)
410 CALL HCHAR(1,30,144,1)
420 C=0
430 AB=0
440 X=24
450 Y=3+INT(RND*28)
460 V=X
470 W=Y
480 CALL SOUND(200,392,0)

```

```

490 CALL KEY(3,J,K)
500 Y=Y+(J=83)-(J=68)
510 Y=Y+(Y=31)-(Y=2)
520 CALL SOUND(-500,-5,15)
530 IF J<>32 THEN 580
540 CALL SOUND(-400,-5,0)
550 X=X+2
560 X=X+(X>25)
570 TI=TI+1
580 TI=TI+1
590 X=X-1
600 IF X=1 THEN 1150
610 CALL GCHAR(X,Y,Z)
620 IF Z<>128 THEN 640
630 C=1
640 CALL HCHAR(V,W,32,1)
650 CALL HCHAR(X,Y,136,1)
660 IF C=1 THEN 700
670 V=X
680 W=Y
690 GOTO 490
700 FOR N=1 TO 20
710 CALL HCHAR(X,Y,137,1)
720 CALL HCHAR(X,Y,136,1)
730 CALL SOUND(-200,-7,N*1.5)
740 NEXT N
750 CALL HCHAR(X,Y,128,1)
760 SP=SP-1
770 IF AB=0 THEN 790
780 MN=MN-1
790 IF SP=0 THEN 940
800 A$="SHIPS LEFT "&STR$(SP)
810 Y=24
820 Z=10
830 GOSUB 1840
840 FOR N=1 TO 6
850 CALL COLOR(3,16,2)
860 FOR NN=1 TO 50
870 NEXT NN
880 CALL COLOR(3,2,16)
890 FOR NN=1 TO 50

```

```

900 NEXT NN
910 NEXT N
920 CALL HCHAR(24,9,32,15)
930 GOTO 1700
940 CALL CLEAR
950 CALL COLOR(3,16,2)
960 A$="ALL SHIPS DESTROYED"
970 Y=8
980 Z=7
990 GOSUB 1840
1000 A$="YOU SAVED "&STR$(MN)&" MEN"
1010 Y=10
1020 Z=9
1030 GOSUB 1840
1040 H=INT(TI/60)
1050 M=TI-H*60
1060 A$="IN "&STR$(H)&" HOURS AND"
1070 Y=12
1080 Z=9
1090 GOSUB 1840
1100 A$=STR$(M)&" MINUTES"
1110 Y=14
1120 Z=11
1130 GOSUB 1840
1140 GOTO 1880
1150 FOR N=1 TO 3
1160 CALL GCHAR(X,Y,Z)
1170 CALL HCHAR(V,W,32,1)
1180 CALL HCHAR(X,Y,136,1)
1190 IF Z<>144 THEN 1250
1200 CALL SOUND(50,262,0)
1210 CALL SOUND(100,330,0)
1220 CALL SOUND(200,392,0)
1230 MN=MN+1
1240 AB=1
1250 V=X
1260 W=Y
1270 CALL KEY(3,J,K)
1280 Y=Y+(J=83)-(J=68)
1290 Y=Y+(Y=31)-(Y=2)

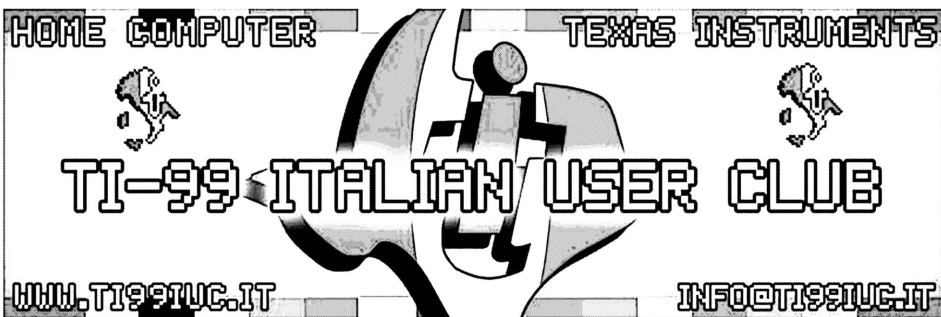
```

```

1300 NEXT N
1310 X=2
1320 CALL HCHAR(V,W,32,1)
1330 CALL HCHAR(X,Y,137,1)
1340 V=X
1350 W=Y
1360 CALL KEY(3,J,K)
1370 Y=Y+(J=83)-(J=68)
1380 Y=Y+(Y=31)-(Y=2)
1390 CALL SOUND(-500,-5,15)
1400 IF J<>32 THEN 1450
1410 CALL SOUND(-400,-5,0)
1420 X=X-2
1430 X=X-(X<1)
1440 TI=TI+1
1450 TI=TI+1
1460 X=X+1
1470 IF X=24 THEN 1570
1480 CALL GCHAR(X,Y,Z)
1490 IF Z<>128 THEN 1510
1500 C=1
1510 CALL HCHAR(V,W,32,1)
1520 CALL HCHAR(X,Y,137,1)
1530 IF C=1 THEN 700
1540 V=X
1550 W=Y
1560 GOTO 1360
1570 IF AB=0 THEN 1660
1580 CALL SOUND(100,262,0)
1590 CALL SOUND(100,330,0)
1600 CALL SOUND(100,392,0)
1610 CALL SOUND(100,330,0)
1620 CALL SOUND(100,349,0)
1630 CALL SOUND(100,294,0)
1640 CALL SOUND(300,262,0)
1650 GOTO 1690
1660 FOR N=1110 TO 110 STEP -100
1670 CALL SOUND(5,N,0)
1680 NEXT N
1690 CALL HCHAR(V,W,32,1)

```

```
1700 NN=0
1710 FOR N=1 TO 32
1720 CALL GCHAR(1,N,Z)
1730 IF Z<>144 THEN 1750
1740 NN=1
1750 NEXT N
1760 IF NN=1 THEN 420
1770 CALL CLEAR
1780 CALL COLOR(3,16,2)
1790 A$="YOU DESTROYED "&STR$(3-SF)&" SH
IPS"
1800 Y=8
1810 Z=7
1820 GOSUB 1840
1830 GOTO 1000
1840 FOR N=1 TO LEN(A$)
1850 CALL HCHAR(Y,Z+N,ASC(SEG$(A$,N,1)),
1)
1860 NEXT N
1870 RETURN
1880 A$="ANOTHER GO ?"
1890 Y=20
1900 Z=12
1910 GOSUB 1840
1920 CALL KEY(3,J,K)
1930 IF J=89 THEN 60
1940 IF J<>78 THEN 1920
1950 END
```

Editing and digital rework in 2022 by:

TI99 Italian User Club

(info@ti99iuc.it) Downloaded from www.ti99iuc.it

In this book Scott Vincent has brought together an exciting collection of dynamic games for your TI99/4A. With 29 programs just waiting to be typed in and run, this book will give you hours of fun in the weeks and months ahead. All program listings are direct from the printer and guaranteed to run.

Games in this book include:

- NINE HOLE GOLF
 - ROULETTE
 - DIGET BRAIN
 - SPACE SURVIVAL
 - GOLD MINER
 - CUBIK'S RUBE
 - TREASURE TRAIL
- and many, many more !

**Another Great Book from
Interface Publications**