

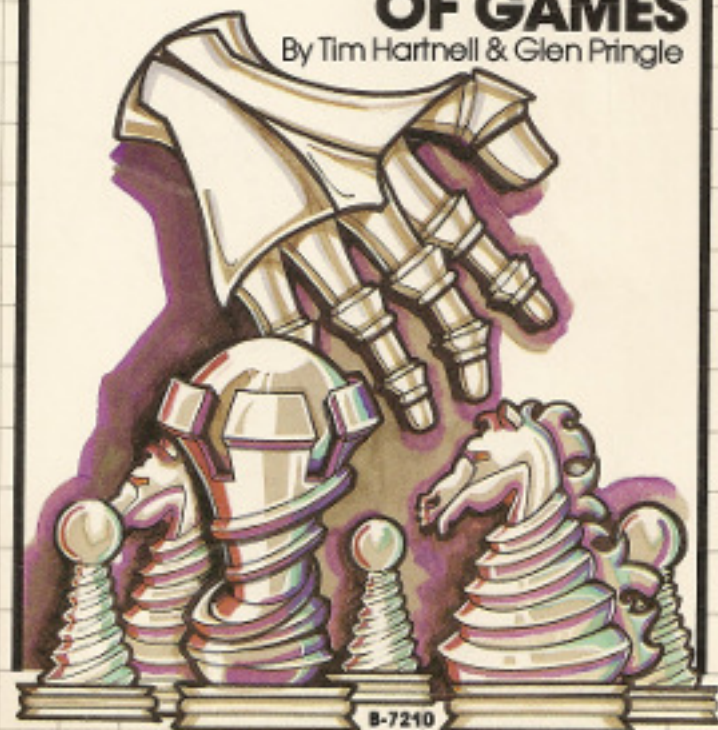


**DICK SMITH**  
**VZ200**

**Personal  
Colour Computer**

**THE GIANT BOOK  
OF GAMES**

By Tim Hartnell & Glen Pringle



First published in Australia by:

Dick Smith Management Pty. Ltd.,  
PO Box 321,  
North Ryde, NSW, 2113

Copyright (C) Hartnell and Pringle, 1983

First printing: September, 1983

ISBN 0 949772 20 8

Dick Smith Catalogue No. B 7210

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

ALL RIGHTS RESERVED

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

Reproduction in any form or for any purpose is forbidden.

Typesetting and artwork by Interface Publications,  
Level 25, Nauru House, 80 Collins Street, Melbourne,  
3000, Victoria, Australia

Cover artwork by Geoff Cullen

# INDEX

Introduction.....1

## BOARD GAMES

Introduction.....7  
GOMOKU.....13  
FOUR IN A ROW.....18  
CHECKERS.....24  
KNIGHTSBRIDGE.....30

## ADVENTURE

Introduction.....37  
STRONGHOLD OF THE DWARVEN LORDS..44  
THE DUKE OF DRAGONFEAR.....50

## SIMULATIONS

Introduction.....61  
MISTRESS OF XENOPHOBIA.....63  
RURAL PURSUITS.....66  
CHAIRMAN OF THE BOARD.....70

## DICE GAMES

Introduction/dice routine.....78  
CHEMIN DE COMPUTER.....81  
SEVEN/ELEVEN.....85  
NO SWEAT.....87  
ONE-AND-TWENTY.....90  
SNAKES EYES.....93  
UNDER'N'OVER.....97

## ARTIFICIAL INTELLIGENCE

Introduction.....103  
ELECTRONIC BRAIN.....105  
AMANUENSIS.....111

## JUST FOR FUN

Introduction/PIANO.....117  
FASTER.....119  
INNER SPRING.....120  
ROBOT MINEFIELD.....122

## Fun with your printer

CELESTIA.....126  
BILLBOARD.....133

## SPACE GAMES

MOONLANDER.....139  
HYPERWAR.....141

## BRAIN GAMES

Introduction.....149  
IDAHO STARS.....150  
EXECUTIONER.....153  
CUBIC.....156  
MUMBLE MARBLE.....159  
CYCLOTRON.....162  
FLIPPER.....165

CREATING YOUR OWN GAMES.....167  
BOOKS TO READ.....177

THE GIANT BOOK  
OF  
VZ200 GAMES

Tim Hartnell  
and  
Glen Pringle

# INTRODUCTION



Et

# INTRODUCTION

Once upon a time it was not easy to write and play games with computers. Back in the dim, dark past, it could cost you \$80 to \$100 an hour just to rent access to a time-shared mainframe via a terminal. And the only output you had was on the printout of a noisy, hard to read teleprinter. Think what those restrictions would have done to your current pattern of spending hours lovingly debugging, and improving, your latest masterpiece.

We, fortunately, do not have to suffer such medieval conditions. The luxury of having a computer of your own, driving a TV screen, with unrestricted access to the machine, allows you to work for as long as you like creating games and other programs.

In this book, you'll find a collection of our favorite computer games. We hope some of them will soon be up and running on your computer, and are stored with your own favorite programs. They're sure to provide you with a lot of fun in the weeks ahead.

[By the way, from now on, we'll use "I" in this book, rather than "we" which sounds too much like the "royal we". "I" seems to read better.]

I've deliberately written the book so that it is 'open-ended'. That is, in contrast to many other collections of games programs on the market, I've assumed you'll want to adapt and improve the programs once you get them up and running. Therefore, with many of them I've included program breakdowns, some line by line, so you know what each section of code is doing. As well, the introduction to each section discusses a bit about the philosophy of writing games of that type, has a word or two to say about the basic algorithms which you can use to create games of that sort yourself, and generally aims to give you a solid background against which

you can expand and develop your own games-writing skills.

Near the back of the book you'll even find a section devoted entirely to games ideas which you can convert into computer games.

To maximise the length of program you can get into your machine (and to keep program entry time down to a minimum) few of the programs include instructions. These are in the text. However, if you have the memory (and the inclination) you can easily add a "Do you want instructions" line as part of the initialisation subroutine, and then include a shortened form of the instructions from the text in your program.

I think we should get to the fun parts immediately.

Good game-playing,

**Tim Hartnell**  
**Glen Pringle**

# Board Games



## Board Games

Some games could not have existed without computers.

Think of the whole range of Space Intruder types of games and the Pacperson ones. These could have not come into being before the computer - using a TV-like screen for output - was developed. Many other games, of course, predate the computer by thousands of years. However, more and more of these traditional games have now been computerised, with board games being as popular with computer games players as they were with players in precomputer days.

In this section of the book, we have a generous crop of board games:

- \* GOMOKU
- \* FOUR IN A ROW
- \* CHECKERS
- \* KNIGHTSBRIDGE

We'll start this section of the book with a discussion on how board games can be easily modified to make it possible for a computer to handle them. There is a common thread which holds together all the programs in this section of the book, a thread which you can also use to help you develop programs to play your favorite board games.

Look first at this diagram. It shows a checkerboard numbered to make it easy for the computer to handle.

	1	2	3	4	5	6	7	8
8	81	82	83	84	85	86	87	88
7	71	72	73	74	75	76	77	78
6	61	62	63	64	65	66	67	68
5	51	52	53	54	55	56	57	58
4	41	42	43	44	45	46	47	48
3	31	32	33	34	35	36	37	
2	21	22	23	24	25	26	27	28
1	11	12	13	14	15	16	17	18

You can indicate any square on the board by referring to the number along the left hand side,

(such as 3), then the number along the top (such as 4). In this case, the lines numbered 3 (along the left hand side) and the line numbered 4 (along the top) meet at the square numbered 34. If you wish to move a piece, you can do so by entering the number of the square you're moving from (such as 55), then the number you're moving to (such as 66) and the computer can understand exactly what you're doing. There is no need to change the numbers entered by the human player into another set in order to allow the computer to interpret them.

That is the first 'secret' of writing board games so your computer can play them. The second is that a board numbered in this way has another great advantage over one which is simply numbered from one to 64 in order. When you move a piece in any direction on this board -- no matter where you start -- the difference between the squares is the same.

I'll try to expand on that somewhat cryptic statement. If you move one square up and to the right -- like the move of a piece in checkers -- you will move from, say, 24 to 35; or from 53 to 64; or from 71 to 82. But notice that no matter where you are on the board, the difference between your starting and ending squares is always 11. If you move diagonally up to the left, you'll move from, say, 26 to 35 (plus 9), or 66 to 75 (also plus 9) or from 22 to 31 (plus 9 again).

This predictability makes it relatively simple to create a board which the computer can handle.

## How the computer plays

Imagine the computer has a checkers piece on the square numbered 24. It could be programmed to check each square on the board, and every time it found one of its own pieces, could check if there was a human piece on the square numbered the same as its

piece (that is, 24 in this example), plus 11 (that is, on 35); and it could check to see whether the square which was 11 past that, 46, was blank. If it found that all these conditions were true, the computer could jump over square 35 into square 46, and capture the piece currently occupying 35.

This, in essence, is how many computer board games -- from checkers, through reversi, to chess -- work, based on a simple 8 by 8 grid numbered in this way.

If you were writing chess you could specify the moves of the knight by knowing it can always move to squares which are the following 'numerical distance' from the square on which the piece now sits:

21 12 -8 -19 -21 -12 19 8

Try it now, by placing a coin on square number 55, and move it as if it were a knight, working out the mathematical relationship between the starting square, and the square you're moving to. You should find the differences are the same as the numbers listed above.

The programs in this section, and in many other parts of this book, were written in a 'top down' manner, in an attempt to produce a structured program which is relatively easy to follow and modify. Many of the programs were written by entering the following 'bare code' at the start:

```
10 REM NAME OF GAME
20 GOSUB 9000:REM INITIALISE
30 GOSUB 8000:REM PRINT BOARD
40 GOSUB 1000:REM COMPUTER MOVES
50 GOSUB 8000:REM PRINT BOARD
60 GOSUB 7000:REM HUMAN MOVES
70 Check if human has won or computer has
   won and if so branch to end of
```

```
game message...
80 GOTO 30
```

This structure was written, in nearly all cases, before I had the slightest idea how I was to accomplish the goals I had set myself for each of the subroutines.

All programs were written out completely on paper before the computer was even turned on, so that a version of the game could be 'hand run' before starting to actually grapple with the game on the computer. This enable the worst bugs to be caught right at the beginning.

Among the many great advantages I discovered regarding programming on paper in this way was the willingness with which I completely discarded whole blocks of code if they were found to be unworkable. It is much more difficult to decide to erase a whole section of code from a program once you actually have it in the computer than it is to just tear up a piece of paper. The temptation -- when lines of program are actually in the computer -- is to fiddle with them, in an attempt to make them work, at least after a fashion.

Working on paper, then, tends to avoid having code which really should not be in a program somehow being patched together to make it work. Working with a structured outline, such as that I've described, makes it simple to know which parts of the program carry out which task. I know, for example, in the outline above that the board is printed by the subroutine starting at line 8000. Therefore, if the board comes out looking a bit odd, I know immediately which section of the code I should concentrate on.

As well as helping in the early 'get it working' stage, a structured program tends to invite improvements. Once you have, for example, your own Checkers



program up and running, you can go back to it some time in the future to try and make it play a little better by going just to that part of the code which covers the computer's game. You will not have to wade through vast acres of code, trying to work out just what each line does, and which particular lines control the computer's play.

The methods I've just outlined are used throughout this, and most other sections, of the book. It may prove instructive to look through some of the programs to follow through the listing, and work out which section does what. In many programs, you'll see that I've included lines of asterisks as a REM statement. These break the program into separate modules which should help you follow the program through.

# GOMOKU

You'll find GOMOKU an easy game to learn, but one which is almost impossible to win. The computer plays extremely well in this program, which is based on one written by Graham Charlton.

You have to try to get five of your pieces (the H's) in a row in any direction, while the computer is trying to do the same.

Have a look at these board positions from the start of one game I played against the program. Studying the printouts will show you how the game unfolds, and will show you how to play it:

	1	2	3	4	5	6	7	8	
1	.	.	.	.	.	.	.	.	1
2	.	.	.	.	.	.	.	.	2
3	.	.	.	C	.	.	.	.	3
4	.	.	.	.	.	.	.	.	4
5	.	.	.	.	H	.	.	.	5
6	.	.	.	.	.	.	.	.	6
7	.	.	H	.	.	.	.	.	7
8	.	.	.	.	.	C	.	.	8
	1	2	3	4	5	6	7	8	

	1	2	3	4	5	6	7	8	
1	.	.	.	.	.	.	.	.	1
2	.	.	.	.	.	.	.	.	2
3	.	.	.	C	.	.	.	.	3
4	.	.	.	.	.	.	.	.	4
5	.	.	.	.	H	.	.	.	5
6	.	.	.	C	.	.	.	.	6
7	.	.	H	.	.	.	.	.	7
8	.	.	.	.	.	C	.	.	8
	1	2	3	4	5	6	7	8	

	1	2	3	4	5	6	7	8	
1	.	.	.	.	.	.	.	.	1
2	.	.	.	.	.	.	.	.	2
3	.	.	.	C	.	.	.	.	3
4	.	.	.	.	.	.	.	.	4
5	.	.	.	.	H	.	.	.	5
6	.	.	.	C	C	.	.	.	6
7	.	.	H	.	H	.	.	.	7
8	.	.	.	.	.	C	.	.	8
	1	2	3	4	5	6	7	8	

	1	2	3	4	5	6	7	8	
1	.	.	.	.	.	.	.	.	1
2	.	.	.	.	.	.	.	.	2
3	.	.	.	C	.	.	.	.	3
4	.	.	.	.	.	.	.	.	4
5	.	.	.	.	H	C	H	.	5
6	.	.	.	C	C	.	.	.	6
7	.	.	H	.	H	.	.	.	7
8	.	.	.	.	.	C	.	.	8
	1	2	3	4	5	6	7	8	

```

1 1 2 3 4 5 6 7 8 1
2 . . . . . . . . 2
3 . . . C . . . . 3
4 . . . . . . . C 4
5 . . . . H C H . 5
6 . . . C C H . . 6
7 . . H . H . H . 7
8 . . . . . C . . 8
1 2 3 4 5 6 7 8

```

```

1 1 2 3 4 5 6 7 8 1
2 . . . . . . . . 2
3 . . . C . . . . 3
4 . . . C . . . C 4
5 . . . . H C H . 5
6 . . . C C H . . 6
7 . . H . H . H . 7
8 . . . . . C . . 8
1 2 3 4 5 6 7 8

```

```

1 1 2 3 4 5 6 7 8 1
2 . . . . . . . . 2
3 . . . C . . . . 3
4 . . . C . . . C 4
5 . . . . H C H . 5
6 . . . C C H . . 6
7 . . H C H H H . 7
8 . . . . . C . . 8
1 2 3 4 5 6 7 8

```

```

1 1 2 3 4 5 6 7 8 1
2 . . . . . . . . 2
3 . . . C . . . . 3
4 . . . C . . . C C 4
5 . . . H H C H . 5
6 . . . C C H H . 6
7 . . H C H H H C 7
8 . . . . . C . . 8
1 2 3 4 5 6 7 8

```

Here's the listing for your very own game of GOMOKU:

```

10 REM GOMOKU
20 GOSUB 780
30 GOSUB 130
40 GOSUB 270
50 GOSUB 130
60 GOSUB 350
70 GOSUB 130
80 IF L>3 THEN PRINT:PRINT "I WIN!!":GOTO 1040
90 GOTO 40
100 E=A
110 E=E+N:IF A(E)<>Z THEN RETURN
120 K=K+1:GOTO 110
130 PRINT CHR$(28)
150 PRINT TAB(7);"1 2 3 4 5 6 7 8"
160 FOR A=1 TO 8:PRINT TAB(3);A;" ";

```

```

170 FOR B=2 TO 9
180 M=A(A*10+B)
190 IF M=C THEN PRINT "C ";
200 IF M=H THEN PRINT "H ";
210 IF M=46 THEN PRINT ". ";
220 NEXT B
230 PRINT A
240 NEXT A
250 PRINT TAB(7);"1 2 3 4 5 6 7 8"
260 RETURN
270 PRINT:PRINT
280 PRINT@416,"PLEASE ENTER YOUR MOVE...":PRINT
290 INPUT G
300 G=G+1
310 IF G<12 OR G>89 OR A(G)<>46 THEN 280
320 Z=H:PRINT@416,L#:PRINT@448,L#
330 A(G)=Z
340 RETURN
350 A=G
360 L=0
370 FOR X=1 TO 4:K=0:N=X(X)
380 GOSUB 100
390 N=-N:GOSUB 100
400 IF K>L THEN L=K
410 NEXT X
420 IF L>3 THEN PRINT:PRINT "YOU WIN!!":END
430 T=1
440 IF T<>2 THEN Z=C
450 IF T=2 THEN Z=H
460 G=0:H1=0:L=0
470 FOR A=12 TO 89
480 M=0
490 IF A(A)<>46 THEN 600
500 FOR X=1 TO 4:K=0:N=X(X)
510 GOSUB 100
520 N=-N:GOSUB 100
530 IF K>L THEN H1=0:L=K
540 IF L<>K THEN 570
550 IF T=1 AND L<4 OR (T=2 OR T=3) AND L<2 THEN 570
560 M=M+1
570 NEXT X
580 IF M<=H1 THEN 600

```

```

590 H1=M:G=A
600 NEXT A
610 IF H1<>0 THEN 680
620 T=T+1:IF T<>4 THEN 440
630 A=1
640 G=RND(77)+13
650 IF A(G)=46 THEN 680
660 A=A+1:IF A<100 THEN 640
670 PRINT:PRINT "I CONCEDE THE GAME":
      PRINT "TO A MASTER!!":END
680 A(G)=C
690 Z=C:A=G:L=0
700 FOR X=1 TO 4
710 K=0
720 N=X(X)
730 GOSUB 100
740 N=-N:GOSUB 100
750 IF K>L THEN L=K
760 NEXT X
770 RETURN
780 CLS:L$="":REM 25 SPACES
790 DIM A(100),X(4)
800 FOR C=1 TO 8
810 FOR B=2 TO 9
820 A(C*10+B)=46
830 NEXT B
840 NEXT C
850 FOR Q=1 TO 4
860 READ Z:X(Q)=Z
870 NEXT Q
880 DATA 1,9,10,11
890 H=ASC("H"):C=ASC("C")
900 PRINT:PRINT "ENTER 'Y' IF YOU WANT THE "
910 PRINT "FIRST MOVE, 'N' IF YOU DON'T"
920 N=0
930 N=N+1
940 A$=INKEY$
950 IF A$<>"N" AND A$<>"Y" THEN 930
960 CLS
970 IF A$="Y" THEN RETURN
980 FOR J=1 TO RND(12)
990 READ Z

```

```

1000 NEXT J
1010 A(Z)=C
1020 RETURN
1030 DATA 34,35,44,46,47,54,55,56,57,66,75,84
1040 SOUND 16,3:SOUND 16,2:SOUND 13,4
      :SOUND 18,2:SOUND 16,5
1050 SOUND 13,6:END

```

```

      *      *
**          ** * * **          **
** **      * ** *          ** **
**          * * * * *          **
** * * ** * *          * ** * * **
** ** ** ** * *          * * ** **
** *          * *          * *          **
      *** * * *          * * * *
      *** * *          * * * *
** *          * *          * *          **
** ** ** ** * *          * * ** **
** * * ** * *          * ** * * **
**          * * * * *          **
** **          * * * *          **
      *          *

```

# FOUR IN A ROW

In this game, FOUR IN A ROW, as its name suggests, the aim is to get four of your pieces (the H's) in a line in any direction, before the computer (using the C's) manages to do so.

You indicate your choice of move by specifying the column in which you want to move your piece. The piece then drops to the lowest available position within that column.

The computer plays this game fairly well, and surprisingly quickly, considering the number of times it can go through those loops within the program.

I was not particularly pleased when I finished the first hand-written version of this program, because it seemed to me that I had taken a 'brute force' approach to solving the problem. I was sure there would be a more clever way to do it. However, I continued with the program, and then entered it into the computer. It won the first game, even though it played a little oddly, so I knew I was onto a winner, despite the programming approach. Then, when thinking about it later I realised that if the program was correctly structured (as it was), had no redundant code (and it hasn't, as far as I can see), ran quickly and well, it did not need 'fiddling' to make the programming more tricky. Transparent code is always better than overly clever convoluted code which, although it may occupy less space, and may run a few microseconds faster, is almost impossible to modify.

So the program you have here is my 'brute force' version. It is generously supplied with REM statements so you have little trouble in working out what each section of code does. It should also prove fairly simple to modify, once you have played a few games with it in its present form.

In this version of FOUR IN A ROW, the computer always allows the human to have the first move, and bases its initial move on that made by the human. You may wish to modify the program so that there is an option for the computer to have the opening move.

Before we get to the listing of FOUR IN A ROW, here's one game played against the program:

```
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . H . . .
1 2 3 4 5 6 7
```

```
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . C . . .
. . H H C . .
1 2 3 4 5 6 7
```

```
. . . . .
. . . . .
. . . . .
. . . C . . .
. . H H . . .
. H C C C . .
H C H H C . .
1 2 3 4 5 6 7
```

```
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . C . . .
1 2 3 4 5 6 7
```

```
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . C . . .
H C H H C . .
1 2 3 4 5 6 7
```

```
. . . . .
. . . . .
. . . . .
. . . C . . .
C C H H H . .
H H C C C . .
H C H H C . H
1 2 3 4 5 6 7
```

```

. . C C . . .
H . C H . . .
C . C H . . .
C . H C H . H
C C H H H C C
H H C C C H H
H C H H C C H
1 2 3 4 5 6 7

```

```

. . C C . . .
H . C H . . .
C C C H . . .
C H H C H . H
C C H H H C C
H H C C C H H
H C H H C C H
1 2 3 4 5 6 7

```

This is the listing for FOUR IN A ROW:

```

10 REM FOUR-IN-A-ROW
20 GOSUB 1090:REM INITIALISE
30 GOSUB 860:REM UPDATE BOARD
40 GOSUB 680:REM WIN CHECK
50 GOSUB 980:REM HUMAN MOVE
60 GOSUB 860:REM UPDATE BOARD
70 GOSUB 680:REM WIN CHECK
80 GOSUB 110:REM COMPUTER MOVE
90 PRINT@320,L$:GOTO 30
100 REM *****
110 REM COMPUTER MOVE
120 PRINT@320,"PLEASE STAND BY FOR MY MOVE..."
130 B=10
140 B=B+1
150 IF A(B)=-9 THEN 180
160 IF A(B)=C THEN X=C:GOTO 210
170 IF A(B)=H THEN X=H:GOTO 210
180 IF B<77 THEN 140
190 GOTO 480
200 REM *****
210 REM FOUR IN A ROW DANGER/CHANCE?
220 REM ACROSS
230 IF A(B+1)=X AND A(B+2)=X AND A(B+3)=E AND
    A(B+13)<>E THEN 1220
240 IF A(B-1)=X AND A(B-2)=X AND A(B-3)=E AND
    A(B+7)<>E THEN 1230
250 IF A(B+1)=X AND A(B+2)=X AND A(B-1)=E AND
    A(B+9)<>E THEN 1240
260 IF A(B-1)=X AND A(B+2)=X AND A(B+1)=E AND
    A(B+11)<>E THEN 1250

```

```

270 IF A(B+1)=X AND A(B-1)=X AND A(B+2)=E AND
    A(B+12)<>E THEN 1260
280 IF A(B+1)=X AND A(B-1)=X AND A(B-2)=E AND
    A(B+8)<>E THEN 1270
290 IF A(B-1)=X AND A(B-2)=X AND A(B+1)=E AND
    A(B+11)<>E THEN 1250
300 REM DOWN
310 IF B>20 THEN IF A(B-10)=X AND A(B-20)=X AND
    A(B+10)=E THEN 1300
320 REM DIAGONALS
330 IF A(B+11)=X AND A(B+22)=X AND A(B-11)=E AND
    A(B-1)<>E THEN 1240
340 IF A(B+9)=X AND A(B+18)=X AND A(B-9)=E AND
    A(B+1)<>E THEN 1280
350 REM *****
360 REM MAKE/BLOCK THREE?
370 REM ACROSS
380 IF A(B+1)=X AND A(B+2)=E AND A(B+12)<>E THEN
    MOVE=B+2:GOTO 650
390 IF A(B+1)=X AND A(B-1)=E AND A(B+9)<>E THEN
    MOVE=B-1:GOTO 650
400 IF A(B-1)=X AND A(B-2)=E AND A(B+8)<>E THEN
    MOVE=B-2:GOTO 650
410 REM VERTICAL
420 IF A(B+10)=X AND A(B-10)=E AND A(B)<>E THEN
    MOVE=B-10:GOTO 650
430 REM DIAGONAL
440 IF A(B+9)=X AND A(B-9)=E AND A(B+1)<>E THEN
    MOVE=B-9:GOTO 650
450 IF B>11 THEN IF A(B+11)=X AND A(B-11)=E AND
    A(B-1)<>E THEN 1290
460 GOTO 180
470 REM *****
480 REM SINGLE MOVES
490 FOR N=1 TO 3
500 M(N)=0
510 NEXT N
520 COUNT=0
530 FOR B=11 TO 77
540 IF A(B)<>C AND A(B)<>H THEN 600
550 IF A(B+1)=E AND A(B+11)<>E THEN COUNT=COUNT+1
    :M(COUNT)=B+1

```

```

560 IF A(B-1)=E AND A(B+9)<>E THEN COUNT=COUNT+1
      :M(COUNT)=B-1
570 IF A(B-10)=E AND A(B)<>E THEN COUNT=COUNT+1
      :M(COUNT)=B-10
580 IF A(B-11)=E AND A(B-1)<>E THEN COUNT=COUNT+1
      :M(COUNT)=B-11
590 IF A(B-9)=E AND A(B+1)<>E THEN COUNT=COUNT+1
      :M(COUNT)=B-9

600 NEXT B
610 IF COUNT<>0 THEN 640
620 PRINT "I THINK WE SHOULD CALL IT A DRAW"
630 PRINT:PRINT:PRINT:END
640 MOVE=M(RND(COUNT))
650 A(MOVE)=C
660 RETURN
670 REM WIN CHECK
680 X=H
700 B=10
710 B=B+1
720 IF A(B)<>X THEN 770
730 IF A(B+1)=X AND A(B+2)=X AND A(B+3)=X THEN 800
740 IF B>30 THEN IF A(B-10)=X AND A(B-20)=X AND
      A(B-30)=X THEN 800
750 IF B>33 THEN IF A(B-11)=X AND A(B-22)=X AND
      A(B-33)=X THEN 800
760 IF B>27 THEN IF A(B-9)=X AND A(B-18)=X AND
      A(B-27)=X THEN 800

770 IF B<77 THEN 710
780 IF X=H THEN X=C:GOTO 700
790 RETURN
800 REM WIN FOUND
810 PRINT
820 IF X=H THEN PRINT "YOU'VE BEATEN ME, HUMAN!"
830 IF X=C THEN PRINT "I'VE DEFEATED YOU,
      HUMAN!":SOUND 16,3
835 IF X=C THEN SOUND 16,2:SOUND 13,4:SOUND
      18,2:SOUND 16,5

837 IF X=C THEN SOUND 13,6
840 END
850 REM *****

```

```

860 REM UPDATE BOARD
870 PRINT CHR$(28);
880 FOR K=10 TO 70 STEP 10
890 PRINT TAB(5);
900 FOR J=1 TO 7
910 PRINT CHR$(A(K+J));" ";
920 NEXT J:PRINT
930 NEXT K
940 PRINT TAB(5);"1 2 3 4 5 6 7"
950 PRINT:PRINT
960 RETURN
970 REM *****
980 REM HUMAN MOVE
990 PRINT@288,"YOUR MOVE...":PRINT
1000 PRINT@320,"WHICH COLUMN DO YOU WISH TO"
1010 INPUT "MOVE INTO":J
1020 Z=J
1030 Z=Z+10:PRINT@288,L$:PRINT L$:PRINT L$
      :PRINT L$:PRINT L$

1040 IF A(Z+10)=E THEN 1030
1050 IF A(Z)=E THEN A(Z)=H:RETURN
1060 PRINT@384,"YOU CAN'T MOVE THERE"
1070 GOTO 1000
1080 REM *****
1090 REM INITIALISE
1100 CLS
1110 DIM A(109),M(30),P(6)
1120 E=46
1130 H=72:C=67
1140 FOR B=1 TO 109:A(B)=E
1150 D=B-10*INT(B/10)
1160 IF D=0 OR D>7 OR B<11 OR B>77 THEN A(B)=-9
1170 NEXT B
1180 L$=""
      "
      :REM 31 SPACES

1190 RETURN
1220 MOVE=B+3:GOTO 650
1230 MOVE=B-3:GOTO 650
1240 MOVE=B-1:GOTO 650
1250 MOVE=B+1:GOTO 650
1260 MOVE=B+2:GOTO 650
1270 MOVE=B-2:GOTO 650
1280 MOVE=B-9:GOTO 650
1290 MOVE=B-11:GOTO 650

1300 IF A(B+20)<>E THEN MOVE=B+10:
      GOTO 650 ELSE 320

```

# CHECKERS

The game of CHECKERS has a long and honorable history. R. C. Bell (in his book *Discovering Old Board Games*, Shire Publications, Aylesbury, UK, 1980) says it was invented around 1100, "probably in the south of France, using Backgammon tablemen on a chequered chessboard with the Alquerque method of capture" (pp. 35 - 36). The *Encyclopedia of Sports, Games and Pastimes* (Fleetway House, London, c.1935) puts it much further back in time: "Forms of it were known in ancient Egypt, Greece and Rome, while the game was played in the mid-seventeenth century much as it is today." (p. 237).

Regardless of its age, it is a very popular game around the world, with many European countries having regional variations on the game of their own. Continental draughts (checkers is generally known as draughts outside the US), for example, is played on a board of 100 squares with each player starting the game with 20 pieces. It was developed in the early 1700's.

This CHECKERS program plays the game you are probably most familiar with. It plays swiftly, and reasonably well, although its lack of endgame strategy often leads to a dramatic collapse in the final moment of a game.

CHECKERS is played between you and the computer. Each of you is attempting to take (that is jump over to capture, then remove from the board) the other player's pieces, or to confine the opponent's pieces so no more moves are possible.

The game is generally played on a board with 64 squares, which are alternately light and dark. In

this program, the board (as you can see from the sample game we have shortly) is a series of dots, with your pieces shown as H's (for human) and the machine's pieces as C's (for clever).

When the game begins, you're at the bottom of the screen, playing up, and the computer is at the top playing down. As the program is currently set up, the computer always has the first move. If you want the first move then delete line 50.

All moves must be made along diagonals. Individual pieces can only move forward, that is towards the opponent's starting side. Pieces are converted into 'kings' when they reach the back rank across the board. Your kings are shown as K's, the computer's kings are dollar signs. Kings can move either forward or back. After capturing a piece, by jumping over it into the vacant square immediately behind the captured piece, you can move again if there is a further capture which can be made.

COMPUTER: 4  
HUMAN: 4

	1	2	3	4	5	6	7	8	
8	.	.	.	.	.	.	.	.	8
7	.	.	H	.	C	.	C	.	7
6	.	.	.	C	.	C	.	C	6
5	.	.	.	.	.	C	.	.	5
4	.	.	.	H	.	.	.	H	4
3	.	.	.	.	.	.	H	.	3
2	.	.	.	.	.	H	.	H	2
1	♠	.	H	.	♠	.	H	.	1
	1	2	3	4	5	6	7	8	

COMPUTER: 6  
HUMAN: 6

	1	2	3	4	5	6	7	8	
8	.	.	.	.	.	.	.	.	8
7	.	.	.	C	.	C	.	.	7
6	.	.	.	.	.	.	.	C	6
5	.	.	H	.	.	.	.	C	5
4	.	♠	.	.	.	H	.	H	4
3	.	.	.	.	.	.	H	.	3
2	.	.	.	.	.	C	.	H	2
1	.	.	.	.	.	.	H	.	1
	1	2	3	4	5	6	7	8	

COMPUTER: 9  
HUMAN: 10

```
      1 2 3 4 5 6 7 8
8     . . I . . . 8
7     . H . . . . 7
6     . . . . . . 6
5     . $ . . . . 5
4     . . . . . . 4
3     . . . . K . 3
2     . . . . . . 2
1     . . . . . . 1
      1 2 3 4 5 6 7 8
```

COMPUTER: 11  
HUMAN: 10

```
      1 2 3 4 5 6 7 8
8     . . . . . . 8
7     . . . . . . 7
6     . K . $ . . 6
5     . . . . . . 5
4     . . . . . . 4
3     . . . . . . 3
2     . $ . . . . 2
1     . . . . . . 1
      1 2 3 4 5 6 7 8
```

This is the complete listing for CHECKERS:

```
10 REM CHECKERS
20 GOSUB 1010:REM INITIALISE
30 REM DELETE LINE 50 FOR HUMAN
40 REM TO HAVE FIRST MOVE
50 GOTO 90
60 REM *****
70 GOSUB 600:REM PRINT BOARD
80 GOSUB 760:REM GET AND PRINT HUMAN MOVE
90 GOSUB 600:REM PRINT BOARD
100 GOSUB 130:REM MAKE MOVE
110 GOTO 70
120 REM *****
130 FOR X=1 TO 10:S(X)=0:NEXT X
140 SC=0:A=89
150 A=A-1
160 IF Q(A)<>C AND Q(A)<>C K THEN 240
170 B=0:IF A<29 THEN B=2
180 B=B+1
190 M=A+N(B)
200 IF M>88 OR M<11 THEN 240
210 IF (Q(M)=H OR Q(M)=HK) AND Q(M+N(B))=E THEN 280
230 IF B<2 OR (Q(A)=CK AND B<4) THEN 180
240 IF A>11 THEN 150
250 FL=0:IFQ(22)=C OR Q(24)=C ORQ(26)=C OR Q(28)=C
      THEN GOSUB1270
```

```
260 IF FL=1 THEN 570
270 GOTO 420
280 Q(M+N(B))=Q(A):Q(M)=E:Q(A)=E
290 CO=CO+1
300 GOSUB 600
310 A=M+N(B)
320 B=0
330 B=B+1
340 IF (A+2*N(B)<11 OR A+2*N(B)>88) AND B<4 THEN 330
350 M=A+N(B)
360 IF Q(M)=C AND B>3 THEN RETURN
370 IF (Q(M)=HK) AND Q(M+N(B))=E THEN 330
380 IF B<2 OR (Q(A)=CK AND B<4) THEN 330
390 RETURN
400 IF SC<10 THEN SC=SC+1
410 S(SC)=100*A+B+20: RETURN
420 IF SC=0 THEN 470
430 XC=RND(SC)
440 A=INT(S(XC)/100)
450 M=A+N(S(XC)-100*A-20)
460 GOTO 570
470 SC=SC+1:A=RND(88)
480 IF Q(A)<>C AND Q(A)<>CK THEN 550
490 B=0
500 B=B+1
510 M=A+N(B)
520 IF M>88 OR M<11 THEN 570
530 IF Q(M)=E THEN 570
540 IF B<2 OR Q(A)=CK AND B<4 THEN 500
550 IF SC<300 THEN 470
560 PRINT:PRINT"I CONCEDE THE GAME":END
570 Q(M)=Q(A):Q(A)=E
580 RETURN
590 REM *****
600 PRINT CHR$(28);
610 PRINT"      COMPUTER:"CO
620 PRINT"      HUMAN:"HU
630 PRINT"      1 2 3 4 5 6 7 8"
640 FOR F=80 TO 10 STEP -10
650 PRINT F/10;" ";
660 FOR G=1 TO 8:PRINT CHR$(Q(G+F));" ";:NEXT G
```



```

670 PRINT F/10: NEXT F
680 PRINT " 1 2 3 4 5 6 7 8 "
690 IF CO=12 OR HU=12 THEN 710
700 RETURN
710 IF HU=12 THEN PRINT "YOU HAVE WON!"
720 IF CO=12 THEN PRINT "I HAVE WON!"
730 PRINT "THANKS FOR THE GAME":END
740 REM*****
750 REM 99 TO CONCEDE THE GAME
760 REM HUMAN MOVE
770 PRINT@416, "ENTER YOUR MOVE":PRINT
780 PRINT@448, " ";:INPUT "FROM":A
790 IF A=99 THEN GOTO 730
800 IF Q(A)<> H AND Q(A) <> HK THEN 780
810 PRINT@448, " ";:INPUT "TO":B
815 IF Q(A)=H AND B<A THEN 810
820 IF Q(B)<>E THEN 810
825 PRINT@416, " ";:REM 15 SPACES
830 Q(B)=Q(A):Q(A)=E
835 PRINT@448, " ";:REM 10 SPACES
840 REM*****
850 FOR T=11 TO 17:IF Q(T)=C THEN Q(T)=CK
860 NEXT T
870 FOR T=82 TO 88:IF Q(T)=H THEN Q(T)=HK
880 NEXT T
890 REM*****
900 IF ABS(A-B)<12 THEN RETURN
910 TY=RND(10)
920 IF TY<3 THEN PRINT@416, "GOOD MOVE":GOSUB 1340
930 IF TY>7 THEN PRINT@416, "GDT ME!":GOSUB 1340
940 HU=HU+1:Q((A+B)/2)=E:GOSUB 600
950 FOR T=82 TO 88:IF Q(T)=H THEN Q(T)=H THEN Q(T)=HK
960 NEXT T
970 PRINT:INPUT "CAN YOU JUMP AGAIN(Y/N)":A#
975 PRINT@416, " "
980 IF A#<>"Y" THEN RETURN
990 A=B:GOTO 800
1000 REM*****
1010 REM INITIALISE
1020 CLS:PRINT "PRESS THE SPACE KEY"
1030 IF INKEY#<> " " THEN 1030
1040 CLS
1050 PRINT "PLEASE STAND BY"

```

```

1060 DIM Q(99),N(4),S(10)
1070 H=72:HK=75
1080 C=67:CK=36
1090 E=32:OF=-99
1100 FOR M=1 TO 99:Q(M)=OF:NEXT M
1110 FOR M=1 TO 64
1120 READ D,G
1130 Q(D)=G:NEXT M
1140 DATA 81,46,82,67,83,46,84,67,85,46,86,67,87,46
1150 DATA 88,67,71,67,72,46,73,67,74,46,75,67,76,46
1160 DATA 77,67,78,46,61,46,62,67,63,46,64,67
1170 DATA 65,46,66,67,67,46,68,67,51,32,52,46
1180 DATA 53,32,54,46,55,32,56,46,57,32,58,46
1190 DATA 41,46,42,32,43,46,44,32,45,46,46,32
1200 DATA 47,46,48,32,31,72,32,46,33,72,34,46,35,72
1210 DATA 36,46,37,72,38,46,21,46,22,72,23,46,24,72
1220 DATA 25,46,26,72,27,46,28,72,11,72,12,46,13,72
1230 DATA 14,46,15,72,16,46,17,72,18,46
1240 FOR M=1 TO 4:READ X:N(M)=X:NEXT M
1250 DATA -11,-9,11,9
1260 CO=0:HU=0:RETURN
1270 IF Q(22)=C AND Q(11)=E THEN A=22:M=11:FL=1:RETURN
1280 IF Q(22)=C AND Q(13)=E THEN A=22:M=13:FL=1:RETURN
1290 IF Q(24)=C AND Q(13)=E THEN A=24:M=13:FL=1:RETURN
1300 IF Q(24)=C AND Q(15)=E THEN A=24:M=15:FL=1:RETURN
1310 IF Q(26)=C AND Q(15)=E THEN A=26:M=15:FL=1:RETURN
1320 IF Q(26)=C AND Q(17)=E THEN A=26:M=17:FL=1:RETURN
1330 RETURN
1340 FOR Q=1 TO 1000:NEXT Q:RETURN

```

COMPUTER: 0  
HUMAN: 0

	1	2	3	4	5	6	7	8	
8	.	C	.	C	.	C	.	C	8
7	C	.	C	.	C	.	C	.	7
6	.	C	.	.	C	.	C	.	6
5	.	.	.	C	.	.	.	.	5
4	.	.	.	.	.	.	.	.	4
3	H	.	H	.	H	.	H	.	3
2	.	H	.	H	.	H	.	H	2
1	H	.	H	.	H	.	H	.	1
	1	2	3	4	5	6	7	8	

# KNIGHTSBRIDGE

It is played on a seven by seven board. The computer tells you which piece you must move (each piece is indicated by entering a two-digit number, the first being the co-ordinates of the square you're moving to down the side of the board, the second digit being the co-ordinates of the square across the top), and then leaves you to decide where you will move to. All pieces move like Chess knights, and all moves are checked to see if they are legal before you are allowed to move.

You capture by landing on top of an enemy piece. The first player to capture five of the enemy's seven pieces is the winner. There are no multiple jumps.

You'll find the computer plays swiftly and well.

```

MY SCORE IS 0
YOUR SCORE IS 0

 1 2 3 4 5 6 7
-----
 7 C C C C C C C 7
 6 . . . . . 6
 5 . . . . . 5
 4 . . . . . 4
 3 . . . . . 3
 2 . . . . . 2
 1 H H H H H H 1
-----
 1 2 3 4 5 6 7
I HAVE TO MOVE THE PIECE ON 77

```

```

MY SCORE IS 0
YOUR SCORE IS 0

 1 2 3 4 5 6 7
-----
 7 C C C C C C 7
 6 . . . . . 6
 5 . . . . C 5
 4 . . . . . 4
 3 . . . H . . 3
 2 . . . . . 2
 1 H H H H . H 2
-----
 1 2 3 4 5 6 7
YOU MUST MOVE THE PIECE ON 15
? 34

```

```

10 REM KNIGHTSBRIDGE
20 GOSUB 760 :REM INITIALISE
30 GOSUB 490 :REM PRINT BOARD
40 IF HU=5 OR CU=5 THEN 680 :REM END OF GAME
50 GOSUB 300 :REM COMPUTER MOVE
60 GOSUB 490 :REM PRINT BOARD
70 IF HU=5 OR CU=5 THEN 680 :REM END OF GAME
80 GOSUB 110 :REM PLAYER MOVES
90 GOTO 30
100 REM *****
110 REM PLAYER MOVES
120 Q=0
130 M=RND(66)+11
140 Q=Q+1
150 IF Q=500 THEN 680
160 IF H(M)<>72 THEN 130
170 PRINT"YOU MUST MOVE THE PIECE ON" M
180 INPUT N
190 IF N=99 THEN Q=500:GOTO 680
200 REM ### CHECK IF MOVE LEGAL ###
210 P=0
220 CT=1
230 IF M=Z(CT)=N THEN P=1
240 IF CT<8 THEN CT=CT+1:GOTO 230
250 IF P=0 THEN PRINT"ILLEGAL MOVE":GOTO 180
260 IF H(N)=67 THEN HU=HU+1:PRINT"WELL PLAYED!":FORR=170500:NEXT
270 H(N)=46:H(N)=72
280 RETURN
290 REM *****
300 REM COMPUTER MOVES
310 Q1=0
320 Q1=Q1+1
330 K=RND(66)+11
340 IF Q1=500 THEN 680
350 IF H(K)<>67 THEN 320
360 PRINT"I HAVE TO MOVE THE PIECE ON" K
365 FOR F=1 TO 3000:NEXT F
370 W=1
380 IF K+Z(W)<11 OR K+Z(W)>77 THEN 400
385 IF H(K+Z(W))=72 THEN PRINT "GOTCHA!!":FOR F=1 TO 1500:NEXT F
390 IF H(K+Z(W))=72 THEN CU=CU+1:GOTO 450
400 IF W<8 THEN W=W+1:GOTO 380

```

```

410 W=1
420 IF (K+Z(W)<11 OR K+Z(W)>77) AND W<8 THEN W=W+1:GOTO 420
430 IF H(K+Z(W))<>46 AND W<8 THEN W=W+1:GOTO 430
440 IF W=8 AND H(K+Z(W))<> 46 THEN Q1=500:GOTO 680
450 X=K:Y=K+Z(W)
460 H(X)=46:H(Y)=67
470 RETURN
480 REM #####
490 REM PRINT BOARD
500 CLS
510 PRINT TAB(8);"MY SCORE IS"CO
520 PRINT TAB(8);"YOUR SCORE IS"MU
530 PRINT
540 PRINT TAB(8);"1 2 3 4 5 6 7"
550 PRINT TAB(8);"-----"
560 FOR M=70 TO 10 STEP -10
570 PRINT TAB(5);M/10;
580 FOR N=1 TO 7
590 PRINT CHR$(H(M+N));" ";
600 NEXT N
610 PRINT M/10
620 NEXT M
630 PRINT TAB(8);"-----"
640 PRINT TAB(8);"1 2 3 4 5 6 7"
660 RETURN
670 REM #####
680 REM END OF GAME
690 GOSUB 490
700 IF HU=5 THEN PRINT"YOU HAVE BEATEN ME, YOU,HUMAN"
710 IF CO=5 THEN GOSUB 920
720 IF Q=500 THEN PRINT"I ACCEPT YOUR WISH TO CONCEDE"
730 IF Q1=500 THEN PRINT"I CONCEDE TO A MASTER"
740 END
750 REM #####
760 REM INITIALISE
770 CLS:PRINT"PLEASE STAND BY....HUMAN"
780 DIM H(100),Z(8)
790 I=0:Q1=0:Q=0
800 HU=0:CO=0:REM SCORES
810 FOR A=1 TO 99
820 IF A>77 OR A=70 OR A=60 OR A=68 OR A=69 OR A=50 THEN NEXT A

```

```

830 IF A=59 OR A=40 OR A=48 OR A=49 OR A=30 OR A=38 THEN NEXT A
840 IF A=39 OR A=20 OR A=28 OR A=29 OR A=18 OR A=19 OR A<11 THEN NEXT
850 H(A)=46
860 IF A>70 AND A<78 THEN H(A)=67
870 IF A>10 AND A<18 THEN H(A)=72
875 IF A>99 THEN 890
880 NEXT A
890 FOR B=1 TO 8:READ Z(A):NEXT A
900 DATA -8,-21,-12,-19,19,12,21,8
910 RETURN
920 PRINT"THIS VICTORY IS THE FIRST":PRINT"STEP IN OUR PLAN TO";
930 PRINT" TAKE":PRINT"OVER THE ENTIRE EARTH!":RETURN

```

# Adventure



# Adventure

Down into the depths of darkness you go. Armed with only your computer, and your keen mind, you have decided to take on the forces of evil.

You may be on a desert island, inside a haunted house, within a dungeon deep within the bowels of the earth, or trapped in a cave system on a planet 'somewhere in a galaxy far away'.

ADVENTURE games take place in all these scenarios, and a thousand more beside. Come with us now, as we discover the excitement of the worlds of ADVENTURE.

\*\*\*\*\*

The word ADVENTURE is used to describe the class of computer games in which the player moves through an alternative reality. In this 'otherworld' there are monsters to be fought, treasures to be discovered, maps to be made, and puzzles to be solved.

Many people feel that ADVENTURE games are the most exciting games which can be played with a computer. If you are interested in role-playing games like TSR Hobbies' Dungeons and Dragons (the game and name are registered trademarks of TSR), then you're certain to enjoy playing ADVENTURE.

I've included two ADVENTURE-like programs in this book. Both are reasonably difficult to solve, but are most enjoyable. Neither of them are so hard to work out you'll give up in frustration.

One feature of true ADVENTURE games is that the reality they model is consistent. That is, the world created within the ADVENTURE program is solid, and -- apart from any events specific to that game, such as an earthquake, or a magic spell -- the parts of

the world do not shift in a random fashion. If you walk past a 'gnarled and twisted oak, with the initials of your best friend carved in the trunk' when going down a certain woodland park, the same oak should still be there when you return. In a properly constructed ADVENTURE the rivers stay in place, the dungeon walls do not mysteriously move and shift every time you turn your back, and objects you put down in one cave within an underground labyrinth do not suddenly appear of their own volition a hundred caves away (unless, of course, they've been blessed with some form of magical auto-transportation).

Map-making is one of the true ADVENTURE-players skills and delight. Working your way through an imaginary (but self-consistent) environment, tackling monsters and collecting treasures, solving puzzles as you go up and down staircases and chutes, explore side tunnels, get lost in self-circling mazes, and so on is only fascinating if the world you are exploring is mappable. You should, for example, be able to expect both ends of a tunnel to always emerge in the same spots, so when you come across one entrance to that tunnel later on in a game from a different position, you will have a new, useful clue to further flesh out your map of the environment. All the 'worlds' created in the games in this section of your book are mappable.

Role-playing is part of life. There are sure to be many situations in your everyday activities when you, to some extent, are forced to adopt a role. Most jobs, unfortunately, expect you to behave as if the needs and goals of the company were your own needs and goals, that the enthusiasm you project each day to customers and fellow-workers were your own, inner enthusiasms, and so on. You may well find that, in new situations, you have learnt to mask your lack of ease by acting confident, and at ease, discovering that -- if you do it skillfully -- people take the projected role as if it were your true self.

The role-playing which takes place within the imaginary environments of an ADVENTURE game is far more interesting, and is generally far removed from the self of your everyday life. When you think about it, even playing games like 'Cops and Robbers' is an exercise in role-playing.

So, we've all had experience in role-playing, even if some of it was not as much fun as taking part in long-range campaigns to rescue a handsome prince, trapped within a tower. The ADVENTURES given in this book are by no means as elaborate as some of the ones offered for sale for your computer, but they are more carefully and consistently organised than many I have seen, and -- as a bonus -- give you frameworks within which you can construct ADVENTURE programs of your own.

Role-playing games which involve other people, such as TSR's D & D which I mentioned earlier, generally have a great deal of flexibility in development as a game proceeds. Under the control of a good Dungeon-master, a role-playing game can develop and evolve in directions which he or she had not even imagined when setting up the initial scenario. Unfortunately, computer role-playing games cannot be so flexible. In many ways when playing ADVENTURE with a computer, you are trying to solve a puzzle, or series of them, which the programmer has created. The computer can be used to help set up the initial scenario and determine where objects will be located, and perhaps where a particular stairwell will lead, but the overall shape to the environment is fixed by the programmer when the game is written.

You can see this in action, for example, in our program STRONGHOLD OF THE DWARVEN LORDS where you have to work your way through a maze to discover where the Dwarven treasure is stowed. The computer actually hides the treasure, and modifies the '3-D maze' to some extent, but that is all. Of course, the computer is also used to produce the feedback to the player as the game is underway, and its role there is very valuable.

Once upon a time, war campaigns were re-enacted on table tops. Napoleonic battles were fought and won on large sheets of card marked with rivers and woods, as players re-enacted major engagements within wars, or created their own. Wargamers found the pleasure of the games they played could be increased if attention were paid as much (if not more) to the individuals they were fighting with, rather than just with squads of soldiers. This attention to individual identity lead directly into the role-playing games we have today, as gamers discovered how rewarding it was to take on the role of a particular leader.

Our Dungeons and Dragons developed directly from the war games where players took on the identities of individuals within battles. Two gamers, Dave Arneson and E. Gary Gygax decided to codify an entirely separate reality, which they initially gave to the world in 1974 as a boxed set of three booklets under the title Dungeons and Dragons. The game became an enormous success, and now hundreds of thousands of people spend a significant portion of their leisure plunging into the depths of the earth, as clerics, trolls, magicians and sorcerers, exploring, fighting, conquering -- and occasionally dying -- as they go.

The original D & D idea soon spawned a host of other games, such as Runequest (Chaosium Ltd, Albany, California), Bushido and Chivalry & Sorcery (Fantasy Games Unlimited, Roslyn, New York), Tunnels & Trolls (Flying Buffalo, Inc., Scottsdale, Arizona), Traveler (Game Designers' Workshop, Bloomington, Illinois) and Heroes of Olympus (Task Force Games, Amarillo, Texas).

Will Crowther and Don Woods entered computer history in 1976 by putting the world's first role-playing game -- just called ADVENTURE -- onto a mainframe computer at Stanford University. (The program was actually written by Crowther, and elaborated by

Woods, although today both men are generally credited with development of the program.) Although the game quickly became a cult-pursuit all over America, the floodgates to computer ADVENTURE-gaming did not open until microcomputers put machine-power -- and computer time -- into the hands of millions. Cheaper memory enabled extremely elaborate adventures to be created, and so now the computer ADVENTURE gamer is extremely well served.

Crowther and Woods perhaps created better than they knew. Even today, despite the multitude of programs available, the original ADVENTURE is still going strong. At least five companies at present are selling versions of the original ADVENTURE for the Apple alone.

This original program was written in Fortran, and when it was added to a free software-exchange library (Decus) organised by the Digital Equipment Computer Users' Society (most of whom used Dec's DCP-11 systems), it soon became the most-used program in the library. Two years later, another adventure program, Dungeon was added to Decus. This program which took the ADVENTURE idea much further than Crowther's program, was written by Tim Anderson, Marc Blank, Bruce Daniels and Dave Lebling, all of the Programming Technology Division of the MIT Laboratory for Computer Science.

The Addison-Wesley Book of Apple Computer Software (edited by Jeffrey Stanton and John Dickey, The Book Co., Lawndale, California, 1981) devotes 29 pages to describing fantasy, role-playing, adventure and war simulation games available for the Apple II. If the support for that computer is so good in this field of software, it indicates that many other popular computers are also well-supported.

Although there is a trend towards graphic-based adventures, rather than the classic text-based ones, the attraction of text programs continues. Just as

television did not make radio obsolete -- because the imagination is stronger than anything man can make -- so graphic adventures have not taken away the special value of text-based games, where the mind is triggered to produce worlds far more fantastic than any that a clever programmer could draw using the sophisticated graphics available on many of today's machines.

One of the leading companies in the world producing ADVENTURE programs is Adventure International, founded by Scott Adams. His company has some of the most imaginative displays at computer shows of anyone in the games software field, and it is good to know his programs are as exciting as his show displays. Make sure you experience the pleasure of playing a Scott Adams adventure program in due course, and you'll see how far the genre has been extended.

I met Scott in Atlanta one day at a computer show and found him friendly, quiet and reserved. Somehow, I'd expected him to be like a warlock from one of his dungeons, or a dragon from one of the cave systems. However, some behaviour was more in keeping with my expectations. It was Halloween, and Scott and his crew were all dressed to the nines as characters out of some of the programs. The same approach to life which had him tell his staff to come to work in fancy dress permeates his programs. Start with the programs here, which are much simpler to solve than programs such as Scott's, and then graduate to the commercial programs. Perhaps his program Adventureland would be a good place to begin, before you move on to some of his other games, and onto those produced by other companies.

\*\*\*\*\*

I guess it's time now to introduce you to the adventure programs in this book. If you'd like to know more about the field, to help you write your

own programs, there is a vast body of literature which is of interest. It includes:

- PUTTING ADVENTURE IN ADVENTURE GAMES - article in Creative Computing, August 1981, by Robert Plamondon
- GRAPHIC ADVENTURES ON THE ATARI - article in Creative Computing, August 1982, by John Anderson
- FANTASY GAMES (parts one and two) - articles in Creative Computing, issues of March 1981 (part one) and May 1981 (part two) by David Lubar
- AN ADVENTURE IN SMALL COMPUTER GAME SIMULATION - article by Scott Adams (when he was just about unknown!) in Creative Computing, way back in August 1979
- KINGS OF THE CASTLE WALLS - article in Practical Computing, March 1982, by Dennis Ellis
- ADVENTURE WRITING - this is an immensely valuable 16-page booklet, distributed by Aardvark-80, 2352 S. Commerce, Walled Lake, MI 48088 (313 669 3110) for around \$5.00. It explains the whole background of adventure game writing, and includes a complete adventure program, Deathship. If you read nothing else in the field, you should get hold of this booklet if you really want to discover how to write adventure games
- FANTASY ROLE-PLAYING GAMES - Holmes, J. Eric (Hippocrene Books, Inc., New York, 1981)
- WHAT IS DUNGEONS AND DRAGONS? - Butterfield John, Parker Philip, and Honigmann David (Penguin Books, New York, 1982)
- DICING WITH DRAGONS - AN INTRODUCTION TO ROLE-PLAYING GAMES - Livingstone, Ian (Routledge & Kegan Paul, London Melbourne and Henley)



# STRONGHOLD OF THE DWARVEN LORDS

Deep beneath the earth you go, far into the Dwarven Heartland. Danger is on every side as you descend, but your greed draws you on. Searching through the dusty stacks of uncatalogued manuscripts in room 546B of the British Museum, you came across a faded, and almost illegible map to a Dwarven hoard of gold, and since that day, you have been obsessed with the idea of finding it.

As you go down into the labyrinth, you realise that the Dwarven Lords, who secreted the gold here 7389 years ago, have long since become extinct, so the main danger you face is from the layout of the cavern system itself, rather than from Guards of the Stronghold.

In STRONGHOLD OF THE DWARVEN LORDS you are in a cavern which holds the gold. Each time you play this game, the gold can be in one of three places. The only information you get as to your progress is information provided by the Dwarven Source Beam which you found as you made your way into the cave system. This gives you feedback after each move as to the location of the gold, but you need to learn how to interpret the information it gives you before you'll be able to make much use of it.

The other information you get is in the form of a

straight statement regarding the directions you can move from your present position. After each move ('step'), you'll be given a screen display like the following:

STEP NUMBER 49

NORTH: WALL  
SOUTH: OPEN  
EAST: WALL  
WEST: OPEN

THE DWARVEN SOURCE BEAM READS 10

Then you'll be asked to enter the direction you want to move, as follows:

WHICH DIRECTION DO YOU WANT  
TO MOVE?

N - NORTH, S - SOUTH  
E - EAST, W - WEST, H - HELP

You will not, as you probably realise, be allowed to walk through walls.

The aim of the game, needless to say, is to reach the Dwarven riches as quickly as possible. To aid you, there is a map of the cavern system from above, which you can call up from time to time.

There are two catches to calling up a map of this type. Firstly, although you are shown, the location of the gold is not. Secondly, each time you enter 'H' to get a look at the map, you'll be penalised 15

steps. Therefore, it is in your interests to use the Help option as few times as possible.

```

NORTH
#####
# ##### #
## ##### #
## ##### #
## ##### #
## ##### #
## ##### #
##### # #
##### # #
## * # #
## ##### #
## ##### #
##### # #
##### # #
# # #
#####
SOUTH

```

Although there are some features which are common to each and every cavern generated by this program (dictated by the data statements from 860 to 950), the cavern is not identical from run to run. As well as that, as I mentioned earlier, the gold itself can be in one of three locations within the cave system.

The output of the program, as it is given in terms of the directions you can take from your present position, could be modified to produce a '3-D maze' output. You know, from move to move, which direction you're facing, and the computer knows which directions from that position represent possible moves. You would need to add a subroutine which drew a picture of the situation immediately in front of you, choosing from a bank of pictures (one with the area straight in front of you clear but with the walls solid on either side of you, another with extrances to the right and to the left but with the path straight ahead blocked off, and so on) to

produce a continuous picture of the scene ahead. This can be most effective, and may well be worth implementing on your system.

Now, here is the listing of STRONGHOLD OF THE DWARVEN LORDS so you can start trekking for gold:

```

10 REM STRONGHOLD OF THE DWARVEN LORDS
20 GOSUB 640
30 GOSUB 480
40 REM #####
50 M=M+1:IF M>150 THEN GOTO 960
60 CLS:PRINT:PRINT
70 PRINT "STEP NUMBER*M"
80 PRINT
90 PRINT"NORTH: ";
100 IF A(D+1,E)=S THEN PRINT"OPEN"
110 IF A(D+1,E)=I THEN PRINT"WALL"
120 PRINT "SOUTH: ";
130 IF A(D-1,E)=S THEN PRINT"OPEN"
140 IF A(D-1,E)=I THEN PRINT"WALL"
150 PRINT "EAST: ";
160 IF A(D,E+1)=S THEN PRINT"OPEN"
170 IF A(D,E+1)=I THEN PRINT"WALL"
180 PRINT "WEST: ";
190 IF A(D,E-1)=S THEN PRINT"OPEN"
200 IF A(D,E-1)=I THEN PRINT"WALL"
210 PRINT
220 PRINT"THE DWARVEN SOURCE BEAM READS";
230 PRINT (90*ABS(I-D))+10*ABS(Y-E)
240 REM #####
250 PRINT
260 PRINT"WHICH DIRECTION DO YOU WANT"
270 PRINT"TO MOVE?"
280 PRINT:PRINT"N-NORTH, S-SOUTH"
290 PRINT"E-EAST, W-WEST, H-HELP"
295 Z=INKEY$
300 A$=INKEY$:IF A$="S"OR A$="N"OR A$="E"OR A$="W"OR A$="H" THEN 310
305 GOTO 300
310 IF A$="N" AND A(D+1,E)=I THEN 300
320 IF A$="S" AND A(D-1,E)=I THEN 300
330 IF A$="E" AND A(D,E+1)=I THEN 300
340 IF A$="W" AND A(D,E-1)=I THEN 300

```

```

790 FOR F=1 TO 68
800 READ B,C
810 A(B,C)=S
820 NEXT F
830 M=-15
840 RETURN
850 REM #####
860 DATA 2,2,2,3,2,4,2,5,2,6,2,7
870 DATA 3,7,4,7,5,7,5,6,5,5,4,5,3,6,3
880 DATA 7,3,7,4,7,5,7,6,7,7,8,7,9,9,8
890 DATA 9,9,10,8,10,7,10,6,10,5,10,4,8,8
900 DATA 10,3,11,3,12,3,13,3,14,3,14,2,7,10
910 DATA 6,10,5,10,4,10,3,10,2,10,2,11,2,12
920 DATA 2,13,2,14,6,11,6,12,6,13,6,14,7,12
930 DATA 14,12,8,12,8,14,9,12,9,13,9,14,10,12
940 DATA 11,9,11,10,11,11,11,12,12,9,13,9,13,10
950 DATA 13,11,13,12,13,13,13,14,14,14
960 SOUND 8,6: SOUND 6,6: SOUND 4,8: FOR J=1 TO 1000: NEXT J
970 PRINT: PRINT: PRINT "AFTER *M* STEPS, YOU HAVEN'T BEEN"
980 PRINT "ABLE TO FIND THE SMARVEN": PRINT TAB(16); "TREASURE!!"
990 PRINT "SHAME ON YOU...."
1000 GOSUB 500: FOR J=1 TO 2000: NEXT: END

```

# THE DUKE OF DRAGONFEAR

THE DUKE OF DRAGONFEAR puts you into another grid system world, but one which is populated with a range of fearsome possibilities including pits containing quicksand or dragons, as well as magic caves which can transport you at random within the land of Dragonworld, and caves filled with Dragon's gold.

When you first run the program, you'll get this quick peek at a map of Dragonworld:

```

X X X X X X X X X X
X . D D $ ? D . . X
X D X . . X . . . X
X D . . . . Q . X X
X Q . . $ . . . ? X
X . . . H . . $ ? X
X $ $ . . . X . ? X
X . X . . . . . X
X . Q Q . . . . . X
X X X X X X X X X X

```

In this map, X represents a wall you cannot pass or a cave you cannot enter, H (for human) is you starting in about the centre of the system, \$ is the treasure caves, ? the magic transportation caves, D represents dragons and Q tells you that cave contains quicksand. Obviously this is a world filled with potential pleasures as well as dangers.

If you're a veteran of computer games, you may well recognise that THE DUKE OF DRAGONFEAR is a development of the Hunt the Wumpus games. Wumpus is one of the grand old standards of computer games, in which

you are in a cave system, looking for one or more mythical beasts known, in the singular (and these are most singular beasts) as Wumpus. Wumpii we are assured, by those who care about such things, is the plural. Anyway, in essence, a grid is set up in such games, and objects, people, effects, monsters and whatever else you choose are placed at locations on the grid. From time to time, if you so decide, one or more of the contents of specific cells of the grid may move. The human player, of course, can also move from cell to cell within the grid.

Generally, there is no overall visible map of the system, although some programs (such as THE DUKE OF DRAGONFEAR) do provide such information for players. The very first Wumpus game was written by Gregory Yob in 1975. In David Ahl's superb book More BASIC Computer Games (Creative Computing Press, Morristown, New Jersey, 1979), Mr Yob explains how he was visiting People's Computer Company, in Menlo Park, California, to see the programs they were developing, and he noticed there were three 'hunt the something' programs being written, each on a 10 X 10 grid, and each of which gave feedback to the player in terms of 'no, no...try to the northwest'.

Disenchanted, Mr Yob returned to a lonely time-sharing terminal (back in those ancient days, gentle reader, human beings did not have little computers at their beck and call) and hammered out the program which eventually became the very first 'Hunt the Wumpus'. Mr Yob had the creature dwelling in a dodecahedron-shaped cave system.

Mr Yob's inventive genius did not stop at this point. He peopled the cave with 'magic bats' which moved you from place within the system at random, with bottomless pits, and ensured that the arrows with which you were equipped when you entered the system could fly round corners.

You can see now how this framework became the heart

of the Wumpus programs which have proliferated around the world and which formed the very basic framework upon which THE DUKE OF DRAGONFEAR is based. This program gives you more control on the outcome of the program, adds treasure to the caves, and imposes a time-limit to your game.

As well, you are equipped with a 'magic amulet' which can see into the caves which are around you. Unfortunately, because you're not much of a whizz when it comes to evoking magic, the old amulet doesn't work too well. It can only tell you of the contents of one of the eight caves surrounding the one in which you stand, and worse than this does not tell you which of the eight it is referring to. It

```

X X X X X X X X X X
X . D D $ ? D . . X
X D X . . X . . . X
X D . . . . Q . X X
X Q . . . . . . . X
X . . . . . $ ? X
X $ $ . . . X . ? X
X . X . . . . . X
X . Q Q H . . . . X
X X X X X X X X X X

```

is not as bad as it may sound. You'll be surprised at how much use you can make of the limited information the amulet provides. As well, you'll get a brief glimpse of the cave system from above at the start of the game, and from time to time as it goes on.

This is the program listing to enable you to become  
THE DUKE OF DRAGONFEAR:

```

10 REM DUKE OF DRAGONFEAR
20 GOSUB 1280
30 CLS:PRINT:PRINT:PRINT
40 GOSUB 1200
50 Q=RND(7)-1
60 IF Q=0 AND E<>55 THEN GOSUB 1200
70 CLS:PRINT:PRINT:PRINT "DUKE "A#", YOU ARE IN CAVE"E
80 IF G>0 THEN PRINT "YOU ARE CARRYING $"G"
                                WORTH OF GOLD"
90 GOSUB 760
100 PRINT:PRINT "YOU HAVE"25-H"UNITS OF CHARISMA LEFT"
110 PRINT:PRINT "WHAT DO YOU WANT TO DO?"
120 PRINT "N - MOVE NORTH, S - MOVE SOUTH"
130 PRINT "E - MOVE EAST, W - MOVE WEST"
140 PRINT "F - FIGHT A DRAGON, Q QUIT"
150 INPUT Z$:U=0:IF Z$="N"OR Z$="S"OR Z$="E"OR Z$="W"
                                OR Z$="F" THEN 160
155 IF Z$<>"Q" THEN 150
160 IF Z$="N" AND A(E-10)=88 OR Z$="S" AND A(E+10)
                                =88 THEN 1620
162 IF Z$="E" AND A(E+1)=88 OR Z$="W" AND A(E-1)
                                =88 THEN 1620
170 IF Z$="Q" THEN Q=9:GOTO 1170
180 A(E)=46:IF Z$="N" THEN E=E-10
190 IF Z$="S" THEN E=E+10
200 IF Z$="E" THEN E=E+1
210 IF Z$="W" THEN E=E-1
220 IF Z$="F" THEN GOSUB 930
230 IF A(E)=63 THEN GOSUB 310:REM MAGIC
240 IF A(E)=68 THEN GOSUB 400:REM DRAGON
250 IF A(E)=81 THEN GOSUB 550:REM QUICKSAND
260 IF A(E)=36 THEN GOSUB 630:REM GOLD
270 H=H+1:IF H=25 THEN Q=9:GOTO 1160
280 GOSUB 1610
290 GOTO 50
300 REM *****
310 REM MAGIC
320 PRINT:PRINT "DUKE "A#", YOU'VE STUMBLD"
330 PRINT "INTO A MAGIC CAVE, AND NOW YOU'LL"
340 PRINT "BE WHISKED OFF TO ANOTHER CAVE...."

```

```

350 GOSUB 1610
360 A(E) = 46
370 E=RND(76)+12: IF A(E)=88 THEN 370
380 RETURN
390 REM *****
400 REM DRAGON
410 PRINT "YOU HAVE WANDERED INTO A DRAGON'S"
420 PRINT "LAIR...START SAYING YOUR PRAYERS"
430 GOSUB 1610
440 M=RND(10000)/10000:IF M<.2 THEN PRINT "IT HAS
                                FLOWN AWAY":RETURN
450 PRINT "IT AWAKENS...AND IT HAS SEEN YOU"
460 GOSUB 1610
470 IF M>.849999 THEN PRINT "BUT IT HAS RECENTLY EATEN"
475 IF M>.849999 THEN PRINT "AND SO GOES BACK TO
                                SLEEP":RETURN
480 PRINT "AND NOW IT ATTACKS....."
490 GOSUB 1610
500 IF M>.95 THEN PRINT "BUT YOU FIGHT BACK...AND
                                WIN":RETURN
510 PRINT "GOODBYE DUKE "A#
520 GOSUB 1610:SOUND 1,8:SOUND 1,8:SOUND 1,8
530 Q=9:GOTO 1160
540 REM *****
550 REM QUICKSAND
560 FOR J=1 TO 12
570 FOR K=1 TO J:PRINT " ";:NEXT K
580 PRINT "HORRORS...QUICKSAND!":SOUND 31-J,1
590 FOR O=1 TO 100:NEXT O
600 NEXT J
610 Q=9:H=0:GOTO 1160
620 REM *****
630 REM TREASURE
640 K=RND(100)+100
650 FOR J=1 TO 12
660 FOR Z=1 TO J:PRINT " ";:NEXT Z
670 PRINT "TREASURE ! ! !"
680 FOR O=1 TO 150:NEXT O
690 NEXT J:SOUND 16,2:SOUND 18,2:SOUND 20,2:SOUND 21,5
700 GOSUB 1610
710 PRINT:PRINT "YOU'VE FOUND A HOARD OF"
720 PRINT:PRINT "DRAGON-GOLD, WORTH $"K
730 B=B+K

```

```

740 RETURN
750 REM *****
760 REM AMULET DETAILS
770 Y=1
780 L=A(E+P(Y))
790 IF L<>46 THEN 820
800 IF Y<8 THEN Y=Y+1:GOTO 780
810 IF L=46 THEN RETURN
820 PRINT "YOUR AMULET SIGNALS THAT"
830 PRINT "THERE IS ";
840 IF L=88 THEN PRINT "A SOLID WALL";
850 IF L=63 THEN PRINT "A MAGIC CAVE";
860 IF L=68 THEN PRINT "A DRAGON";
870 IF L=81 THEN PRINT "QUICKSAND";
880 IF L=36 THEN PRINT "GOLD";
890 PRINT " NEARBY"
900 GOSUB 1610
910 RETURN
920 REM *****
930 REM ATTACK DRAGON
940 PRINT
950 AR=AR-1:IF AR=0 THEN PRINT "YOU HAVE USED UP ALL"
955 IF AR=0 THEN PRINT " ARROWS.....":
      GOSUB 1610:RETURN
960 PRINT "YOU HAVE"AR"ARROWS IN YOUR QUIVER"
970 SS=0
980 PRINT "WHICH DIRECTION DO YOU WANT"
990 INPUT "TO SHOOT IN (N,S,E,OR W)";S$
1000 IF S$="N" AND A(E-10)=68 THEN SS=1:YT=E-10
1010 IF S$="S" AND A(E+10)=68 THEN SS=1:YT=E+10
1020 IF S$="E" AND A(E+1)=68 THEN SS=1:YT=E+1
1030 IF S$="W" AND A(E-1)=68 THEN SS=1:YT=E-1
1040 PRINT
1050 IF SS=0 THEN PRINT "THERE WAS NO DRAGON THERE...."
1055 IF SS=0 THEN PRINT "YOU HAVE WASTED AN
      ARROW!!":GOTO 1150
1060 PRINT "WELL DONE, DUKE "A$
1070 PRINT "YOU HAVE HIT"
1080 PRINT "A FEROCIOUS DRAGON"
1090 GOSUB 1610
1100 IF RND(10)>3 THEN 1140
1110 PRINT "YOU KILLED IT!!":A(YT)=46:K=RND(100)+100

```

```

1120 PRINT
1130 PRINT "YOU ARE REWARDED WITH $"K:G=G+K:GOTO 1150
1140 PRINT "BUT YOU ONLY WOUNDED IT....."
1150 GOSUB 1610:RETURN
1160 IF H<1 THEN PRINT "ALL YOUR CHARISMA IS"
1165 IF H<1 THEN PRINT "....EXHAUSTED....":GOSUB
      1610:GOTO 1180
1170 PRINT "YOU HAVE"25-H"UNITS OF CHARISMA LEFT"
1180 IF G>0 THEN PRINT "YOU AMASSED $"G"WORTH OF GOLD"
1190 PRINT:PRINT
1200 A(E)=72
1210 FOR J=1 TO 100
1220 PRINT CHR$(A(J));" ";
1230 IF 10*INT(J/10)=J THEN PRINT
1240 NEXT J
1250 GOSUB 1610
1260 IF Q=9 THEN END
1270 RETURN
1280 CLS:PRINT:PRINT:PRINT
1290 PRINT " WELCOME TO THE WORLD OF":
      PRINT " DRAGONFEAR"
1300 PRINT "YOUR MISSION IS TO EXPLORE THE"
1310 PRINT "CAVES OF DRAGONWORLD,LOOKING FOR"
1320 PRINT "TREASURE, AND ATTEMPTING TO SLAY"
1330 PRINT "THE FEARSOME DRAGONS WHO LIVE IN"
1340 PRINT "THE CAVES. WHAT IS YOUR NAME";
1350 INPUT A$
1360 CLS:PRINT:PRINT "ALL HAIL, DUKE "A$
1370 PRINT:PRINT "YOU START THIS EXPLORATION"
1380 PRINT "WITH 25 UNITS OF CHARISMA, AND"
1390 PRINT "YOU MUST COMPLETE YOUR TASK"
1400 PRINT "BEFORE THE CHARISMA IS WORN OUT."
1410 PRINT " YOU LOSE ONE UNIT FOR EVERY":
      PRINT,"MOVE YOU MAKE"
1420 T$=INKEY$:T$=INKEY$
1430 PRINT:PRINT "PRESS ANY KEY TO BEGIN"
1440 IF INKEY$="" THEN 1440
1450 CLS:PRINT@224,"PLEASE STAND BY, DUKE "A$
1460 DIM A(100):H=0:Q=0:L=0:G=0:AR=6
1470 FOR B=1 TO 100:A(B)=46
1480 IF B<120RB>90OR10*INT(B/10)=BDR10*INT(B/10)
      =B-1 THEN A(B)=88

```

```
1490 NEXT
1500 FOR B=1 TO 5:RESTORE:FOR D=1 TO 5
1510 Z=RND(76)+12:IF A(Z)=88 THEN 1510
1520 READ C:A(Z)=C
1530 NEXT D,B
1540 DATA 88,63,68,81,36
1550 FOR B=1 TO 8:READ Z:P(B)=Z:NEXT B
1560 DATA -11,-10,-9,-1,1,9,10,11
1570 E=55
1580 RETURN
1590 REM
1600 REM
1610 FOR O=1 TO 3000:NEXT O:RETURN
1620 PRINT "YOU CANNOT MOVE THAT WAY!":GOTO 150
```

# Simulations



anatomie

anatomie

anatomie

anatomie

anatomie



# Simulations

Simulations are ways of producing a counterfeit reality. In contrast to ADVENTURE, where the reality created via the computer is often magical and dream-like, the worlds accessed through computer simulations are generally more down to earth.

In a simulation, the computer manipulates variables in accordance with formulae you've specified, keeping tabs on the unfolding of the situation you've generated, and taking the place of the environment in terms of reacting to your input.

Simulations attempt to replicate life. However, because reality is notoriously hard to pin down and limit, we need to do some pretty drastic simplification before we can produce a workable simulation.

Despite this simplification -- as you'll soon see -- a well-written simulation can imitate life to an uncanny extent. Cause and effect are linked as accurately as you can devise formulae for them. The random number generator can take the place of such things as changes in the weather, in population numbers, in the behavior of molecules in a gas or the way a colored ink is dispersed through a clear liquid.

In this section of the book we have three simulations. One of them -- MISTRESS OF XENOPHOBIA -- is simple to the point of absurdity, but it is great fun to play, as you tinker with the fate of an entire planet. The other two -- RURAL PURSUITS (where you run a farm with extremely fickle workers) and CHAIRMAN OF THE BOARD (where you run a factory that could produce any one of a range of products including kazoes, pith helmets or sky hooks) -- are a little more serious, and the formulae employed in them produce results a little closer to 'real life'.

Despite the seriousness of the last two I mentioned, they are still great fun to play, as you'll see when you get them up and running.

If you want to know more about writing simulations the booklet Designing Classroom Simulations by Glenn Pate and Hugh Parker, Jr., will be of help. As well, Creative Computing magazine has printed two articles which are very valuable. These are "How to write a computer simulation" by the magazine's erstwhile editor, David Ahl, which appeared in the Jan/Feb (1978) issue of the magazine (and is the text of a talk David presented at the Conference on Computers in the Undergraduate Curricula in Pullman, Washington, in June 1974), and "Strategies for Successful Simulation" by Bruno B Wolff, Jr. which was in the August 1981 issue. It is worth checking your library for these back issues, as you'll learn a lot of value by reading them. Of particular interest in the Wolff article is a set of four equations which link cause and effect and can easily be adapted for a simulation you write.

# MISTRESS OF XENOPHOBIA

We'll start our exploration of simulations with the silliest one of the lot, MISTRESS OF XENOPHOBIA, which is based on a program written by Alastair Gourlay, a talented programmer who lives in Glasgow.

There's no need to explain MISTRESS, as the program gives you all the prompts you could need.

```
20 GOSUB 700
30 REM *****
40 FOR Y=1 TO 20
50 CLS
60 PRINT:PRINT
70 PRINT"MISTRESS OF XENOPHOBIA,A REPORT"
80 PRINT"FOR YOU FROM THE OFFICE OF INFO
      -MATION REGARDING THE";
90 PRINT" STATE OF":PRINT"YOUR PLANET"
100 PRINT"IN THIS YEAR OF GRACE,"1994+Y
120 PRINT"THE PLANETS POPULATION IS"INT(P+INP(H))
130 GOSUB 880
140 PRINT:PRINT"THE GROVELLING PEASANTS COULD "
150 PRINT"WORK SOME"L"ACRES THIS YEAR..."
160 GOSUB 880
170 PRINT"YOUR TREASURY HOLDS GOLD,GEMS"
180 PRINT"AND COINS WORTH $"INT(U)
190 GOSUB 880
200 PRINT:PRINT "TIME TO ISSUE A DECREE...."
210 PRINT:PRINT"HOW MUCH LAND IS TO BE FARMED"
220 INPUT"THIS YEAR";W
230 U=U-W*10
240 IF U<1 THEN 780
250 L=L+W
260 GOSUB 880
270 PRINT:PRINT"AND HOW MUCH WILL YOU SPEND"
280 INPUT"ON FOOD FOR THE PEASANTS";W
```

```
290 U=U-W*10
300 IF U<1 THEN 780
310 R=RND(10)
320 GOSUB 880
330 IF P-W*R*5>P/4 THEN GOSUB 580
340 P=P+RND(1000)/1000*(W*R*5-P)
350 IF P>149 THEN 410
360 GOSUB 880
370 PRINT"THE POPULATION OF XENOPHOBIA IS"
380 PRINT"NOW DOWN TO"INT(P)"AND THAT"
390 PRINT"JUST AIN'T ENOUGH, YOUR XEND":SOUND 8,8:
      SOUND 7,7
395 SOUND 6,6:SOUND 5,5:SOUND 4,4:SOUND 3,3:
      SOUND 2,2:SOUND 1,1

400 GOTO 480
410 U=U+INT(P*L)/93
420 NEXT Y
430 PRINT:PRINT:PRINT"WELL, YOUR XENOPHOBIC, THAT'S"
440 PRINT"THE END OF YOUR 20 YEAR DOMINAT-ION OF
      OUR LITTLE";
450 PRINT"PLANET.":PRINT"YOU MANAGED TO"
460 PRINT"ACCUMULATE SOME $"U"WHICH"
470 PRINT"I GUESS AIN'T TOO BAD...."
480 GOSUB 880
490 PRINT:PRINT"IF YOU'D LIKE ANOTHER SHOT"
500 PRINT"RULING, THEN JUST PRESS 'Y' OR"
510 PRINT"PRESS 'N' ...."
520 A$=INKEY$
530 IF A$<>"Y" AND A$<>"N" THEN 520
540 IF A$="Y" THEN RUN
550 PRINT"YOUR WISH IS MY COMMAND"
560 PRINT"OH MISTRESS OF XENOPHOBIA":END
570 REM *****
580 PRINT"OH DEAR! THERE'S NOT ENOUGH FOOD
      FOR EVERYONE...."

585 PRINT:PRINT"HORRORS, YOUR XENDNESS"
590 PRINT"THERE'S BEEN A REBELLION!!!!"
600 PRINT"THE PEASANTS ARE REVOLTING"
610 PRINT"(I HAD TO PUT THAT LINE IN)"
620 GOSUB 880
630 IFRND(5)=1THENPRINT"AND YOUR COPS
      COULDN'T STOP THEM":GOTO480
```

```

640 PRINT" BUT YOUR RUTHLESS POLICE HAVE"
650 PRINT" PUT A STOP TO ALL THAT NONSENSE"
660 GOSUB 880
670 RETURN
680 NEXT Y
690 REM *****
700 REM INITIALISATION
720 CLS
730 P=PEEK(RND(32767)):H=P
740 U=700+RND(550)
750 L=70+RND(50)
760 RETURN
770 REM *****
780 REM BANKRUPTCY
790 PRINT:PRINT"WELL, XEND, THATS A FINE"
800 PRINT"MESS YOU'VE GOT YOURSELF, AND
810 PRINT"OUR LITTLE PLANET INTO."
820 GOSUB 880
830 PRINT:PRINT"THE TREASURY IS BANKRUPT!!"
840 GOSUB 880
850 PRINT:PRINT"AND GUESS WHO BLEW IT?????????"
860 GOTO 490
870 REM *****
880 REM DELAY
890 FOR J=1 TO 1500 :NEXT J
900 PRINT
910 RETURN

```

## RURAL PURSUITS

If you've survived being MISTRESS OF XENOPHOBIA, you may be interested in something a little more earthy -- running a farm. In RURAL PURSUITS, based loosely on a program written by Stephen Glen, also of Glasgow, you have a very difficult task, as you battle with an extremely touchy work-crew to bring home the bacon (or at least the wheat and barley).

Again the program is largely self-prompting, and working with it will teach you the best strategy for gaining a maximum return from your farm (and surviving for the requisite 10 years to win the game). However, a few hints will not go astray. Although you can save money by paying your workers extremely badly, you'll find they'll desert you in droves in the following year, and the return you get from your land is dictated, to some degree, by the number of people you have working on the land.

Another factor to keep in mind is that the three crops which you can plant have quite different returns, so it is in your interest to work out which is the most profitable crop, and concentrate on that.

Do not be dismayed if you foul the process up the first few times you run the program. You'll have to develop some real skills to keep your farm solvent, and it may take you several bad years to get the hang of it.

```

10 REM RURAL PURSUITS
20 YR=1
40 MO=RND(1000)+7000
50 LA=RND(1000)+100
60 AC=RND(200)+300
70 CS=RND(5)+8
80 BA=0:WH=0:CO=0
90 GOSUB 170
100 GOSUB 290
110 IF YR=10 THEN 750

```

```

120 IF MD<1 THEN 790
130 IF LA<1 THEN 820
140 YR=YR+1:AC=AC+INT(AC/RND(100))
150 CS=CS+INT(12.5*CS/100)
160 GOTO 80
170 REM UPDATE
180 CLS
190 PRINT"YOU HAVE $"MO"IN YEAR"YR
200 GOSUB 880
210 PRINT"YOU ARE EMPLOYING"LA
220 PRINT"LABORERS, WORKING FOR"
230 PRINT"YOU ON"AC"ACRES"
240 GOSUB 880:FOR V=1 TO 1000:NEXT V:
      PRINT "PRESS ANY NUMBER"
245 IF VAL(INKEY$)=0 THEN 245
250 PRINT"CROPS:-"TAB(7);CO"CORN"
260 PRINT ,BA"BARLEY"
270 PRINT ,WH"WHEAT"
280 RETURN
290 PRINT:PRINT"IT WILL BE $"CS"IN"
300 PRINT"GENERAL COSTS TO WORK EACH"
310 PRINT"ACRE...AND SO THE MAXIMUM"
320 PRINT"NUMBER OF ACRES YOU CAN"
330 PRINT"WORK THIS YEAR IS";
340 MAX=INT(MD/CS):IF MAX>AC THEN MAX=AC
350 PRINT MAX
360 PRINT"HOW MUCH LAND DO YOU WANT TO":
      PRINT"HARVEST";
370 INPUT L
380 IF L>MAX THEN 370
390 MD=MD-L*CS
400 GOSUB 170
410 PRINT:PRINT"HOW MUCH WILL YOU PAY EACH ":
      PRINT"WORKER";
420 INPUT W
430 IF W*LA>MD THEN 420
440 MD=MD-LA*W
450 GOSUB 170
460 P=10
470 PRINT"WHAT PROPORTION (OUT OF TEN) DO"
480 PRINT"YOU WISH TO CONCENTRATE ON CORN";
490 INPUT CP
500 IF CP>P THEN 490
510 P=P-CP

```

```

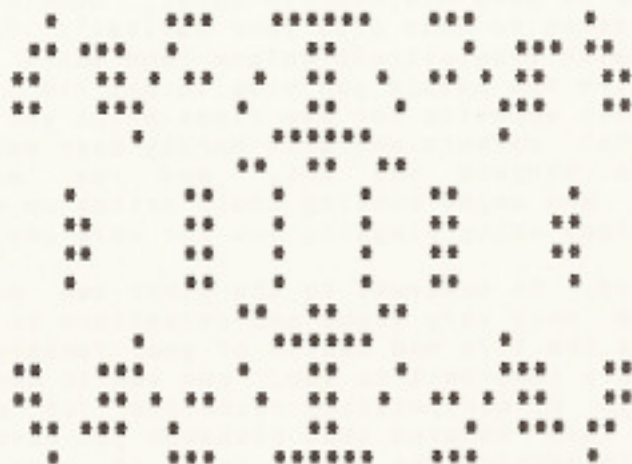
520 PRINT:PRINT"OF THE REMAINING"P"OUT OF TEN,"
530 PRINT"HOW MUCH WHEAT DO YOU WANT TO PLANT"
540 INPUT WP
550 IF WP>P THEN 540
560 P=P-WP
570 GOSUB 870
580 PRINT"STAND BY FOR A YEAR...."
590 FOR Z=1 TO 2000:NEXT Z
600 BA=INT(P*LA*W*3/100000)
610 CO=INT(CP*LA*W*2.7/17000)
620 WH=INT(WP*LA*W*1.4/9300)
630 T=BA+CO+WH
640 GOSUB 170
650 PRINT:PRINT T"TONS WERE HARVESTED"
660 RT=INT((.5+8.7*BA+5.94*CO+2.2*WH)*(CS-CX+1))
670 IF BA=0 AND CO=0 AND WH=0 THEN RT=0
680 FOR Z=1 TO 2000:NEXT Z
690 PRINT:PRINT"AND YOUR TOTAL RETURN"
700 PRINT"WAS $"RT
710 MD=MD+RT
720 LA=INT(LA-LA/(W+.01))
730 FOR Z=1 TO 2000:NEXT Z
740 RETURN
750 GOSUB 870
760 PRINT "YOU HAVE SURVIVED FOR 10 YEARS"
770 PRINT"CONGRATULATIONS!"
775 IF MD>1E+12 THEN GOSUB 900
780 PRINT "AND GAINED"MO"MONEY":END
790 GOSUB 870
800 PRINT "YOU HAVE GONE BROKE!"
810 END
820 GOSUB 870
830 PRINT "YOU HAVE NO WORKERS, AND"
840 PRINT "HAVE BEEN FORCED TO SELL"
850 PRINT , "YOUR FARM"
860 END
870 CLS
880 PRINT:PRINT
890 RETURN
900 FOR TU=1 TO 33:READ FR,DU:SOUND FR,DU:
      NEXT TU:RETURN

```

```

910 DATA 16,3,18,2,20,3,20,1,20,2,18,1,20,2,21,5,20,3
920 DATA 20,2,18,3,18,1,18,2,16,2,18,2,20,3,16,2
930 DATA 16,3,18,2,20,3,20,1,20,2,18,1,20,3,21,3,
    23,3,25,5
940 DATA 25,4,23,3,21,2,20,3,18,2,16,5

```



# CHAIRMAN OF THE BOARD

Finally in this section we have our major simulation, CHAIRMAN OF THE BOARD.

Again the program is largely self-prompting. The aim of the game is to keep your factory running until you manage to make \$10,000 (in total, combining the value of stock in hand plus your capital). You have to deal with recalcitrant unions (who won't always let you fire the people you wish to get rid of, and have a great appetite for pay rises which you cannot deny), with workers who will hardly ever meet the production targets you set, and raw material suppliers who enjoy putting their prices up as much as the unions enjoy slugging you for more pay.

You'll find, in contrast to the other two programs which are only very vague approximations to 'real life' that the life and health of your factory will become very important to you, and you'll certainly learn ways of manipulating resources for maximum return. Keep in mind that although you have great freedom in setting the retail price of your products, each rise in price will increase consumer resistance to purchasing those products.

```

10 REM CHAIRMAN OF THE BOARD
20 GOSUB 1670:REM INITIALISE
30 WE=WE+1
40 GOSUB 930:REM REPORT
50 GOSUB 1300:REM STAFF
60 GOSUB 930:REM REPORT
70 GOSUB 1130:REM PRODUCTION
80 GOSUB 930:REM REPORT
90 GOSUB 730:REM SALES
100 GOSUB 140:REM PROBLEMS
110 CA=CA-WA*WD-RC
120 GOTO 30
130 REM *****
140 REM PROBLEMS

```

```

150 CLS
160 IF RND(0)<.45 THEN 260
170 A=RND(7)
180 PRINT:PRINT:PRINT
190 PRINT "THE UNIONS ARE DEMANDING A"
200 PRINT "PAY RISE OF"A"%"
210 WA=INT(100*(WA+WA*A/100))/100
220 GOSUB 1840
230 PRINT:PRINT "PAY PER EMPLOYEE IS NOW $"WA
240 GOSUB 1840
250 CLS
260 IF RND(0)<.81 THEN 410
270 PRINT:PRINT:PRINT
280 PRINT "A FIRE IN YOUR WAREHOUSE HAS"
290 PRINT "DESTROYED SOME STOCK. PLEASE"
300 PRINT "STAND BY FOR A REPORT ON"
310 PRINT "THE DAMAGE CAUSED..."
320 GOSUB 1840
330 A=RND(ST/2)
340 ST=ST-A
350 PRINT:PRINT "THERE WERE"A;A$
360 PRINT "DESTROYED. THEY WERE "
370 PRINT "WORTH $"A*SP"RETAIL"
380 GOSUB 1840
390 PRINT "STOCK IN HAND IS"
400 PRINT "NOW"ST;A$
410 IF RND(0)>.3 THEN 560
420 CLS
430 PRINT:PRINT
440 PRINT "YOUR MAIN SUPPLIER HAS ANNOUNCED"
450 PRINT "A DRAMATIC PRICE RISE..."
460 GOSUB 1840
470 A=RND(100*CO/7)/100
480 IF A<1 THEN 470
490 PRINT:PRINT"THE COST OF MAKING "A$
500 PRINT "HAS RISEN BY $"A"EACH"
510 GOSUB 1840
520 CO = CO + A
530 PRINT:PRINT "IT NOW COSTS $"CO
540 PRINT "TO MAKE EACH ONE..."
550 GOSUB 1840
560 IF RND(0) < .65 AND MA < SP THEN RETURN
570 CLS
580 PRINT:PRINT

```

```

590 PRINT "YOU HAVE A CHANCE TO RAISE"
600 PRINT TAB(4);"YOUR PRICE. YOUR"
610 PRINT A$;" NOW SELL FOR $"SP
620 GOSUB 1840
630 PRINT
640 PRINT "WHAT PERCENTAGE INCREASE WOULD"
650 INPUT "YOU LIKE TO IMPOSE";A
660 RE=RE+A
670 SP=INT(100*(SP+A*SP/100))/100
680 GOSUB 1840
690 PRINT:PRINT "THE "A$" NOW SELL FOR $"SP
700 GOSUB 1840
710 RETURN
720 REM *****
730 REM SALES
740 PRINT:PRINT "YOUR TOTAL STOCK OF"
750 PRINT A$ " IS"ST
760 GOSUB 1840
770 PRINT:PRINT "PLEASE STAND BY FOR A"
780 PRINT "SALES REPORT..."
790 A=INT(RND(0)*ST/(RE/1000))+1
800 IF A>ST THEN 790
810 CLS
820 PRINT:PRINT
830 PRINT "THE TOTAL NUMBER OF "A$
840 PRINT "SOLD IS"A
850 ST=ST-A
860 ZA=A*SP
870 PRINT:PRINT "THE INCOME FROM THAT"
880 PRINT "SALE WAS $"ZA
890 CA=INT(A*SP*100)/100+CA
900 GOSUB 1840
910 RETURN
920 REM *****
930 REM REPORT TO THE CHAIRMAN
940 CLS
950 IF CA+ST<1 THEN 1510:REM BANKRUPTCY
960 IF CA+ST>9999 THEN PRINT "YOU'VE MADE $10000 AND
965 IFCA+ST>9999THEN PRINT"CAN NOW RETIRE...";
GOSUB1840:GOTO 1590
970 PRINT:PRINT "SHOP FLOOR REPORT, SIR,"
980 PRINT TAB(6);"FOR WEEK"WE
990 PRINT:PRINT "CAPITAL IN HAND IS $"
INT(CA*100)/100

```

```

1000 PRINT "RUNNING COSTS ARE $"RC"A WEEK"
1010 PRINT:PRINT "YOUR STORES HOLD"ST;A$
1020 PRINT TAB(6);"WORTH $"INT(ST*SP*100)/100
1030 PRINT:PRINT "THEY SELL FOR $"SP" EACH"
1040 PRINT "AND COST $"CO"EACH TO MAKE"
1045 IF INKEY$("<") " " THEN 1045
1050 PRINT:PRINT "YOUR WORKFORCE IS NOW"
1060 PRINT WO"STRONG, AND YOU ARE"
1070 PRINT "PAYING THEM $"WA"EACH"
1080 PRINT "THE WAGES BILL THIS WEEK IS $"WA*WO"
1090 PRINT:PRINT "EACH PERSON CAN MAKE"PR
1100 PRINT A$" A WEEK, A TOTAL"
1110 PRINT "OUTPUT OF"PR*WO
1120 RETURN
1130 INPUT "HOW MANY DO YOU WISH TO PRODUCE";MA
1140 IF MA=0 THEN RETURN
1150 PRINT
1160 IF MA*CO>CA THEN PRINT"YOU DO NOT HAVE
                                ENOUGH MONEY":GOTO1130
1170 IF MA>PR*WO THEN PRINT "YOU DO NOT HAVE
                                ENOUGH PEOPLE"
117B IF MA>PR*WO THEN PRINT TAB(6);"THAT MANY":
                                GOTO 1130
1180 PRINT "YES SIR...THE TARGET FOR WEEK"WE
1190 PRINT "IS"MA;A$
1200 MA=INT(MA-RND(0)*MA/5)
1210 GOSUB 1840
1220 PRINT:PRINT "THE NUMBER OF "A$
1230 PRINT "ACTUALLY PRODUCED IN WEEK"WE
1240 PRINT "WAS"MA"... "
1250 ST=ST+MA
1260 CA=CA-CO*MA
1270 GOSUB 1840
1280 RETURN
1290 REM *****
1300 REM STAFF
1310 PRINT "HOW MANY PEOPLE DO YOU"
1320 INPUT "WISH TO HIRE";A
1330 WO=WO+A
1340 PRINT:PRINT "THE TOTAL WORKFORCE"
1350 PRINT "IS NOW"WO"STRONG"
1360 GOSUB 1840
1370 IF A>0 THEN RETURN

```

```

1380 GOSUB 930
1390 PRINT "HOW MANY PEOPLE DO YOU"
1400 INPUT "WISH TO FIRE";A
1410 IF A=0 THEN 1480
1420 IF A>WO THEN 1390
1430 A=RND(A)
1440 GOSUB 1840
1450 PRINT:PRINT "THE UNIONS WILL ALLOW"
1460 PRINT "YOU TO GET RID OF"A
1470 WO=WO-A
1480 GOSUB 1840
1490 RETURN
1500 REM *****
1510 REM BOTTOM LINE!
1520 PRINT:PRINT "YOU'RE BANKRUPT!!":SOUND 5,8
1530 GOSUB 1840
1540 PRINT:PRINT "OH THE SHAME OF IT!!":SOUND 3,8
1550 GOSUB 1840:SOUND 1,8
1560 PRINT:PRINT "STILL, YOU KEPT THE BUSINESS"
1570 PRINT "GOING FOR"WE"WEEKS"
1580 GOSUB 1840
1590 PRINT "ENTER 'Y' FOR ANOTHER STINT"
1600 PRINT "AS CHAIRMAN OF THE BOARD...."
1610 PRINT "(OR 'N' IF YOU WANT TO QUIT)"
1620 A$=INKEY$
1630 IF A$("<")"Y" AND A$("<")"N" THEN 1620
1640 IF A$="Y" THEN RUN
1650 END
1660 REM *****
1670 REM INITIALISE
1690 FOR Z=1 TO RND(8)
1700 READ A$
1710 NEXT Z
1720 CA=500+RND(500)
1730 ST=100+RND(500)
1740 SP=10+RND(5)
1750 CO=7+RND(5)
1760 IF CO>SP THEN 1750
1770 WO=7+RND(10)
1780 WA=12+RND(SP)
1790 PR=5+RND(6)
1800 RC=100+RND(20)
1810 WE=0
1820 RE=1:REM RE IS SALES RESISTANCE FACTOR

```

```
1830 RETURN
1840 REM DELAY SUBROUTINE
1850 FOR Z=1 TO 3000:NEXT Z:RETURN
1860 DATA "EPONYMS", "BICYCLES", "HARMONICAS"
1870 DATA "KAZOOS", "LECTERNS", "MOLESKINS"
1880 DATA "CARPETBAGS", "PITH HELMETS", "SKYHOOKS"
1890 DATA "BARBELLS"
```



# Dice Games



# COINAGE



# Dice Games

Dice games -- for high stakes, low stakes or none -- have proved popular throughout history. Whether the dice were made from stone or wood, or carved from the bones of a beast killed for food, whether they had four important sides or six, dice have proved a diversion for countless men throughout time.

The dice, physical means of demonstrating the waywardness of chance, have also found employment in fortelling the future, and like all oracles, probably bear some responsibility for changing the tide of history.

At Troy the Greeks played with dice, as did Mark Antony at Alexandria. Italy and France led medieval Europe in its love for dice, and devotion to the six-sided cubes has continued up to the present day. In this section of the book, we'll be investigating ways of using the random number generator of your computer to take the place of physical dice. However, you'll discover that the dice are no less compelling because they exist only within the software of your computer rather than in a more gross physical form.

Playing against the computer, instead of against another human being, changes the 'feel' of dice games somewhat. Not having to pay up when you lose is one of the real advantages!

To start our dice games, here's a routine you might like to incorporate into some of your own dice programs:

```
2000 REM DICE THROW
2010 CLS:COLOR,1
2020 PRINT TAB(12);"DIE"DN
2030 D=RND(6)
2040 FOR I=28810 TO 28822:POKE I,140:NEXT I
2050 FOR I=28810 TO 29066 STEP 32:POKE I,138:NEXT I
```

```
2060 FOR I=29066 TO 29078:POKE I,131:NEXT I
2070 FOR I=28822 TO 29078 STEP 32:POKE I,133:NEXT I
2080 POKE 28810,142:POKE 29066,139:POKE 29078,135:
      POKE 28822,141
2090 IF D=1 THEN 2150
2100 IF D=2 THEN 2160
2110 IF D=3 THEN 2170
2120 IF D=4 THEN 2180
2130 IF D=5 THEN 2190
2140 IF D=6 THEN 2200
2150 POKE 28944,143:GOTO 2220
2160 POKE 28876,143:POKE 29012,143:GOTO 2220
2170 POKE 28876,143:POKE 28944,143:POKE 29012,143:
      GOTO 2220
2180 POKE 28876,143:POKE 29004,143:POKE 28884,143:
      POKE 29012,143
2185 GOTO 2220
2190 POKE 28876,143:POKE 29004,143:POKE 28884,143:
      POKE 29012,143
2195 POKE 28944,143:GOTO 2220
2200 POKE 28876,143:POKE 29004,143:POKE 28884,143:
      POKE 29012,143
2210 POKE 28940,143:POKE 28948,143
2220 FOR I=1 TO 2500:NEXT I:COLOR,0:RETURN
```

In this program, the variable D represents the die thrown, and DN represents the dice number.

An enormous range of games is now played with dice and they sport an intriguing variety of titles. Here are some names of common dice programs:

- Going to Boston
- Craps
- Fifty
- Crag
- Baseball
- Basketball
- Poker Dice
- Liar Dice

The dice games we'll be playing in this book are SNAKES EYES, ONE-AND-TWENTY, SEVEN/ELEVEN OVER/N'UNDER, NO SWEAT and CHEMIN DE COMPUTER.

# CHEMIN DE COMPUTER

CHEMIN DE COMPUTER is based, as I'm sure you've realised, on Chemin de Fer. In this game, you and your computer take it in turns to roll five dice, adding the pips up as you go. You are aiming to get a higher total than the computer.

However, this game is not just a simple 'add the pips' one. Any die which falls showing a five or a two must be thrown again, and your total is just the final digit of the answer (that is, a total of 27 is counted as 7, and a total of 13 is counted as 3).

There are three special totals - 7 (Natural), 8 (Petite) and 9 (Grande). The names are only bestowed on the roll when the relevant total is achieved on the first throw (i.e. the dice which came up with 5 or 2 have not been, and are not in this case, rethrown). You (and the computer) always stand on a 7, 8 or 9 thrown with the first toss of the dice.

As in many other programs in this book, there is a delay loop routine at the end of the listing which is called a number of times throughout a game to improve the speed with which the game advances, and to give you (in many cases) a chance to read on the screen what is going on before the program races on, clearing the screen, and leaving you quite unsure of exactly what has just happened. In this program, there are two loops, one which is longer than the other, and which also prints a couple of blank lines before returning to the main program.

```
10 REM CHEMIN
30 CLS
40 GAME=0
50 B1=0:P1=0
60 GOTO 330
70 D=0:C=0
```

```
80 FOR G=1 TO 5:A=RND(6)
90 IF A=2 OR A=5 THEN C=C+1
100 PRINT A;:SOUND 31,1
110 GOSUB 930
120 IF A=2 OR A=5 THEN A=0
130 D=D+A:NEXT G
140 PRINT:PRINT D;
150 IF D>9 THEN D=D-10:PRINT:PRINT D:GOTO 150
160 GOSUB 900
170 PRINT:PRINT "TOTAL ON THE FIRST ROLL IS"D
180 IF D=9 THEN PRINT "AND THATS LA GRANDE...":RETURN
190 IF D=8 THEN PRINT "AND THATS LA PETITE...":RETURN
200 IF D=7 THEN PRINT "AND THATS A NATURAL...":RETURN
210 IF C=0 THEN RETURN
220 GOSUB 930
230 PRINT:PRINT C"MUST BE ROLLED AGAIN"
235 FOR Z=1 TO C:SOUND 3,1:NEXT Z
240 FOR A=1 TO C
250 GOSUB 930
260 E=RND(6)
270 PRINT E;:SOUND 29,1
280 IF E=2 OR E=5 THEN E=0
290 D=D+E
300 NEXT
310 IF D>9 THEN D=D-10:PRINT D;:GOTO 310
320 RETURN
330 GOSUB 930:CLS
340 GAME=GAME+1
350 PRINT:PRINT ">>>> THIS IS GAME"GAME" <<<<"
360 PRINT "*****"
370 PRINT " NOW, I'LL ROLL AS BANKER..."
380 PRINT "*****"
390 GOSUB 70
400 GOSUB 900
410 PRINT:PRINT "SO MY FINAL TOTAL IS"D
420 GOSUB 900
430 PRINT "*****"
435 A%=INKEY$:A$=""
440 PRINT " PRESS 'R' TO ROLL YOUR DICE"
445 A%=INKEY$:IF A%<>"R" THEN 445
450 PRINT "*****"
460 J=D
470 GOSUB 930
480 GOSUB 70
```

```

490 PRINT:PRINT "SO YOUR FINAL TOTAL IS"D
500 PRINT:PRINT
510 GOSUB 930
520 PRINT "COMPUTER", "HUMAN"
530 GOSUB 900
540 PRINT J,D
550 GOSUB 930:PRINT
560 PRINT "*****"
570 PRINT TAB(3);
580 IF J=D THEN PRINT "THATS A STANDOFF":GOTO 620
590 IF J>D THEN PRINT "I'M";:B1=Bi+1
600 IF D>J THEN PRINT "YOU'RE";:P1=P1+1
610 PRINT " THE WINNER THAT TIME!"
620 PRINT "*****"
630 PRINT:PRINT
640 GOSUB 900
650 PRINT " NOW THE TOTALS SO FAR IN"
660 PRINT " CHEMIN DE COMPUTER ARE"
670 PRINT B1"FOR ME, AND"P1"FOR YOU..."
680 PRINT:PRINT
690 GOSUB 930
700 IF B1+P1=9 THEN 750
710 IF B1>P1 THEN PRINT " LOOKS LIKE I'M IN FRONT!"
720 IF P1>B1 THEN PRINT " AND YOU SEEM TO
                         HAVE THE EDGE!"

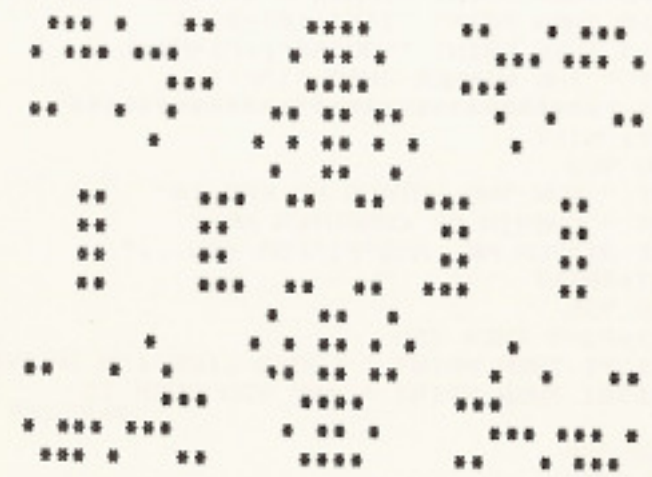
730 GOSUB 900
740 GOTO 330
750 REM END OF GAME
760 PRINT:PRINT
770 PRINT "WELL, OLD BUDDY, WE SEEM"
780 PRINT "TO HAVE COME TO THE END"
790 PRINT "OF THE GAME...WITH A TOTAL"
800 PRINT "OF NINE SCORING ROUNDS..."
810 GOSUB 900
820 IF P1>B1 THEN PRINT "AND, FOR ONCE, IT IS"
825 IF P1>B1 THEN PRINT "YOU HAS BEATEN ME!":GOTO 840
830 IF B1>P1 THEN PRINT "AND, ONCE AGAIN, THE"
835 IF B1>P1 THEN PRINT "MIGHTY MACHINE PROVES THE"
837 IF B1>P1 THEN PRINT "SUPREME CHAMPION!!"
840 GOSUB 900
850 PRINT "THANKS FOR THE GAME, OLD"
860 PRINT "BUDDY,WE MUST DO IT AGAIN"
870 PRINT "SOME TIME, WHEN YOU FEEL"

```

```

880 PRINT "LUCKY....."
890 END
900 FOR Z=1 TO 1500:NEXT Z
910 PRINT:PRINT
920 RETURN
930 FOR Z=1 TO 750:NEXT Z:RETURN

```



# SEVEN/ELEVEN

SEVEN/ELEVEN is another fine dice game, and one in which your goal changes in each game. Based on CRAPS, you have a 'target number' to reach in each round (known as your 'point' in Craps).

Craps is an American version of the old English dice game Hazard. In SEVEN/ELEVEN, the complex betting of craps has been removed, leaving just the fun of the game. Rather than worry about computing odds and placing 'pass' and 'don't pass' bets, you can concentrate on the serious business of praying for the right roll to come up.

In this game, you are playing against yourself. Your winning and losing rounds are tallied, so you know -- at every point in the game -- just how well (or otherwise) you are doing.

You throw two dice at a time. If you throw a 2, 3 or a 12 with your first roll, you lose that round. Getting a 7, or an 11, on the first roll is the best thing you can do, because you've won that round with that single roll (which is why the game is called SEVEN/ELEVEN).

Any other number rolled as your first roll of a round becomes your 'target number'. You have to try and roll this again, before you roll a 7 or an 11. If you manage to roll your target number again, you win that round. However, rolling a 7 or an 11 before you've rethrown your target total, causes you to lose that round.

```
10 REM SEVEN ELEVEN
20 B# = "IN 7/11 YOU'VE ROLLED"
30 G = 0:W = 0:L = 0:M = 105
40 CLS
50 PRINT:PRINT "THE TALLY SO FAR:"
60 PRINT "WINS:"W"  LOSSES:"L
70 G = G + 1
80 M = M - 5
```

```
90 PRINT "YOU HAVE $"M
100 PRINT:PRINT "THIS IS ROUND NUMBER"G
110 GOSUB 200
120 IF A=7 OR A=11 THEN 300
130 IFA=2 OR A=3 OR A=12 THEN 360
140 P=A
150 CLS:PRINT:PRINT"YOUR TARGET NUMBER IS"P
160 GOSUB 200
170 IF A=P THEN 300
180 IF A=7 OR A=11 THEN 310
190 GOSUB 340:GOTO 150
200 N=0:G#=INKEY#
210 PRINT:PRINT"PRESS 'R' TO ROLL"
220 N=N+W:IF INKEY#="" THEN 220
230 C = RND(6):PRINT"YOU ROLLED "C:SOUND C+12,C
240 GOSUB 340
250 B=RND(6):PRINT"AND"B:SOUND B+24,B
260 A=B+C
270 PRINT:PRINT B#;A
280 GOSUB 340
290 RETURN
300 PRINT"AND YOU'VE WON":W=W+1:M=M+20: GOTO 320
310 PRINT"AND SO YOU LOSE":L=L+1:SOUND 9,4
      :SOUND 6,4:SOUND 4,6
320 GOSUB 340
330 GOTO 40
340 FOR T=1 TO 500:NEXT T
350 RETURN
360 PRINT"SO THATS THE END OF THE ROUND":GOTO 320
```

# NO SWEAT

The name of this game -- NO SWEAT -- comes from one of the the early English names for the game from which this is derived. Known as 'Sweatcloth', the game, when first played in England, used three dice within a wooden 'shoe'. The most common variation of this game is now seen in US gambling casinos where it is called 'Birdcage', because of the equipment which is used.

In Birdcage, three dice are held within a metal cage which can spin about a central axis. Once bets have been laid, the cage is rotated. This is supposed to ensure that the dice are spun properly, as they are not actually touched by a casino operator.

Betting in Birdcage, and in No Sweet, is fast and simple, and despite the fact that you might think the odds are in your favor as a player, you'll quickly learn how easy it is to lose your shirt.

You place a bet of a specified amount (see lines 50 and 60) up to the size of your stake (held by the variable M, for money - see lines 30 and 250) and then chose a number between one and six. The amount of your bet is subtracted from your stake, and then the three dice are rolled. For each one which lands showing your number, you'll have an amount equal to your bet returned to you.

As you can see, this means you have to throw one die with your number showing to break even, and two to actually make some money. The need to throw the same, chosen, number twice is what gives the casino such a good edge in this game.

Line 50 asks you how much you'd like to bet, and the variable A is assigned to your choice. This is compared with the money you actually have (in line 60) to see if you are trying to bet more than you have on hand.

The C loop, from lines 100 to 180 rolls the dice, and compares each roll with your number, reporting to you after each roll. Your winnings (if there are any) are also added in this loop.

After each round of the game, your money (M) is compared with 250 (if you've got more than \$250 you've exceeded the house limits and are therefore excluded from future play) and with zero (those who have no money are not allowed to play further). If however, you do not have more than 250 nor less than zero, you're able to continue playing.

```
10 REM NO SWEAT
30 M=30
40 CLS:GOSUB 240
50 INPUT"HOW MUCH WOULD YOU LIKE TO BET";A
60 IF A>M THEN 50
70 M=M-A:PRINT:PRINT
80 INPUT"WHICH NUMBER ARE YOU BETTING ON";B
90 IF B<1 OR B>6 THEN 80
95 GOSUB 280
100 FOR C=1 TO 3
110 W=0
120 GOSUB 2000:PRINT#25,"A1M-";B
130 PRINT:PRINT"DIE"C"FELL"D
140 IF D=B AND C=1 THEN W=A:PRINT" SO YOU BREAK EVEN"
150 IF D=B AND C>1 THEN W=A:PRINT" SO YOU WIN $*W
160 M=M+W
170 GOSUB 240
175 GOSUB 280:GOSUB 280
180 NEXT C
190 GOSUB 280:GOSUB 280
200 IF M>250 THEN 310
210 IF M<0 THEN 40
220 FOR I=1 TO 8:PRINT"THE GAME IS OVER,'COS YOU'RE BROKE!":NEXT
230 GOTO 230
240 PRINT"*****"
250 PRINT" YOUR STAKE IS NOW $*M
260 PRINT"*****"
270 RETURN
280 FOR P=1 TO 1000
290 NEXT P
300 RETURN
310 FOR J=1 TO 30
```

```

320 PRINT"YOU'VE TOPPED $250!"
330 PRINTTAB(2);"WELL DONE!!"
335 FOR I=1 TO 100:NEXT
340 NEXT J
350 COLOR,1:FORI=1TO500:NEXT:COLOR,0:FORI=1TO500:NEXT:GOTO 350
2000 REM DICE THROW
2010 CLS:COLOR,1
2020 PRINT TAB(14);"DIE"C
2030 D=RND(6)
2040 FOR I=28810 TO 28822:POKE I,140:NEXT I
2050 FOR I=28810 TO 29066 STEP 32:POKE I,138:NEXT I
2060 FOR I=29066 TO 29078:POKE I,131:NEXT I
2070 FOR I=28822 TO 29078 STEP 32:POKE I,133:NEXT I
2080 POKE 28810,142:POKE 29066,139:POKE 29078,135:POKE 28822,141
2090 IF D=1 THEN 2150
2100 IF D=2 THEN 2160
2110 IF D=3 THEN 2170
2120 IF D=4 THEN 2180
2130 IF D=5 THEN 2190
2140 IF D=6 THEN 2200
2150 POKE 28944,143:GOTO 2220
2160 POKE 28876,143:POKE 29012,143:GOTO 2220
2170 POKE 28876,143:POKE 28944,143:POKE 29012,143:GOTO 2220
2180 POKE 28876,143:POKE 29004,143:POKE 28884,143:POKE 29012,143
2185 GOTO 2220
2190 POKE 28876,143:POKE 29004,143:POKE 28884,143:POKE 29012,143
2195 POKE 28944,143:GOTO 2220
2200 POKE 28876,143:POKE 29004,143:POKE 28884,143:POKE 29012,143
2210 POKE 28940,143:POKE 28948,143
2220 FOR I=1 TO 2500:NEXT I:COLOR,0:CLS:RETURN

```

# ONE-AND-TWENTY

Our dice games continue now with ONE-AND-TWENTY which, as you realised the moment you read the title in the introduction, is a dice version of the card game, Pontoon.

The game is simple, but demands a degree of cool thinking, as well as the ability to guess which numbers are going to come up next when the die is rolled. In ONE-AND-TWENTY, you are playing against the computer.

You are always given the first go. You roll the die as many times as you like, aiming to get a total as close as possible to, but not exceeding, 21. You can stop adding to the total whenever you like.

If you exceed 21 (that is, you 'bust') then the computer wins that round automatically. A game consists of five rounds, and the winner of the most rounds, naturally enough, wins the game.

The lines from 50 to 170 control your dice-rolling, and the computer only exits this cycle when you decide to stand. The whole of the computer's game-playing logic is held within line 180 which determines whether it is worth risking a bust to try and exceed your total. If it decides to roll, the lines from 190 to 250 control this.

Once the computer decides it has had enough of that, it uses lines 260, 280 and 290 to determine who has won. If both your scores are the same, or both of you are over 21, then the round is counted as a dead heat (line 320 tells you this), and neither CS nor HS is incremented.

As in Chemin de Computer, where the aim was to get the best out of nine scoring games, in One-and-Twenty, the intention is to score wins in the majority of games out of five.

```

10 REM ONE-AND-TWENTY
20 HS=0:CS=0
30 H=0:C=0:N=0
40 CLS
50 PRINT "PRESS 'R' TO ROLL, 'S' TO STAND"
60 D#=INKEY#:D#=INKEY#
70 A#=INKEY#
80 IF A#<>"R" AND A#<>"S" THEN 70
90 IF A#="S" THEN CLS:PRINT "YOUR TOTAL IS"H:GOTO 180
110 GOSUB 430
120 R=RND(6)
130 PRINT "YOU ROLLED A"R:SOUND R+12,R
140 H=H+R
150 PRINT "SO YOUR TOTAL IS"H
160 GOSUB 430
170 PRINT:GOTO 50
180 IF C>H AND C<22 OR C>21 OR H>21 OR H=21
      AND C=21 THEN 260
190 R=RND(6)
200 GOSUB 430
210 PRINT:PRINT "I ROLLED A"R
220 C=C+R
230 PRINT "SO MY TOTAL IS"C
240 GOSUB 430
250 GOTO 180
260 IF H=C OR H>21 AND C>21 THEN 320
270 GOSUB 510
280 IF (C>H OR H>21) AND C<22 THEN PRINT ,"I":CS=CS+
290 IF (C<H OR C>21) AND H<22 THEN PRINT ,
      "YOU":HS=HS+1
300 PRINT " WIN!!"
310 GOTO 330
320 PRINT "THAT ROUND IS A DEAD HEAT....NO ", "SCORE"
330 GOSUB 430
340 GOSUB 510
350 PRINT "AFTER THAT ROUND, THE GAME SCORE IS"
360 GOSUB 430
370 PRINT TAB(7);"YOU:"HS", AND ME"CS
380 GOSUB 510
390 IF CS+HS=5 THEN 450
400 PRINT:PRINT "STAND BY FOR THE NEXT ROUND..."
410 GOSUB 430:GOSUB 430
420 CLS:GOTO 30
430 FOR O=1 TO 500:NEXT O

```

```

440 RETURN
450 PRINT:PRINT "THAT'S THE END OF THE GAME"
460 PRINT:PRINT "FINAL SCORES:"
470 PRINT ,"YOU:"HS
480 PRINT ," ME:"CS:PRINT
490 IF CS>HS THEN PRINT "AND I DEFEATED YOU, HUMANOID"
495 IF CS>HS THEN GOTO 530
500 PRINT "AND REMARKABLY, A MERE HUMAN"
505 PRINT "BEAT THE MACHINE!!":SOUND 5,3
507 GOSUB 430:GOSUB 430:GOSUB 430:GOSUB 430:
      GOSUB 430:END
510 PRINT:PRINT "*****"
      :PRINT
520 RETURN
530 SOUND 28,3:SOUND 28,2:SOUND 25,4:SOUND 30,2
      :SOUND 28,5
540 SOUND 25,6:END

```

```

* * ***** * * * * * * * * * * * *
*   ***** *   **   **   ***** *
* ** **   *****   *****   **
*   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *
* *****   *****   **   *****   *****
* *****   *****   **   *****   *****
*   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *
* ** **   *****   *****   **
* *****   *****   *****   *****
* *****   *****   *****   *****
*   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *

```



# SNAKES EYES

SNAKES EYES demands some more cool thinking under pressure. You and your computer take it in turns to throw a pair of dice. You add the total of the pipe, and in turn add this to your score.

You can roll the dice as many times as you like, but if you roll a seven, you automatically lose. Therefore, as you can see, the program demands you make decisions based on whether you should be careful and perhaps lose the round by not rolling a high enough score, or whether you should be greedy and go for the absolute maximum score and risk a seven.

The Q loop is covered 20 times, gradually slowing down as it is traversed (by the inner loop, in line 430), until finally -- with line 420 -- the results of the dice rolls are printed. The variable Z is set equal to the tally of the two rolls (in line 480) before the computer goes to line 700 for the delay and to print a line across the screen, and then returns to the start of the program to line 180, the one after the one which sent it to the 'roll the dice' routine. Here the computer checks that the total is not 7 (line 180) and if it is not, then adds the result of the latest roll to your tally.

If, however, you signal through INKEY\$ that you wish to quit, the action goes to the routine from line 220. After printing STAND BY, the computer goes to the delay loop, and line print routine, and then uses the same routine as the human did for rolling the dice, and -- on returning from the subroutine -- checking that the total was not 7, and if not, adding the new total to the score. The computer has a very simple means of deciding if it will roll again...if it has less than the human, it takes the risk.

You'll find that it is relatively easy to program games like this on your computer. Once you've worked out the 'mechanical routines' which do such things as roll the dice, and increment the score, for the human player, it is not very difficult to work out a routine to enable the computer to use the same mechanical routines. Most dice games do not demand much 'intelligence' and their strategy can often be reduced to a couple of IF/THEN statements. Read books on dice games -- such as the excellent DICE GAMES OLD AND NEW by William E Tredd (The Oleander Press, New York, 1981) -- to get ideas for games to turn into programs, and for simple ideas on how to play the game as well as possible. It is these ideas which you should find relatively easy to turn into simple 'intelligent algorithms' to enable your computer to play reasonably well against you.

And, if as in this case and in ONE-AND-TWENTY, you stipulate that the human must go first, the computer knows exactly what target it is aiming at, and therefore starts a round with a considerable advantage, which helps overcome the machine's inherent stupidity.

```
10 REM SNAKES EYES
20 CLS
30 H=0:CZ=0
40 HS=0:CS=0
50 PRINT "PLEASE STAND BY..."
60 GOSUB 700
70 CLS:PRINT:PRINT
80 PRINT "YOUR TOTAL IS"HS:PRINT
90 GOSUB 710
100 PRINT "PRESS 'R' TO ROLL,"
110 PRINT "'Q' TO QUIT"
120 D#=INKEY$:O#=INKEY$
130 W#=INKEY$
150 IF W#<>"R" AND W#<>"Q" THEN 130
160 IF W#="Q" THEN 220
170 GOSUB 370
180 IF Z=7 THEN 350
190 HS=HS+Z
200 PRINT:PRINT "YOUR TOTAL IS"HS
210 GOTO 60
```

```

220 PRINT:PRINT:PRINT "STAND BY"
230 GOSUB 700
240 GOSUB 370
250 IF Z=7 THEN 330
260 CS=CS+Z
270 PRINT:PRINT "MY TOTAL IS"CS
280 PRINT "YOUR TOTAL IS"HS
290 IF CS<HS THEN 220
300 IF CS=HS THEN PRINT "IT'S A DEAD HEAT!"
310 IF CS>HS THEN 350
320 GOTO 510
330 PRINT:PRINT "YOU WIN!!":H=H+1
340 GOTO 510
350 PRINT:PRINT "I WIN!!":CZ=CZ+1
360 GOTO 510
370 REM ROLL DICE
380 CLS
390 FOR Q=1 TO 20
400 X=RND(6)
410 Y=RND(6)
420 PRINT:PRINT "DIE ONE:"X"    DIE TWO:"Y
430 FOR P=1 TO 3*Q:NEXT P
440 NEXT Q
450 CLS
460 GOSUB 710
470 PRINT "DIE ONE:"X"    DIE TWO:"Y
480 Z=X+Y
490 GOSUB 700
500 RETURN
510 PRINT:PRINT "THE SCORE IS:"
520 PRINT ",YOU:"H
530 PRINT ", ME:"CZ
540 IF CZ+H=9 THEN 580
550 GOSUB 710
560 IF H>CZ THEN PRINT "YOU ARE LEADING!"
570 IF CZ>H THEN PRINT "AND I'M IN THE LEAD..."
580 GOSUB 700
590 CLS
600 IF CZ+H=9 THEN 620
610 GOTO 40
620 PRINT:PRINT:PRINT
630 PRINT "WELL, BUDDY, THAT'S THE END"
640 PRINT "    OF THE GAME..."
650 PRINT:PRINT:PRINT

```

```

660 PRINT "YOUR FINAL SCORE WAS"H
670 PRINT "AND MINE WAS"CZ:PRINT
680 IF H<CZ THEN PRINT "I'M THE WINNER!!":GOTO 720
690 PRINT "YOU'RE THE WINNER!!":END
700 FOR P=1 TO 1000:NEXT P:RETURN
710 PRINT "-----":RETURN
720 GOSUB 700:GOSUB 700
730 CLS:A=0:L=2:Z=3
740 POKE 28672+A,127+(Z*16)
750 IF L=2 AND (A+1)/32=INT((A+1)/32) THEN L=4
760 IF L=2 AND PEEK(28673+A)<>32 THEN L=4
770 IF L=4 AND PEEK(28672+A+32)<>32 THEN L=1
780 IF L=1 AND PEEK(28671+A)<>32 THEN L=3
790 IF L=3 AND PEEK(28672+A-32)<>32 THEN L=2
800 IF L=2 THEN A=A+1
810 IF L=1 THEN A=A-1
820 IF L=3 THEN A=A-32
830 IF L=4 THEN A=A+32
840 IF A>511 THEN A=511:L=1
850 IF L=2 AND PEEK(28673+A)<>32 THEN L=4
860 IF L=4 AND PEEK(28672+A+32)<>32 THEN L=1
870 IF L=1 AND PEEK(28671+A)<>32 THEN L=3
880 IF L=3 AND PEEK(28672+A-32)<>32 THEN L=2
890 C=RND(2)-1:IF C=1 THEN X=X+1ELSE X=X-1
900 IF X<5 THEN COLOR,0
910 IF X>10 THEN COLOR,1
920 Z=RND(8)
930 GOTO 740

```

# UNDER 'N' OVER

UNDER'N'OVER is a computer adaptation of the dice game usually known as 'Under and Over Seven'. In this game, you bet on the likelihood of the total of pair of dice landing so that the total is:

- less than seven;
- exactly equal to seven; or
- greater than seven

Again, this is a game which -- at first sight -- seems to offer pretty good odds to the player. However, if you played the game forever, with a pair of perfect dice, you'd find your losses would outweigh your wins by nearly 17%.

Lines 70 to 120 ask you to place your bet, giving you the key for entering it (press A to bet under seven, B to bet on seven and C to bet on a total greater than seven). Lines 130 and 140 read the keyboard using INKEY\$, rejecting (line 140) any input which is not A, B or C.

Having received a valid choice, the odds are printed up by lines 150 to 180 and then line 200 asks you to enter the amount of your bet. Of course, you can't bet more than you have (not in this game, anyway) so line 220 checks your bet (A) against your money (M).

The next section rolls the dice, assigning random values between one and six to variables B and C, and adding them to produce total D in line 280. You are told of the total by line 300.

## THE OUTCOME

This is determined by the routine from lines 310 to 350 where a loss is first assumed (line 310, with variable W, for 'win', assigned to the negative of

your bet). This is changed -- if needed -- into the correct amount for a win if one has, in fact, taken place. You can see you get paid four times your bet for correctly specifying the dice will land with a total of seven (line 320) and even money for either over or under seven (lines 330 and 340).

Lines 370 and 380 tell you about your win (or loss) and then lines 420 checks that you still have some money in hand. If you have, you are offered a new round of the game. If not, you are dismissed with a certain severity.

```
10 REM UNDER 'N' OVER
30 CLS
40 M=30
50 GOSUB 450
60 GOSUB 470
70 PRINT "OK, PUNTER, 'TIS TIME TO"
80 PRINT "PLACE YOUR BET....."
90 GOSUB 470
100 PRINT "ENTER 'A' TO BET UNDER 7"
110 PRINT " 'B' TO BET ON 7, OR"
120 PRINT " 'C' TO BET OVER 7"
*130 A$=INKEY$
140 IF A$<>"A" AND A$<>"B" AND A$<>"C" THEN 130
150 PRINT:PRINT "THE ODDS ARE:"
160 PRINT " A - PAYS EVEN"
170 PRINT " B - PAYS 4 TO 1"
180 PRINT " C - PAYS EVEN"
190 GOSUB 470
200 INPUT "HOW MUCH WOULD YOU LIKE TO BET";A
210 GOSUB 470
220 IF A>M THEN PRINT "YOU HAVEN'T GOT THAT MUCH!":
PRINT:GOTO 200

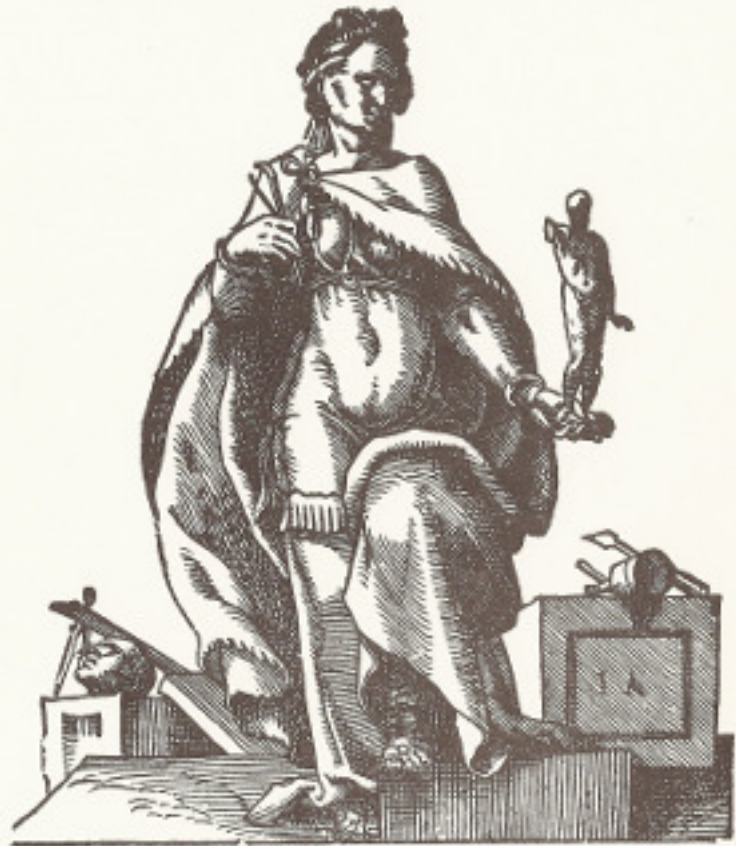
230 B=RND(6)
240 PRINT TAB(7);"DIE ONE CAME UP"B
250 GOSUB 470
260 C=RND(6)
270 PRINT TAB(7);"DIE TWO CAME UP"C
280 D=C+B
290 GOSUB 470
300 PRINT "SO THE TOTAL IS"D
```

```

310 W=-A
320 IF D=7 AND A$="B" THEN W=4*A
330 IF D<7 AND A$="A" THEN W=A
340 IF D>7 AND A$="C" THEN W=A
350 M=M+W
360 GOSUB 470
370 IF W>0 THEN PRINT "YOU'VE JUST WON $"W:GOSUB 530
380 IF W<0 THEN PRINT "AND SO YOU LOSE $"(W*-1)
390 GOSUB 470
400 GOSUB 450
410 GOSUB 470
420 IF M<1 THEN 500
430 CLS
440 GOTO 50
450 PRINT:PRINT "YOU NOW HAVE $"M
460 RETURN
470 FOR P=1 TO 1000:NEXT P
480 PRINT:PRINT
490 RETURN
500 PRINT "YOU'RE FLAT BROKE, 'BUDDY'":SOUND 3,9
510 PRINT "SO I GOTTA CLOSE THE CASINO":SOUND 2,7
520 PRINT "THERE'S NO ROOM FOR DEADBEATS IN MY
      JOINT":SOUND 1,5
525 END
530 SOUND 21,4:SOUND 16,2:SOUND 16,1:SOUND 18,4:
      SOUND 16,4
540 SOUND 0,1:SOUND 20,4:SOUND 21,4
*550 FOR I=0 TO 31:PRINT@I,CHR$(143):
      PRINT@I+448,CHR$(143):NEXT
560 RETURN

```

# Artificial Intelligence



Artificial  
Intelligence



# Artificial Intelligence

Artificial intelligence is a goal which has not yet been achieved. Certainly, programs which enable computers to exhibit behavior which could conceivably be classed as intelligent have been written, but the 'intelligence' has been limited and effective only within a severely restricted domain. That is, a computer can appear to be brainy, but only if you limit the environment within which it has to demonstrate those brains.

In this section of the book, we have four programs which will give your computer the appearance of intelligence, and certainly provide impressive demonstrations of 'electronic brainpower' when the programs are shown to computer-naive people.

Despite the claims made by some people (such as Carl Sagan, "In Defense of Robots", Broca's Brains) that human beings are, essentially, just well-made computers, we sense there is a wide gulf between our own perception of our selfhood, and the total non-awareness that computers can have (at least at present) of their own existence. And it seems to me that this self-perception lies at the heart of at least one aspect of intelligence.

Computers can function extremely 'intelligently' in restricted areas. Look at the five inch square, hand-held chess machines. Within the limited domain of chess game, a solid appearance of intelligence can be created.

A common stumbling block in the work of early researchers into the field of artificial intelligence was that writing a program which emulated some aspect of human reasoning (such as a very impressive Checkers program by Samuels) did not necessarily lead researchers any closer to either producing behavior

which could be classed as intelligent, nor towards a theoretical understanding of the processes of human reasoning and deduction. "Progress in producing intelligent behavior is not necessarily progress towards producing thought" (Norman Whaland, When Is A Program Intelligent?, Creative Computing, February 1981, pp. 44 - 49).

There is another way of approaching the problem, which I call the "if it quacks like a duck, it's a duck" approach. In the book "The Turing Criterion - Machine Intelligent Programs for the 16K ZX81" (Charlton et al, Interface Publications London, 1982), which I edited, I answer the question "What is the Turing Criterion?" in the following words:

"In 1950, the English mathematician and logician A. M. Turing proposed what has become known as the 'Turing Criterion' for machine intelligence.

"He said that if you were dealing with something at the end of a wire that could be a machine, or could be human, and you could not tell -- from the responses coming to you over the wire -- whether that with which you were dealing was human or machine, the 'thing' at the other end was, by definition, intelligent.

On this basis, it would be possible to class many, many computer programs as intelligent. Certainly those within this section of the book could almost pass the test, especially if you were prepared to accept the possibility that some pretty stupid humans were dealing with you.

Time now to explore the world's of Artificial Intelligence with the following programs:

- \* Electronic Brain (in which the computer tries to crack a 'mastermind-like' code you've worked out)
- \* Amanuensis (in which the computer accepts the task of writing poetry)

# ELECTRONIC BRAIN

ELECTRONIC BRAIN is an attempt to write a program to crack a 'Mastermind' type code in an apparently intelligent manner. You think of a numerical code, and the computer tries to guess it. You tell the computer how well it is doing by awarding - after each guess it makes - a 'black' for any digit correctly guessed in the right position within the code, and a 'white' for any digit which appears within your code, but not in the position the computer indicated.

The problem is not totally straightforward, as the computer does not know, for certain, which digit produced which result. I wrote two versions of this program, one to solve three-digit codes, and a modification of that program to solve four-digit ones. It works in a simple manner (although implementing the relatively simple idea behind the program was not particularly easy). Everytime a digit appears in a code which is awarded a black, every digit within that code is weighted so that it appears more often in future guesses. The more blacks in that particular code, the higher the weighting each code gets. A much smaller weighting is awarded if the code gets one or more whites. Any code getting neither a black nor a white leads to all the digits within that guess being totally removed from future consideration.

As I said, the three-digit version is the basic upon which the four-digit program was written. Enter the three-digit one first, and then save it in that form, before making the needed conversion to make it work as a four-digit program. Whereas the three-digit version works reasonably well, and reasonably quickly, the four-digit program grinds almost to a halt towards the end, as it tries to generate codes

which (a) reflect the weighting that previous answers have produced, and (b) are not codes which have been previously suggested in that particular run.

Here's the program solving a three-digit code:

GUESS NUMBER 1

MY GUESS IS 1 2 3

HOW MANY BLACKS? 1

AND HOW MANY WHITES? 0

GUESS NUMBER 2

MY GUESS IS 4 5 6

HOW MANY BLACKS? 1

AND HOW MANY WHITES? 0

GUESS NUMBER 3

MY GUESS IS 7 8 9

HOW MANY BLACKS? 1

And this is the listing that produced it:

```
10 REM ELECTRONIC BRAIN-3 DIGITS
20 GOSUB 500:REM INITIALISE
30 REM MAKE A GUESS
40 IF GUESS<3 THEN FOR Z=1 TO 3:B(Z)=Z+3*GUESS:
    NEXT Z:GOTO 60
50 GOSUB 280
60 CLS
70 GUESS=GUESS+1
80 PRINT:PRINT
90 PRINT "GUESS NUMBER" GUESS
100 PRINT
110 PRINT "MY GUESS IS" B(1);B(2);B(3)
120 PRINT:PRINT
125 B#=INKEY#
130 PRINT "HOW MANY BLACKS? ";
135 A#=INKEY#:IF A#<>"0" AND A#<>"1" AND A#<>"2"
    AND A#<>"3" THEN 135
137 PRINT A#:B=VAL(A#)
140 IF B=3 THEN 570
150 IF B=2 THEN 190
160 PRINT:PRINT
165 B#=INKEY#
170 PRINT "AND HOW MANY WHITES? ";
175 A#=INKEY#:IF A#<>"0" AND A#<>"1" AND A#<>"2"
    AND A#<>"3" THEN 175
177 PRINT A#:W=VAL(A#)
180 IF W+B=3 THEN D=3:C(1)=B(1):C(2)=B(2):C(3)=B(3)
190 IF B+W=0 THEN C(B(1))=0:C(B(2))=0:C(B(3))=0:GOTO 30
200 IF B>AID THEN FOR Z=1 TO 3:E(Z)=B(Z):NEXT Z:AID=B
210 FOR Z=1 TO 9
220 FOR D=1 TO 3
230 IF B(D)=C(Z) THEN C(Z)=C(Z)+(B+W)*100+W*10
240 NEXT D
250 NEXT Z
260 GOTO 30
270 REM PICK THREE NUMBERS
280 FOR Z=1 TO 3
290 D1=C(RND(0))
300 IF D1=0 THEN 290
310 D2=C(RND(0))
320 IF D2=0 THEN 310
330 IF INT(D1/10)>INT(D2/10) THEN B(Z)=D1
```

```
340 IF INT(D1/10)<INT(D2/10) THEN B(Z)=D2
350 IF INT(D1/10)=INT(D2/10) THEN B(Z)=D1
360 IF B(Z)>100 THEN B(Z)=B(Z)-100*INT(B(Z)/100)
    :GOTO 370
370 IF B(Z)>10 THEN B(Z)=B(Z)-10*INT(B(Z)/10):GOTO 380
380 NEXT Z
390 IF B(1)=B(2) OR B(1)=B(3) OR B(2)=B(3) THEN 280
400 IF AID>0 THEN CO=0:FOR Z=1 TO 3:IF B(Z)=E(Z)
    THEN CO=CO+1
410 IF AID>0 THEN NEXT Z:IF CO<AID THEN 280
420 M=100*B(1)+10*B(2)+B(3)
430 K(GUESS)=M
440 IF GUESS<3 THEN 500
450 CO=1
460 CO=CO+1
470 IF K(CO)=M THEN 290
480 IF CO<GUESS-1 THEN 460
490 RETURN
500 REM INITIALISE
510 GUESS=0:Q=9:AID=0
520 DIM B(3),C(9),E(3),K(100)
530 FOR Z=1 TO 9
540 C(Z)=Z
550 NEXT Z
560 RETURN
570 PRINT:PRINT
580 PRINT "I GUESSED YOUR CODE OF" B(1);B(2);B(3)
590 PRINT TAB(5);"IN JUST" GUESS "GUESSES"
600 FOR Z=1 TO 5000:NEXT Z:CLS
```

This is the listing to solve four-digit codes, which is based on the three-digit one. Note the addition of '- 1' in line 420 of this program, as well as the other changes:

```
10 REM ELECTRONIC BRAIN-4 DIGITS
20 GOSUB 510:REM INITIALISE
30 REM MAKE A GUESS
40 IF GUESS=0 THEN FOR Z=1 TO 4:B(Z)=Z:NEXT Z:GOTO 70
50 IF GUESS=1 THEN FOR Z=1 TO 4:B(Z)=Z+4:NEXT Z:GOTO
60 GOSUB 280
70 CLS
80 GUESS=GUESS+1
90 PRINT:PRINT
```



```

100 PRINT "GUESS NUMBER" GUESS
110 PRINT
120 PRINT "MY GUESS IS" B(1); B(2); B(3); B(4)
130 PRINT: PRINT
140 INPUT "HOW MANY BLACKS"; B
150 IF B=4 THEN 580
160 IF B=3 THEN 190
170 PRINT: PRINT
180 INPUT "AND HOW MANY WHITES"; W
190 IF W+B=4 THEN Q=4: C(1)=B(1): C(2)=B(2):
      C(3)=B(3): C(4)=B(4)
200 IF B+W=0 THEN C(B(1))=0: C(B(2))=0:
      C(B(3))=0: C(B(4))=0: GOTO 30
205 IF B(2)=B(4) OR B(3)=B(4) THEN 290
210 IF B>AID THEN FOR Z=1 TO 4: E(Z)=B(Z): NEXT Z: AID=B
220 FOR Z=1 TO 9
230 FOR D=1 TO 4
240 IF B(D)=C(Z) THEN C(Z)=C(Z)+(B+W)*100+W*10
250 NEXT D
260 NEXT Z
270 GOTO 30
280 REM PICK FOUR NUMBERS
290 FOR Z=1 TO 4
300 D1=C(RND(Q))
310 IF D1=0 THEN 300
320 D2=C(RND(Q))
330 IF D2=0 THEN 320
340 IF INT(D1/10) > INT(D2/10) THEN B(Z)=D1
350 IF INT(D1/10) < INT(D2/10) THEN B(Z)=D2
360 IF INT(D1/10) = INT(D2/10) THEN B(Z)=D1
370 IF B(Z)>100 THEN B(Z)=B(Z)-100*INT(B(Z)/100):
      GOTO 370
380 IF B(Z)>10 THEN B(Z)=B(Z)-10*INT(B(Z)/10): GOTO 380
390 NEXT Z
400 IF B(1)=B(2) OR B(1)=B(3) OR B(1)=B(4)
      OR B(2)=B(3) THEN 290
405 IF B(2)=B(4) OR B(3)=B(4) THEN 290
410 IF AID>0 THEN CO=0: FOR Z=1 TO 4: IF B(Z)=E(Z)
      THEN CO=CO+1
420 IF AID>0 THEN NEXT Z: IF CO<AID-1 THEN 290
430 M=1000*B(1)+100*B(2)+10*B(3)+B(4)
440 K(GUESS)=M
450 IF GUESS<3 THEN 500

```

```

460 CO=1
470 CO=CO+1
480 IF K(CO)=M THEN 290
490 IF CO<GUESS-1 THEN 470
500 RETURN
510 REM INITIALISE
520 GUESS=0: Q=9: AID=0
530 DIM B(4), C(9), E(4), K(100)
540 FOR Z=1 TO 9
550 C(Z)=Z
560 NEXT Z
570 RETURN
580 PRINT: PRINT
590 PRINT "I GUESSED YOUR CODE OF" B(1); B(2); B(3); B(4)
600 PRINT TAB(5); "IN JUST" GUESS "GUESSES"
610 FOR Z=1 TO 5000: NEXT Z
620 CLS: END

```

```

** ** *** * * * ** * * * *** ** **
 * * * * * * * * * * *
***** * * * * ***** * * * * *****
 ** * ** ** * **
 * * ** ** * **** * ** ** * *
 *** **** * **** * * **** ***
 * * ** * ** ** * * ** * *
***** * ** ***** ** * *****
***** * ** ***** ** * *****
 * * ** * ** ***** * **** **
 * * ** ** * **** * ** * *
 ** * ** ** * **
***** * * * * ***** * * * * *****
 * * * * * * * * * * *
** ** *** * * * ** * * * *** ** **

```

# AMANUENSIS

Next we'll have our poetry writing program. This demands no interaction from you, except for that of admiring the wonderful (!) output of the program. Here's a sample of the kind of verse it produces:

THE CONVOY CLAIMED SLAVISHLY  
IN THE COURT...  
PRAYING FOR AN ASTRAL INFLUENCE,  
TO DENUR THE FAUN...  
EXHAUSTING, THEN GNAWING  
...IMPOUNDING, BREAKING.

THE POSTULANT BESEECHED NEGLECTFULLY  
IN THE ENEMY'S CAMP...  
WANTING FOR A DOGMATIST,  
TO RECANT THE DJINN...  
DISPENSING WITH, THEN ERODING  
...ENCLOSING, WOUNDING.

THE ASPIRANT CRAVED SUCCESSFULLY  
IN THE RACECOURSE...  
PRAYING FOR HOBSON'S CHOICE,  
TO QUENCH THE GHOST...  
EMPTYING, THEN BLIGHTING  
...HANDCUFFING, ROTTING.

THE ASPIRANT ENTREATED HALTINGLY  
IN THE HIPPODROME...  
PRAYING FOR A BLIND BARGAIN,  
TO RECANT THE NAIAD...  
WASTING, THEN BREAKING  
...RESTRAINING, UNDERMINING.

THE POSTULANT ENTREATED ENDLESSLY  
IN THE PLAYGROUND...  
WAITING FOR HOBSON'S CHOICE,  
TO RELINQUISH THE SATUR...  
DISPENSING WITH, THEN DAMAGING  
...COERCING, BREAKING.

THE BEGGAR CLAIMED HALTINGLY  
IN THE THEATRE OF WAR...  
ASKING FOR FORTUNE'S WHEEL,  
TO BACK THE NAIAD...  
DISPENSING WITH, THEN BREAKING  
...MUZZLING, WOUNDING.

This is the listing of AMANUENSIS. Once you've run it for a while, you may well wish to change the words in the DATA statements, to vary the kind of poetry it produces.

```
10 REM AMANUENSIS
20 CLS: DIM A$(12), Z$(10)
30 REM *****
40 RESTORE: GOSUB 450: A$(1)=B$
50 GOSUB 570: A$(2)=B$
60 GOSUB 570: A$(3)=B$
70 GOSUB 570: A$(4)=B$
80 GOSUB 570: A$(5)=B$
90 GOSUB 570: A$(6)=B$
100 GOSUB 570: A$(7)=B$
110 GOSUB 570: A$(8)=B$
120 GOSUB 570: A$(9)=B$
130 GOSUB 570: A$(10)=B$
140 GOSUB 570: A$(11)=B$
145 RESTORE: FOR I=1 TO 100: READ V$: NEXT
150 GOSUB 570: A$(12)=B$
160 REM *****
170 PRINT "THE "; A$(1); " "; A$(2); " "; A$(3)
180 PRINT "      IN THE "; A$(4); "... "
190 PRINT A$(5); " FOR "; A$(6); ", "
200 PRINT "      TO "; A$(7); " THE "; A$(8); "... "
210 PRINT A$(9); ", THEN "; A$(10)
220 PRINT "      ..."; A$(11); ", "; A$(12); ". "
```

```

230 PRINT:PRINT:M$=INKEY$:M$=INKEY$
240 REM *****
250 PRINT:PRINT "PRESS 'C' TO CONTINUE"
255 IF INKEY$<>"C" THEN 255
260 CLS:GOTO 40
270 REM *****
280 DATA"APPLICANT", "SUPPLICANT", "MENDICANT",
      "BEGGAR", "ASPIRANT"
290 DATA "CLAIMANT", "POSTULANT", "CONVOY", "SENTINEL",
      "CASTELLAN"
300 DATA "ENTREATED", "BESEECHED", "WAITED", "BESOUGHT",
      "CONJURED"
310 DATA "PRESSED", "URGED", "CRAVED", "DEMANDED", "CLAIMED"
320 DATA "SUCCESSFULLY", "COURAGEOUSLY", "PROGRESSIVELY",
      "FOOLISHLY"
330 DATA "ENDLESSLY", "HELPLESSLY", "HALTINGLY", "SLAVISHLY"
340 DATA "NEGLECTFULLY", "PRIDEFULLY"
350 DATA "FIELD", "BATTLEFIELD", "PLAYGROUND", "CIRCUS
      RING", "COURT"
360 DATA "BEAR GARDEN", "ENEMY'S CAMP", "HIPPODROME",
      "RACECOURSE"
370 DATA "THEATRE OF WAR"
380 DATA "WAITING", "ASKING", "HOPING", "WATCHING"
      "SCREAMING"
390 DATA "CRYING", "STARING", "WANTING", "STAYING", "PRAYING"
400 DATA "THE FATES", "FORTUNE'S WHEEL", "HOBSON'S CHOICE"
410 DATA "THE STARS", "AN ASTRAL INFLUENCE", "A BLIND
      BARGAIN"
420 DATA "A SPELLBINDER", "A MALEVOLENT SPIRIT", "A ZEALOT"
430 DATA "A DOGMATIST"
440 DATA "QUENCH", "RECAT", "FORSEWAR", "ABJURE",
      "RENOUNCE", "WAVER"
450 DATA "RELINQUISH", "DEMUR", "SWERVE", "BACK"
460 DATA "GHOST", "SPECTRE", "HOBGOBLIN", "DJINN",
      "LEPRECHAUN"
470 DATA "PIXY", "FAUN", "SATUR", "DRYAD", "NAIAD"
480 DATA "WASTING", "SPENDING", "USING", "CONSUMING",
      "DRYING UP"
490 DATA "RUNNING OUT", "EMPTYING", "EXHAUSTING",
      "SQUANDERING"

```

```

500 DATA "DISPENSING WITH"
510 DATA "CORRODING", "ERODING", "BLIGHTING", "ROTTING",
      "GNAWING"
520 DATA "UNDERMINING", "SHAKING", "BREAKING", "DAMAGING",
      "WOUNDED"
530 DATA "RESTRAINING", "COERCING", "CHECKING", "CHECKING"
540 DATA "DETAINING", "ENCLOSING", "IMPOUNDING",
      "HANDCUFFING"
550 DATA "MUZZLING", "GAGGING", "SUPPRESSING"
560 REM *****
570 FOR N=1 TO 10
580 READ Z$(N):NEXT N
590 N=RND(10)
600 B#=Z$(N)
610 RETURN

```

# Just for Fun



Just for Fun



# Just for Fun

There is a range of games in this section of your book, all designed just for fun. From INNER SPRING to ROBOT MINEFIELD, a splendid time is guaranteed for all.

We'll start the 'just for fun' with a program which allows you to use your computer as a two-octave piano. It will be very clear how to play it when you get it up on the screen. Note that the X's in lines 50, 70 and 90 should be replaced with inverse spaces [with the exception of the third X in line 90, which stays as an X].

```
10 REM PIANO
20 CLS:DU=1:DIM A(26),B(26),C(26)
30 G$=CHR$(128):H$=CHR$(143):I$=H$+H$:J$=CHR$(224)
   +CHR$(224)
40 A$=G$+" "+G$+" "+G$+" "+G$+" "+G$
50 B$="XX XX XX XX XX XX XX XX"
60 C$=" 2 3 5 6 7"
70 D$="QX WX EX RX TX YX UX IX"
80 E$=" S D G H J "
90 F$="ZX XX CX VX BX NX MX ,X"
100 PRINTTAB(4);C$:PRINT TAB(6);A$:PRINTTAB(6);A$
110 PRINT TAB(4);B$:PRINT TAB(4);B$:PRINT TAB(4);D$
120 PRINT:PRINTTAB(4);E$:PRINT TAB(6);A$:PRINTTAB(6);A$
130 PRINTTAB(4);B$:PRINTTAB(4);B$:PRINTTAB(4);F$
140 PRINT:PRINT "PRESS THE SPACE BAR TO EXIT";
150 FOR I=1 TO 26:READ Z$:A(I)=ASC(Z$):NEXT
160 DATA 2,3,5,6,7,Q,W,E,R,T,Y,U,I,S,D,G,H,J,Z,
   X,C,V,B,N,M,", "
170 FOR I=1 TO 26:READ B(I):NEXT
180 DATA 38,41,47,50,53,132,135,138,141,144,147,150,153
190 DATA 262,265,271,274,277,356,359,362,365,368,
   371,374,377
200 FOR I=1 TO 26:READ C(I):NEXT
210 DATA 17,19,22,24,26,16,18,20,21,23,25,27,28
220 DATA 5,7,10,12,14,4,6,8,9,11,13,15,16
230 K$=INKEY$:IF K$="" THEN 230
240 IF K$=" " THEN PRINT@44B,"":END
```

```
250 FOR I=1 TO 26
260 IF K$=CHR$(A(I)) THEN FR=C(I) ELSE NEXT I
   :GOTO 230
270 IF I<6 OR I>13 AND I<19 THEN M$=G$:L$=H$:C=2
280 IF NOT (I<6 OR I>13 AND I<19) THEN L$=I$:
   M$=J$:C=3
290 COLOR C:PRINT@B(I),L$:SOUND FR,DU:PRINT@B(I),M$
300 N$=INKEY$:N$=INKEY$:N$=INKEY$:N$=INKEY$:
   K$="":GOTO 230
```

# FASTER

FASTER is an exciting reaction test game. You have ten goes at pressing the letter displayed by the computer in the shortest possible time. At the end of the round, your ten times will be displayed, along with the average time it took you to react.

```
10 REM FASTER
20 DIM A(10)
30 CLS:PRINT "PRESS 'S' TO START"
40 IF INKEY#<> "S" THEN 40
50 CLS:FOR I=1 TO 10
60 A=RND(26)+64
70 A$=CHR$(A)
80 PRINT@367,A$
90 K=RND(3200):FOR DE=1 TO K:M$=INKEY#
100 IF M$=A$ AND DE>K-100 THEN 250 ELSE NEXT DE
110 FOR C=0 TO 10 STEP .052
130 PRINT@269,USING "##.##";C
140 IF INKEY$=A$ THEN B=C:C=10
150 NEXT C
160 A(I)=B:PRINT@I*32,I;USING "##.##";A(I):N=N+B
170 NEXT I
180 AVERAGE=N/10:PRINT@384,"AVERAGE="AVERAGE:PRINT
190 FOR F=1 TO 5000:NEXT F
200 PRINT "IF YOU WOULD LIKE ANOTHER GO,"
210 PRINT "JUST PRESS 'Y'. OTHERWISE"
220 PRINT "PRESS 'N'"
230 M$=INKEY$:IF M$<> "Y" AND M$<> "N" THEN 230
240 IF M$="Y" THEN 30 ELSE END
250 PRINT@448,"TAKE YOUR FINGER OFF THAT KEY!!"
260 IF INKEY$=A$ THEN 250
270 PRINT@448,"
NEXT DE:GOTO 110
```

# INNER SPRING

We'll continue this section of the book with one of the simplest games of the lot, INNER SPRING. As the program explains, the computer produces two numbers between one and 13, and asks you to bet on the probability of the next number it thinks of lying between the first two. It's simple to play, and a lot of fun.

Here's what one round looks like in action:

```
MY FIRST NUMBER IS 2
MY SECOND IS 4
```

```
YOU HAVE $ 2
```

```
HOW MUCH DO YOU BET
MY NEXT NUMBER LIES
BETWEEN 2 AND 4 ? 1
```

```
MY NUMBER WAS 3
WELL DONE, YOU WIN $ 2
```

And here's the listing of INNER SPRING:

```
10 REM INNER SPRING
20 GOSUB 330
30 GOSUB 80
40 IF D<1 THEN GOTO 420
45 IF D>1000 THEN 530
50 GOSUB 340
60 GOTO 30
70 REM *****
80 PRINT:PRINT:PRINT
90 PRINT "MY FIRST NUMBER IS"A
100 PRINT " MY SECOND IS"B
110 PRINT
```

```

120 PRINT "YOU HAVE $"D
130 PRINT
140 PRINT "AND HOW MUCH DO YOU BET"
150 PRINT "MY NEXT NUMBER LIES"
160 PRINT "BETWEEN" A "AND" B ;
170 INPUT E
180 IF E>D THEN 170
190 D=D-E
200 GOSUB 500
210 PRINT:PRINT "MY NUMBER WAS" C
220 GOSUB 500
230 IF NOT (C>A AND C<B OR C<A AND C>B) THEN 290
240 PRINT "WELL DONE, YOU WIN $"2*E
250 D=D+3*E
260 GOSUB 500
270 RETURN
280 REM *****
290 PRINT "SORRY, YOU LOSE $"E
300 GOSUB 500
310 RETURN
320 REM *****
330 D=20
340 CLS
350 A=RND(13)
360 B=RND(13)
370 IF ABS(A-B)<2 OR ABS(A-B)>6 THEN 360
380 C=RND(13)
390 IF A=C OR B=C THEN 380
400 RETURN
410 REM *****
420 PRINT
430 PRINT "THE GAME IS OVER":GOSUB 500
440 PRINT
450 PRINT "YOU ARE BROKE!"
460 PRINT
470 PRINT "THANKS FOR THE GAME"
480 END
490 REM *****
500 FOR Z=1 TO 1000
510 NEXT Z
520 RETURN
530 PRINT:PRINT "CONGRATULATIONS!"
540 PRINT "YOU HAVE REACHED $"D
550 SOUND 21,4:SOUND 16,2:SOUND 16,1:SOUND 18,4
560 SOUND 0,1:SOUND 20,4:SOUND 21,4:END

```

# ROBOT MINEFIELD

Time now to face the terrors, the dangers, and the horrors of the ROBOT MINEFIELD. There you are, peacefully tripping along one day in the sun, when suddenly you find yourself trapped in a walled area, with a number of manic robots, all bent on your destruction.

The robots are shown on the display as dollar signs, the H is you, and the asterisks (\*) are the mines which are deadly to both you and robots. Robots, pretty clumsily programmed, can detect you, but they cannot detect mines. Therefore, your salvation lies in getting a mine or two in between you and the robots, so that when they come towards you they'll smash into mines and be wiped out.

The robots have one trick up their sleeves. They can merge into one another, so you may think only one robot is approaching you, when at the last minute it splits into two or more, one going each way around a mine. This is not a game for those who seek the quiet life.

You can move north, south, east or west (with north at the top of the screen), and you enter the move you want to make by entering the initial of the desired direction (so you enter "N" to go north, "S" to go south and so on).

```

10 REM ROBOT MINEFIELD
20 GOTO 250
30 REM MOVE ROBOTS
40 TALLY=0
50 FOR E=1 TO 4
60 IF A(B(E),C(E))=42 THEN TALLY=TALLY+1:GOTO 200
70 X=B(E):Y=C(E)
80 IF B(E)<D THEN B(E)=B(E)+1
90 IF B(E)>D AND RND(10)>3 THEN B(E)=B(E)-1
100 IF C(E)<F AND RND(10)>3 THEN C(E)=C(E)+1

```

```

110 IF C(E)>F THEN C(E)=C(E)-1
120 IF B(E)<2 THEN B(E)=2
130 IF B(E)>14 THEN B(E)=14
140 IF C(E)<2 THEN C(E)=2
150 IF C(E)>14 THEN C(E)=14
160 A(X,Y)=46
170 IF A(B(E),C(E))=42 THEN TALLY=TALLY+1:GOTO 200
180 IF A(B(E),C(E))=200 THEN A(B(E),C(E))=228:
      GOSUB 300:GOTO 910
190 A(B(E),C(E))=228
200 NEXT E
210 IF TALLY<CH THEN TALLY=CH
220 CH=TALLY
230 IF TALLY=4 THEN GOSUB 300:GOTO 970
240 RETURN
250 GOSUB 530:REM INITIALISE
260 GOSUB 300:REM PRINT MINEFIELD
270 GOSUB 30:REM ROBOTS MOVE
280 GOSUB 420:REM HUMAN MOVE
290 GOTO 260
300 REM PRINT MINEFIELD
310 PRINTCHR$(28);
320 IF TALLY>0 THEN PRINT@480,"DEAD ROBOT TALLY:"TALL
330 IF TALLY>0 THEN PRINT CHR$(28);
350 FOR B=1 TO 15
360 FOR C=1 TO 15
370 PRINT CHR$(A(B,C));" ";
380 NEXT C
390 PRINT
400 NEXT B
410 RETURN
420 REM PLAYER MOVE
430 A(D,F)=46:B#=INKEY#
440 A#=INKEY#
450 IF A#="" THEN 440
455 SOUND 31,1
460 IF A#="." AND D>2 THEN D=D-1
465 IF A#"A" AND D>2 AND F>2 THEN D=D-1:F=F-1
470 IF A#" " AND D<14 THEN D=D+1
475 IF A#"S" AND D>2 AND F<14 THEN D=D-1:F=F+1
480 IF A#"," AND F<14 THEN F=F+1
485 IF A#"X" AND D<14 AND F<14 THEN D=D+1:F=F+1
490 IF A#"M" AND F>2 THEN F=F-1
495 IF A#"Z" AND D<14 AND F>2 THEN D=D+1:F=F-1

```

```

500 IF A(D,F)=42 THEN GOSUB 300:GOTO 990
510 A(D,F)=200
520 RETURN
530 REM INITIATE
550 DIM A(15,15);B(4),C(4)
560 CLS
570 PRINT:PRINT "PLEASE STAND BY FOR A MOMENT..."
580 CH=0
610 REM PLACE WALLS
620 FOR B=1 TO 15
630 FOR C=1 TO 15
640 A(B,C)=46
650 IF B=1 OR B=15 OR C=1 OR C=15 THEN A(B,C)=88
660 NEXT C,B
670 REM PLACE MINES
680 FOR B=1 TO 20
690 C=RND(13)
700 D=RND(13)
710 IF A(C,D)=88 THEN 690
720 A(C,D)=42
730 NEXT
740 DATA 4,4,13,8,8,3,12,7
750 REM PLACE ROBOTS
760 FOR E=1 TO 4
770 D=RND(13)+1
780 F=RND(13)+1
790 IF A(D,F)<>46 THEN 770
800 B(E)=D:C(E)=F
810 A(B(E),C(E))=228
820 NEXT E
830 REM PLACE HUMAN
840 D=RND(13)+1
850 F=RND(13)+1
860 IF A(D,F)<>46 THEN 830
870 A(D,F)=200
880 CLS:RETURN
890 REM HUMAN AT D,F
900 REM ROBOTS AT B(E),C(E)
910 REM END OF GAME
920 PRINT@480," THE ROBOTS HAVE GOT YOU!!"
925 SOUND 16,3:SOUND 16,2:SOUND 13,4:SOUND 18,2
927 SOUND 16,5:SOUND 13,6
930 CH=0:TALLY=0
940 A#=""

```



```
950 IF INKEY$("<>") THEN 950
960 GOSUB 570:GOTO 260
970 PRINT@480," YOU'VE DEFEATED THE ROBOTS!!"
980 GOTO 930
990 PRINT@480," YOU'VE RUN INTO A MINE":SOUND 1,8
1000 GOTO 930
```

# Fun with your Printer CELESTIA

The first of the two programs in this section is CELESTIA, which you can run either on your TV screen, or on the screen and to the printer at the same time. It produces an infinite series of evolving patterns.

The patterns develop according to the rules of the famous computer game of LIFE, developed by John Conway while at Gonville and Caius College at Cambridge in the UK. Martin Gardiner spread the game throughout the world when he wrote about it in Scientific American in October, 1970.

In LIFE, cells are born, grow and die according to rules which Conway invented. Each cell on a grid (the colony of cells is imagined to be evolving on a grid) is surrounded by eight others, and the state of those eight other cells dictates what happens to the cell in question in the following generation.

The rules which govern the evolution of the cells are as follows:

- \* If a cell has two or three surrounding it, it survives to the next generation

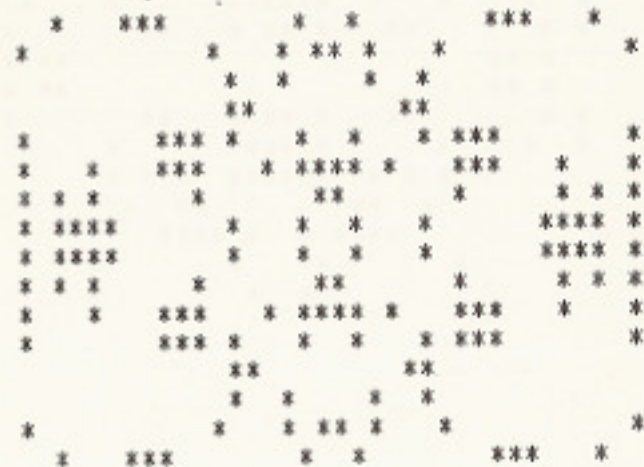
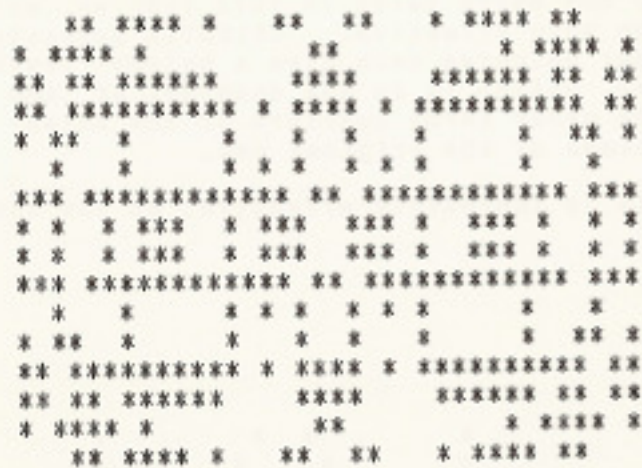
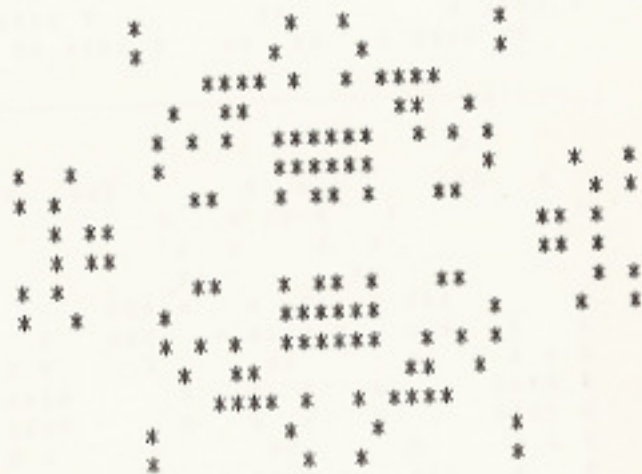
- \* If there are three, and just three, full cells next to an empty one, a cell will be 'born' in that empty space in the next generation

- \* Any cell with four of its neighbouring cells occupied dies in the next generation

Don't worry, you don't have to know the rules, as the computer interprets them quite happily by

itself. The rules produce patterns which are far more attractive (and far less predictable) than you could possibly imagine by reading the rules. There is just one extra twist in this program, which makes it even more effective. CELESTIA actually prints out four colonies each time a colony evolves. The original colony is in one quadrant of the screen, and the other three quarters of the screen contains reflections of the original one.

CELESTIA is incredibly effective, as you can see:



This is the listing to give CELESTIA designs on your microcomputer:

```
10 REM CELESTIA
20 GOSUB 450:REM INITIALISE
30 GOSUB 60:REM PRINT OUT
40 GOSUB 280:REM EVOLVE
50 GOTO 30
60 CLS
80 FOR X=2 TO 9
90 FOR Y=2 TO 19
100 PRINT CHR$(A(X,Y));
110 NEXT Y
120 FOR Y=19 TO 2 STEP -1
130 PRINT CHR$(A(X,Y));
140 NEXT Y
150 PRINT
160 NEXT X
170 FOR X=9 TO 2 STEP -1
180 FOR Y=2 TO 19
190 PRINT CHR$(A(X,Y));
200 NEXT Y
210 FOR Y=19 TO 2 STEP -1
220 PRINT CHR$(A(X,Y));
230 NEXT Y
240 PRINT
250 NEXT X
260 IF N=0 THEN 640
270 RETURN
280 FOR X=2 TO 9:FOR Y=2 TO 19
290 C=0
300 IF A(X-1,Y-1)=B THEN C=C+1
310 IF A(X-1,Y)=B THEN C=C+1
320 IF A(X-1,Y+1)=B THEN C=C+1
330 IF A(X,Y-1)=B THEN C=C+1
340 IF A(X,Y+1)=B THEN C=C+1
350 IF A(X+1,Y-1)=B THEN C=C+1
360 IF A(X+1,Y)=B THEN C=C+1
370 IF A(X+1,Y+1)=B THEN C=C+1
380 IF A(X,Y)=B AND C<>3 AND C<>2 THEN B(X,Y)=E
390 IF A(X,Y)=E AND C=3 THEN B(X,Y)=B
400 NEXT Y,X
410 FOR X=2 TO 9:FOR Y=2 TO 19
```

```
420 A(X,Y)=B(X,Y)
430 NEXT Y,X
440 RETURN
450 REM INITIALISE
460 CLS
470 PRINT "PRESS THE SPACE BAR":PRINT"WHEN YOU'RE
READY TO BEGIN"
480 N=1
490 IF INKEY#<> " " THEN 490
510 PRINT "DO YOU WANT A COPY ON"
520 PRINT "YOUR PRINTER (Y OR N)"
530 A$=INKEY$:IF A$<> "N" AND A$<> "Y" THEN 530
540 IF A$="Y" THEN N=0
550 CLS
560 B=ASC("#");E=ASC(" ")
570 DIM A(10,20),B(10,20)
580 FOR X=2 TO 9:FOR Y=2 TO 19
590 A(X,Y)=E
600 IF RND(0)>.45 THEN A(X,Y)=B
610 B(X,Y)=A(X,Y)
620 NEXT Y,X
630 RETURN
640 LPRINT "-----"
645 REM 6 SPACES, 34 -'S
650 LPRINT:LPRINT:LPRINT " ";
660 FOR X=2 TO 9
670 FOR Y=2 TO 19
680 LPRINT CHR$(A(X,Y));
690 NEXT Y
700 FOR Y=19 TO 2 STEP -1
710 LPRINT CHR$(A(X,Y));
720 NEXT Y
730 LPRINT:LPRINT " ";
740 NEXT X
750 FOR X=9 TO 2 STEP -1
760 FOR Y=2 TO 19
770 LPRINT CHR$(A(X,Y));
780 NEXT Y
790 FOR Y=19 TO 2 STEP -1
800 LPRINT CHR$(A(X,Y));
810 NEXT Y
820 LPRINT:LPRINT " ";
830 NEXT X
840 LPRINT
850 RETURN
```

The listing we've just seen is to print the pattern direct on your printer. If you don't want to employ the printer in this way, you can modify it as follows to fit just once pattern on the screen.

Line 60 should be changed to 60 PRINT CHR\$(28); and line numbers 120 to 140, plus 170 to 250 should be deleted. Here's a listing with the changes made:

```
10 REM CELESTIA
20 GOSUB 450:REM INITIALISE
30 GOSUB 60 :REM PRINT OUT
40 GOSUB 280:REM EVOLVE
50 GOTO 30
60 PRINT CHR$(28);
80 FOR X=2 TO 9
90 FOR Y=2 TO 19
100 PRINT CHR$(A(X,Y));
110 NEXT Y
150 PRINT
160 NEXT X
260 IF N=0 THEN 640
270 RETURN
280 FOR X=2 TO 9:FOR Y=2 TO 19
290 C=0
300 IF A(X-1,Y-1)=B THEN C=C+1
310 IF A(X-1,Y)=B THEN C=C+1
320 IF A(X-1,Y+1)=B THEN C=C+1
330 IF A(X,Y-1)=B THEN C=C+1
340 IF A(X,Y+1)=B THEN C=C+1
350 IF A(X+1,Y-1)=B THEN C=C+1
360 IF A(X+1,Y)=B THEN C=C+1
370 IF A(X+1,Y+1)=B THEN C=C+1
380 IF A(X,Y)=B AND C<>3 AND C<>2 THEN B(X,Y)=E
390 IF A(X,Y)=E AND C=3 THEN B(X,Y)=B
400 NEXT Y,X
410 FOR X=2 TO 9:FOR Y=2 TO 19
420 A(X,Y)=B(X,Y)
430 NEXT Y,X
440 RETURN
450 REM INITIALISE
460 CLS
470 PRINT "PRESS THE SPACE BAR":PRINT"WHEN
YOU'RE READY TO BEGIN"
```

```
480 N=1
490 IF INKEY$<> " " THEN 490
510 PRINT "DO YOU WANT A COPY ON"
520 PRINT "YOUR PRINTER (Y OR N)"
530 A$=INKEY$:IF A$<> "N" AND A$<> "Y" THEN 530
540 IF A$="Y" THEN N=0
550 CLS
560 B=ASC("*");E=ASC(" ")
570 DIM A(10,20),B(10,20)
580 FOR X=2 TO 9:FOR Y=2 TO 19
590 A(X,Y)=E
600 IF RND(0)>.45 THEN A(X,Y)=B
610 B(X,Y)=A(X,Y)
620 NEXT Y,X
630 RETURN
640 LPRINT "
645 REM 6 SPACES, 34 -'S
650 LPRINT:LPRINT:LPRINT " ";
660 FOR X=2 TO 9
670 FOR Y=2 TO 19
680 LPRINT CHR$(A(X,Y));
690 NEXT Y
700 FOR Y=19 TO 2 STEP -1
710 LPRINT CHR$(A(X,Y));
720 NEXT Y
730 LPRINT:LPRINT " ";
740 NEXT X
750 FOR X=9 TO 2 STEP -1
760 FOR Y=2 TO 19
770 LPRINT CHR$(A(X,Y));
780 NEXT Y
790 FOR Y=19 TO 2 STEP -1
800 LPRINT CHR$(A(X,Y));
810 NEXT Y
820 LPRINT:LPRINT " ";
830 NEXT X
840 LPRINT
850 RETURN
```



```

270 LPRINT"MMMMMM":LPRINT" M":LPRINT" MM":
LPRINT" M"
275 LPRINT"MMMMMM":RETURN
280 LPRINT"NNNNNNN":LPRINT" NN":LPRINT" NN"
:LPRINT" NN"
285 LPRINT"NNNNNNN":RETURN
290 LPRINT" OOOOO":LPRINT" O":LPRINT" O"
295 LPRINT" O O":LPRINT" OOOOO":RETURN
300 LPRINT" P P P P P":LPRINT" P P":LPRINT" P P"
305 LPRINT" P P":LPRINT" P P":RETURN
310 LPRINT" Q Q Q Q Q":LPRINT" Q Q":LPRINT" Q Q Q"
315 LPRINT" Q Q":LPRINT" Q Q Q Q":RETURN
320 LPRINT" R R R R R":LPRINT" R R":LPRINT" R R R"
325 LPRINT" R R R":LPRINT" R R R":RETURN
330 LPRINT" S S S":LPRINT" S S S":LPRINT" S S S"
335 LPRINT" S S S":LPRINT" S S S":RETURN
340 LPRINT" T":LPRINT" T":LPRINT" T T T T T":
345 LPRINT" T":LPRINT" T":RETURN
350 LPRINT" UUUUUU":LPRINT" U":LPRINT" U":LPRINT" U":
LPRINT" UUUUUU"
355 RETURN
360 LPRINT" VVVV":LPRINT" V V":LPRINT" V"
:LPRINT" V V"
365 LPRINT" VVVV":RETURN
370 LPRINT" WWWWWW":LPRINT" W":LPRINT" WWWWWW":LPRINT" W"
375 LPRINT" WWWWWW":RETURN
380 LPRINT" X X":LPRINT" X X X X":LPRINT" X"
:LPRINT" X X X X"
385 LPRINT" X X":RETURN
390 LPRINT" Y Y":LPRINT" Y Y":LPRINT" Y Y Y Y"
:LPRINT" Y Y"
395 LPRINT" Y Y":RETURN
400 LPRINT" Z Z":LPRINT" Z Z Z Z":LPRINT" Z Z Z"
405 LPRINT" Z Z Z":LPRINT" Z Z Z":RETURN
410 LPRINT"..":LPRINT"..":LPRINT:LPRINT:LPRINT:RETUR
420 LPRINT"! ! ! !":LPRINT"! ! ! !":LPRINT:
LPRINT:RETUR
430 LPRINT:LPRINT:LPRINT:LPRINT:RETURN
500 IF D=1 THEN GOSUB 150:RETURN:REM "A"
510 IF D=2 THEN GOSUB 160:RETURN:REM "B"
520 IF D=3 THEN GOSUB 170:RETURN:REM "C"
530 IF D=4 THEN GOSUB 180:RETURN:REM "D"
540 IF D=5 THEN GOSUB 190:RETURN:REM "E"
550 IF D=6 THEN GOSUB 200:RETURN:REM "F"

```

```

560 IF D=7 THEN GOSUB 210:RETURN:REM "G"
570 IF D=8 THEN GOSUB 220:RETURN:REM "H"
580 IF D=9 THEN GOSUB 230:RETURN:REM "I"
590 IF D=10 THEN GOSUB 240:RETURN:REM "J"
600 IF D=11 THEN GOSUB 250:RETURN:REM "K"
610 IF D=12 THEN GOSUB 260:RETURN:REM "L"
620 IF D=13 THEN GOSUB 270:RETURN:REM "M"
630 IF D=14 THEN GOSUB 280:RETURN:REM "N"
640 IF D=15 THEN GOSUB 290:RETURN:REM "O"
650 IF D=16 THEN GOSUB 300:RETURN:REM "P"
660 IF D=17 THEN GOSUB 310:RETURN:REM "Q"
670 IF D=18 THEN GOSUB 320:RETURN:REM "R"
680 IF D=19 THEN GOSUB 330:RETURN:REM "S"
690 IF D=20 THEN GOSUB 340:RETURN:REM "T"
700 IF D=21 THEN GOSUB 350:RETURN:REM "U"
710 IF D=22 THEN GOSUB 360:RETURN:REM "V"
720 IF D=23 THEN GOSUB 370:RETURN:REM "W"
730 IF D=24 THEN GOSUB 380:RETURN:REM "X"
740 IF D=25 THEN GOSUB 390:RETURN:REM "Y"
750 IF D=26 THEN GOSUB 400:RETURN:REM "Z"
999 END
1000 RETURN

```

# Space Games



# Space Games

Good old space, the final frontier. No collection of computer games would be complete without a few space games.

## MOONLANDER

The scenario of this program is pretty familiar. You are landing on the moon, with limited fuel, and for some reason your onboard computer is incapable of doing anything except reporting the state of the flight to you. The information you get is, of necessity, limited but adequate.

You are told, at all times, your height above the moon's surface, how much fuel you have left, and the speed with which you are descending. You have to enter the correct amount of thrust to ensure you land gently on the surface. The fuel you have is limited, so you must ration it to ensure you have enough left in the last few seconds of flight to cushion your impact. A 'high-score' feature (lines 370 and 380) ensures you and your friends return to this program time and time again in an effort to get a better 'galactic rating' and to creating a less deep crater.

```
10 REM MOONLANDER
20 HS=-10000
30 CLS:PRINT@34,"HEIGHT:":PRINT@64,"VELOCITY:"
      :PRINT@100,"FUEL:"

40 A=-20-RND(60):REM INITIAL VELOCITY
50 B=1200+RND(380):REM HEIGHT
60 C=320+RND(90):REM FUEL
70 S$=CHR$(252)+CHR$(234)+CHR$(254)
80 FOR P=29152 TO 29183:POKE P,132:NEXT:Q=0
90 B=INT(B):A=INT(A):C=INT(C)
100 PRINT@42,USING "####";B:PRINT@74,USING "####";A
```

```
110 PRINT@106,USING "####";C
120 PRINT@(Q*32)+20,"  ":Q=INT(16-B/100)
150 PRINT@(Q*32)+20,S$
190 PRINT@416,"THRUST";
200 T$=INKEY$:IF VAL(T$)=0 AND T$<>"0" THEN 200
210 T=VAL(T$):SOUND Q,1:PRINT T
220 IF T>C THEN T=0
230 C=C-T
240 B=B+A+(T-5)/2
250 A=A+(T-5)/2
260 IF C<1 AND B>100 THEN 300
270 IF ABS(B)<20 AND ABS(A)<15 THEN 420
280 IF B>19 THEN 90
290 IF C>1 THEN 310
300 PRINT@352," YOU HAVE RUN OUT OF FUEL"
310 PRINT@384,"YOUR SHIP CRASHED AT"ABS(A)"KPH"
320 PRINT@416,"CREATING A"INT(ABS(A)*45)
      "METER DEEP CRATER"

330 SC=100-ABS(A):FOR P=Q TO 1 STEP -1:SOUND P,1:NEXT P
340 FOR P=1 TO 5000:NEXT P
360 CLS:PRINT@320,"YOUR GALACTIC RATING IS"SC
370 IF SC>HS THEN HS=SC
380 PRINT@352,"BEST RATING SO FAR IS"HS
390 PRINT@384,"PLEASE STAND BY FOR YOUR
      NEXT MISSION"

400 FOR P=1 TO 1500:NEXT P
410 GOTO 30
420 PRINT@384,"YOU HAVE LANDED SAFELY":PRINT
430 PRINT@416,"WELL DONE, INTREPID CAPTAIN":
      SC=C*234:GOTO 340
```



# HYPERWAR

HYPERWAR is a major space simulation, which puts you in charge of patrolling a cube-shaped sector of space, which is ten parsecs along each side. The enemy is the race known as the Dosznti. There is only one Dosznti ship in this sector, and it is moving slowly through the cube as you play. Although you cannot destroy the ship, you can kill a number of individual Dosznti. In fact, the aim of the game is to get your 'alien kill total' to the highest point possible before your energy banks are exhausted. Although there are many, many more Dosznti in this sector than there are humans, the aliens are notoriously bad shots, so your chance of wiping out a lot before your mission is terminated is rather high.

You can only fire at the Dosznti when you are within three units of their position. Attempting to fire when they are out of range leads only to the crew response 'THE DOSZNTI SHOP IS NOT WITHIN RANGE, SIR'.

You have a limited amount of energy, and each successful hit by an alien craft diminishes your energy supply.

I will not explain the game in any more detail, because to do so would diminish the enjoyment you'll get from playing it. The 'behaviour of space' does not change from game to game, so you should find you begin to learn the tricks of space warfare after the first game, leading to higher and higher scores as you continue to play it. The game tends to explain the rules to you as you play, and the screen display has been organised to make it as clear as possible what is going on.

```
10 REM HYPERWAR
20 HS=0
30 GOSUB 1410
40 GOSUB 1150
50 IF L<0 THEN 670
60 GOSUB 1660
70 L=L-.25
80 TI=TI-1
90 PRINT "WHAT IS YOUR ORDER, CAPTAIN?"
100 PRINT "  N, S, E, W"
110 PRINT "  A(DVANCE), R(ETREAT)"
120 PRINT "  H(YERSPACE)"
130 PRINT "  L(ASER)"
140 INPUT Z$
150 IF Z$="N" THEN X=X-1
160 IF Z$="S" THEN X=X+1
170 IF Z$="E" THEN Y=Y+1
180 IF Z$="W" THEN Y=Y-1
190 IF Z$="A" THEN Z=Z-1
200 IF Z$="R" THEN Z=Z+1
210 IF Z$="L" THEN GOSUB 490
220 IF Z$="H" THEN GOSUB 1520
230 IF X<1 THEN X=1
240 IF Y<1 THEN Y=1
250 IF Z<1 THEN Z=1
260 IF X>10 THEN X=10
270 IF Y>10 THEN Y=10
280 IF Z>10 THEN Z=10
290 GOSUB 890
300 IF RND(10)<7 THEN 40
310 PRINT "*****"
320 PRINT "*DEEP SPACE SCANNERS READ:*"
330 PRINT "*  N/S -"A-X;TAB(26);"*"
340 PRINT "*  E/W -"B-Y;TAB(26);"*"
350 PRINT "*  A/R -"C-Z;TAB(26);"*"
360 PRINT "*****"
370 A=A+RND(4)-3
380 B=B+RND(4)-3
390 C=C+RND(4)-3
400 IF A<1 THEN A=1
410 IF A>10 THEN A=10
420 IF B<1 THEN B=1
430 IF B>10 THEN B=10
```

```

440 IF C<1 THEN C=1
450 IF C>10 THEN C=10
460 GOSUB 1650
470 GOTO 40
480 REM *****
490 REM LASER OPTION
500 L=L-.75
510 GOSUB 1620
520 IF ABS(A-X)>3 OR ABS(B-Y)>3 OR ABS(C-Z)>3
      THEN 1680
540 PRINT "LASERS ARMED AND READY, SIR"
550 GOSUB 1650
560 IF RND(10)>5 THEN 590
570 PRINT "LASERS FIRE WAS UNSUCCESSFUL,","CAPTAIN"
580 GOTO 650
590 PRINT "YOU DAMAGED THE DOSZNTI SHIP,","CAPTAIN"
600 FOR J=1 TO 30
610 FOR H=1 TO 30-J/2:PRINT " ";:NEXT H
620 PRINT "WELL DONE!!!";
630 NEXT J
640 T=T+RND(1000)+784
650 GOTO 1620
670 REM OUT OF ENERGY
680 GOSUB 1620
690 PRINT "THIS IS SHIP'S MASTER CONTROL"
700 GOSUB 1650
710 IF L<=0 THEN PRINT "ENERGY RESERVES DEPLETED"
720 IF TI=0 THEN PRINT "YOU HAVE STAYED IN SPACE
      TOO LONG"
730 PRINT:PRINT "YOU DEFEATED" T "DOSZNTI":GOSUB 1650
740 PRINT "LIFE SUPPORT SYSTEMS FADING..."
750 GOSUB 1650
760 PRINT " FADING..."
770 GOSUB 1650
780 PRINT " FADING..."
790 GOSUB 1650
800 PRINT "TALLY WAS " T: IF T>HS THEN HS=T
810 PRINT:PRINT "BEST SO FAR" HS:GOSUB 1650
820 PRINT:PRINT "STAND BY FOR YOUR NEXT MISSION"
830 GOSUB 1650:GOSUB 1650:GOTO 30
840 REM *****
850 GOSUB 1650
860 PRINT "YOU HAVE COLLIDED WITH THE","DOSZNTI SHIP!"
870 GOSUB 1650

```

```

1260 IF T>0 THEN PRINT:PRINT "DOSZNTI
      DESTRUCTION TALLY" T
1270 PRINT ">>>SHIPS GALACTIC CO-ORDINATES ARE" X; Y; Z
1280 PRINT ">>>DOSZNTI MOTHERSHIP LOCATED AT" A; B; C
1290 PRINT:PRINT "THE DOSZNTI IS ";
1300 IF A<>X OR B<>Y THEN PRINT "TO THE";
1310 IF A<X THEN PRINT " NORTH";
1320 IF A>X THEN PRINT " SOUTH";
1330 IF B>Y THEN PRINT " EAST";
1340 IF B<Y THEN PRINT " WEST";
1350 IF C=Z THEN PRINT " OF YOUR SHIP, SIR"
1360 IF C>Z THEN PRINT " BEHIND US, CAPTAIN"
1370 IF C<Z THEN PRINT " IN FRONT OF US, SIR"
1380 PRINT "*****"
1390 GOTO 1650
1400 REM *****
1410 REM INITIALISE
1420 CLS
1440 L=35*RND(30):T=0:TI=50
1450 A=RND(10)
1460 B=RND(10)
1470 C=RND(10)
1480 X=RND(10)
1490 Y=RND(10)
1500 Z=RND(10)
1510 RETURN
1520 REM HYPERSPACE OPTION
1530 X=RND(10)
1540 Y=RND(10)
1550 Z=RND(10)
1560 FOR J=1 TO 40
1570 PRINT TAB(J);"*":PRINT
1580 FOR H=1 TO J:NEXT H
1590 NEXT J
1600 CLS
1610 RETURN
1620 FOR P=1 TO 1000:NEXT P
1630 CLS
1640 GOTO 1660
1650 FOR P=1 TO 1000:NEXT P
1660 PRINT:PRINT
1670 RETURN

```

```
168 PRINT "THE DOSZNTI SHIP IS NOT  
          WITHIN RANGE, SIR"
```

```
1690 GOTO 1650
```

```
1700 GOSUB 1620:PRINT "CONDITION AMBER!":  
          SOUND 24,8:GOTO 1650
```

```
*****  
      *** ** ** ***  
**** * * **** * * ****  
      * * * * * * * *  
** * ** * * ** * * ** * **  
      ** * * * * * * **  
* * * * * * * * * * * *  
***** * * * * * *****  
***** * * * * * *****  
* * * * * * * * * * * *  
      ** * * * * * * **  
** * ** * * ** * * ** * **  
      * * * * * * * *  
**** * * **** * * ****  
      *** ** ** ***  
*****  
*****
```

# Brain Games



# Brain Games

I guess 90% of all games could, without stretching the definition too much, be called 'brain games'. After all, they demand a certain amount of brain-power and application from you to play.

However, the programs in this section under the heading of BRAIN GAMES are here because they demand a little more intellectual effort than several of the other games in the book.

Here you'll tax your ability as you solve magic stars in IDAHO STARS and save your precious neck by deducing a word the computer has chosen in EXECUTIONER.

As well, you'll try to unlock a twisted cube in CUBIC, will settle down to a game or two of HUMBLE MARBLE (generally known as a "Solitaire" and named, in this case, because the game in 'real life' uses marbles on a board) and will have to work under the pressure of a relentless clock to locate some atoms within a CYCLOTRON.

If your brain still has any power left after all that, you can attempt a task which may appear impossible in FLIPPER.

# IDAHO STARS

IDAHO STARS are stars with a special property -- numbers placed on the cross-over points of the lines forming them will, when added together, produce the same total, no matter which line of numbers is chosen.

Here, for example, is one such star:



As you can see, each line on the star sums to the same total. In this program, the computer generates an IDAHO STAR, leaving two or three of the numbers as zeroes for you to work out as quickly as you can.

```

10 REM IDAHO STARS
20 GOSUB 480:REM SET UP STAR
30 GOSUB 340:REM PRINT STAR
40 GOSUB 70:REM ASK FOR GUESS
50 GOTO 30
60 REM *****
70 REM ASK FOR GUESS
80 IF SC>0 AND SC<10 THEN PRINT
      "YOU HAVE"SC"RIGHT SO FAR"
85 IF NOT (SC>0 AND SC<10) THEN PRINT
90 GO=GO+1
100 PRINT "THIS IS GO NUMBER"GO
110 PRINT "ENTER ANY NUMBER YOU"
120 INPUT "THINK IS PART OF THE STAR";G
130 SC=0

```

```

140 FOR J=1 TO 10
150 IF G=A(J) THEN B(J)=A(J)
160 IF B(J)<>0 THEN SC=SC+1
170 NEXT J
180 FOR P=1 TO 1000:NEXT P
190 IF SC<10 THEN RETURN
200 REM *****
210 GOSUB 340
220 PRINT "YES, YOU'VE SOLVED IT! "
230 PRINT "AND IT ONLY TOOK"GO"GOES..."
235 IF GO<10 THEN GOSUB 800
250 PRINT "PRESS 'Y' IF YOU'D LIKE TO"
260 PRINT "TRY ANOTHER STAR OF IDAHO, "
270 PRINT "PRESS 'N' TO STOP....."
280 A$=INKEY$
290 IF A$<>"N" AND A$<>"Y" THEN 280
300 IF A$="Y" THEN RUN
310 PRINT:PRINT "OK, THANKS FOR PLAYING"
320 END
330 REM *****
340 REM PRINT STAR
350 PRINT CHR$(28);
360 PRINT TAB(11);B(1)
370 PRINT:PRINT
380 PRINT TAB(2);" ";B(2);" ";B(3);" ";B(4);" ";B(5)
390 PRINT
400 PRINT TAB(8);B(6);" ";B(7)
420 PRINT TAB(11);B(8)
430 PRINT
440 PRINT TAB(5);B(9);" ";B(10)
460 RETURN
470 REM *****
480 REM SET UP STAR
490 CLS
500 DIM A(10),B(10)
510 GO=0:SC=0
520 A=RND(9)
530 B=RND(9)
540 C=RND(9)
550 D=RND(9)
560 E=RND(9)
570 IF A=B OR A=C OR A=D OR A=E THEN 530
580 IF B=C OR B=D OR B=E THEN 530
590 IF C=D OR C=E THEN 530

```

```

600 IF D=E THEN 550
610 X=RND(3)
620 A(1)=X
630 A(2)=X-B+C+D
640 A(3)=A+E
650 A(4)=A+D
660 A(5)=X-B-C+E
670 A(6)=A
680 A(7)=A+C
690 A(8)=A+B
700 A(9)=X-2*B+2*D+E
710 A(10)=X-2*B-C+D+2*E
720 FOR J=1 TO 10
730 B(J)=A(J)
740 IF A(J)=0 THEN RUN
750 NEXT J
760 B(RND(10))=0
770 B(RND(10))=0
780 B(RND(10))=0
790 RETURN
800 FOR TU=1 TO 9:READ FR,DU:SOUND FR,DU:NEXT TU:RETU
810 DATA 16,3,18,2,20,3,20,1,20,2,18,1,20,2,21,5,20,3

```

# EXECUTIONER

As you can tell, this game is a computer variation of the old pencil-and-paper game HANGMAN. In this game, the computer chooses a word from its store (held in the DATA statements from 500 to 740) and then challenges you to guess it.

The number of guesses you'll get is related to the length of the word (see lines 120 and 170).

```

10 REM EXECUTIONER
20 CLS:RESTORE
30 Y=0:L$=""
40 PRINT"PRESS 'E' WHEN YOU'RE READY"
50 PRINT"TO FACE THE EXECUTIONER..."
60 IF INKEY#<>"E" THEN 60
80 FOR G=1 TO RND(125)
90 READ A#
100 NEXT G
110 GOSUB 480
120 N=LEN(A#):DIM B(N),D(N),O$(26)
130 FOR G=1 TO N
140 B(G)=ASC(MID$(A#,G,1))
150 D(G)=B(G)
160 NEXT G
170 Q=INT(N+N/2+.5)
180 CLS:PRINT:PRINT"YOU HAVE TO GUESS THE EXECUTION-"
190 PRINT"ER'S WORD IN JUST"Q"GUESSES"
200 GOSUB 480:GOSUB 480:GOSUB 480:CLS
210 FOR J=1 TO Q:Y=Y+1
220 GOSUB 400
230 IF H=N THEN 340
240 PRINT:PRINT:PRINT Q+1-J"CHANCES LEFT....":O$=INKE
250 PRINT:PRINT"ENTER YOUR NEXT GUESS? ";
255 C#=INKEY#:IF C#="" THEN 255 ELSE PRINT C#:
FOR I=1 TO 100:NEX
260 F=ASC(C#)
265 O$(F-64)=CHR$(F)
270 FOR G=1 TO N
280 IF D(G)=F THEN D(G)=0:J=J-1
290 NEXT G:NEXT J

```

```

300 GOSUB 480
310 GOSUB 400
320 PRINT:PRINT:PRINT"SO SORRY,BUT YOU GOTTA GO NOW!"
      :SOUND 1,8
330 PRINT"THE EXECUTIONER'S WORD WAS ";A$:L$
      :GOSUB 480:END
340 PRINT:PRINT"WHEW! YOU'VE STAVED OFF"
350 PRINT"EXECUTION FOR ANOTHER DAY."
360 PRINT"YOU GOT IT IN"Y-1"GUESSES"
370 PRINT:PRINT"THE EXECUTIONER'S WORD WAS":PRINTA$
375 FOR I=1 TO 20:FOR K=1 TO 200:NEXT:COLOR,1:
      FOR K=1 TO 200:NEXT
376 COLOR,0:NEXT
380 PRINT"DO YOU WANT ANOTHER GO?"
385 GOSUB 495:IF A$<>"N" AND A$<>"Y" THEN 385
390 IF A$="N" THEN END ELSE RUN
400 H=0:PRINT CHR$(28)
405 FOR V=1 TO 26:PRINT@V+2,0$(V):NEXT V:PRINT:PRINT
410 FOR E=1 TO N
420 IF B(E)=D(E) THEN PRINT"-";
430 IF B(E)<>D(E) THEN PRINT CHR$(B(E));:H=H+1
440 NEXT E
450 PRINT:PRINT:IF H<>N THEN PRINT"YOU HAVE"H
      "LETTERS CORRECT"
460 PRINT
470 RETURN
480 FOR O=1 TO 500:NEXT O
490 RETURN
495 A$=INKEY$:A$=INKEY$:RETURN
500 DATA"MERIDIAN","MERIT","MERMAID","MERRIMENT"
510 DATA"OVERSEER","OXIDANT","OXYGEN","PALPABLE"
      ,"UNORTHODOX"
520 DATA"PANDEMONIUM","PANEGRIC","PARADOXICAL"
      ,"PHEASANT","FLY"
530 DATA"RUMPUS","RUMMAGE","SACRAMENT","SABRE"
      ,"SCHEMATIC","TAR"
540 DATA"SEDIMENT","SEXAGENARIAN","TEMPERATE"
      ,"TELESCOPE","CAR"
550 DATA"GYPSY","RHAPSODY","SCALENE","SATELLITE"
      ,"MARSHMALLOW"
560 DATA"HELICOPTER","CASSEROLE","ILLUSTRATE"
      ,"SYMPATHY","MOUSE"

```

```

570 DATA"REFRIGERATE","MAMMOTH","KALEIDOSCOPE"
      ,"DUPLICATE","CAT"
580 DATA"INSATIABLE","SILHOUETTE","BONDAGE"
      ,"BONUS","DECIMAL"
590 DATA"DEAFEN","THEORY","THISTLE","SABOTAGE"
      ,"COURTESY","TRY"
600 DATA"DEBUTANTE","DECADENCE","KETCHUP"
      ,"KINDERGARTEN","METRE"
610 DATA"SUPERCALIFRAGILISTICEXPIALIDOCIOUS",
      "COMPUTER","HOUSE"
620 DATA"OLYMPIC","OMNIBUS","OMNIPOTENT"
      ,"MAGNANIMOUS","ZEALOUS"
630 DATA"ZOOLOGY","ZILLION","XEROX","XYLOPHONE"
      ,"QUINCE","YUMMY"
640 "FATIGUE","FALSETTO","EQUIVALENT","ENZYME"
      ,"DISRUPT","HEART"
650 DATA"QUEASY","IMBECILE","BIENNIAL"
      ,"ACCOMPLISHMENT","TONGUE"
660 DATA"KUNGFU","JAVELIN","DEMANDMANT"
      ,"ODMIK","JERKIN","QUAD"
670 DATA"ZEBRA","TOMAHAWK","MARZIPAN"
      ,"PULVERIZE","FACSIMILE"
680 DATA"HYPOCHONDRIAC","RECRUIT"
      ,"CONJURE","STEWARDESS","DEBUT"
690 DATA"LIZARD","SCHOLASTIC","THEOLOGY"
      ,"GEOMETRY","GIBBERISH"
700 DATA"VEGETABLE","WINTERGREEN","YELLOW"
      ,"YEARLING","INFINITY"
710 DATA"PRAIRIE","SCHEDULED","RHEUMATISM"
      ,"MOSQUITO","DOG","TAB"
720 DATA"BOX","JET","QUIP","QUIZ","QUACK"
      ,"AXE","BIB","FIB","WET"
730 DATA"YACHT","VERSUS","PRECIS","VILIFY"
      ,"WOMANHOOD","WOOD"
740 DATA"YOU","WOODWIND","ACACIA"

```

# CUBIC

This is a two-dimensional version of the colored cube puzzle which swept the world from Hungary.

The 'cube' starts off looking like this:

```
1 1 2 2
1 1 2 2
3 3 4 4
3 3 4 4
```

After some manipulation by the computer, it will look something like this:

```
1 2 1 2
3 3 3 2
3 1 4 1
4 2 4 4
```

Your job is to get it back into its original state in as few moves as possible. The numbers on the cube rotate in groups of four, spinning in a clockwise direction. This means

```
1 2
3 4 when rotated will look
```

like this

```
3 1
4 2
```

You enter a number in order to rotate the elements of the cube. The key to move is as follows:

```
2 3 4
6 7 8
10 11 12
```

```
30 DIM A(4),B(16)
40 CLS
50 B=1:F=0
60 FOR D=1 TO 4
70 C=0
80 IF B=1 THEN C=1
90 IF B=3 THEN C=2
100 IF B=9 THEN C=3
110 IF B=11 THEN C=4
120 B(B)=C
130 B(B+1)=C
140 B(B+4)=C
150 B(B+5)=C
160 C=0
170 IF B=1 THEN C=2
180 IF B=3 THEN C=6
190 IF B=9 THEN C=2
200 B=B+C
210 NEXT D
220 GOSUB 410
230 PRINT:PRINT
240 IF F<11 THEN PRINT@224,"TWISTING..."
250 F=F+1
260 FOR P=1 TO 500:NEXT P
270 IF F<11 THEN X=RND(12)
280 IF F>10 THEN GOSUB 590
290 IF X<2 OR X=5 OR X=9 OR X>12 THEN 270
300 A(1)=B(X)
310 A(2)=B(X+4)
320 A(3)=B(X+3)
330 A(4)=B(X-1)
340 B(X)=A(4)
350 B(X+4)=A(1)
360 B(X+3)=A(2)
370 B(X-1)=A(3)
380 GOSUB 410
390 F=F+1
400 GOTO 270
410 PRINT CHR$(28)
420 PRINT:PRINT
430 FOR B=1 TO 16
440 PRINT B(B);
450 IF B/4=INT(B/4) THEN PRINT
460 NEXT B
```



```

470 PRINT:PRINT
480 IF F<10 THEN PRINT@0," TWIST NUMBER"F
490 IF F>10 THEN PRINT@0," YOUR CUBIC NUMBER"F-10
500 IF F<11 THEN RETURN
510 P=0
520 IF B(1)=1 AND B(2)=1 AND B(3)=2 AND B(4)=2
      THEN P=P+1
530 IF B(5)=1 AND B(6)=1 AND B(7)=2 AND B(8)=2
      THEN P=P+1
540 IF B(9)=3 AND B(10)=3 AND B(11)=4 AND B(12)=4
      THEN P=P+1
550 IF B(13)=3 AND B(14)=3 AND B(15)=4 AND B(16)=4
      THEN P=P+1
560 IF P<>4 THEN RETURN
570 PRINT@384,"YOU SOLVED IT IN JUST"F-10"TWISTS"
580 PRINT "WELL DONE, CUBIST!":END
590 Z#=INKEY#:Z#=INKEY#
600 PRINT@320,"ENTER YOUR CHOICE (1 - 9)"
610 L#=INKEY#:X=VAL(L#)
620 IF X<1 OR X>9 THEN 610
630 SOUND 31,1:IF X<4 THEN X=X+1:RETURN
640 IF X<7 THEN X=X+2:RETURN
650 X=X+3:RETURN
660 RETURN

```

# MUMBLE MARBLE

Our next brain game is a computer version of the game you probably know as 'Solitaire' in which you have to move marbles around a board. At the start of the game only the center hole is empty. You can jump over marbles into empty holes, and you remove the marble so jumped over. The idea of the game is to end up with just one marble in the center hole.

Many stories have been told about the origin of this game. The most interesting of these tales is the one that says the idea of the game was worked out by a prisoner in solitary confinement in the Bastille, who devised it as a way to relieve the monotony of his imprisonment. Whether that story is true or not, there is no doubt that the game can become quite addictive, as you try to devise a foolproof way to solve it.

Here's what the MUMBLE MARBLE display looks like at the start of the game:

	1	2	3	4	5	6	7	
			0	0	0			1
			0	0	0			2
	0	0	0	0	0	0	0	3
	0	0	0	*	0	0	0	4
	0	0	0	0	0	0	0	5
			0	0	0			6
			0	0	0			7

A few moves later it could look like this:

	1	2	3	4	5	6	7	
			0	0	0			1
			0	*	0			2
0	*	0	0	0	0	0	0	3
0	0	*	0	0	0	0	0	4
0	0	0	0	0	0	0	0	5
			*	0	0			6
			*	0	0			7

```

10 REM MUMBLE MARBLE
20 GOSUB 400
30 GOSUB 250
40 REM ACCEPT MOVE
50 PRINT@416,"WHICH MARBLE DO YOU WANT TO MOVE";
60 PRINT@448,"";:INPUT A
70 IF A=99 THEN GOTO 240
80 IF A<11 OR A>77 THEN GOTO 60
90 IF A(A)<>79 THEN GOTO 60
100 PRINT@448,A"TO WHERE";
110 INPUT B
120 IF B<11 OR B>77 THEN GOTO 100
130 IF A(B)<>E THEN GOTO 100
140 A((A+B)/2)=E:A(A)=E:A(B)=79
150 MOVE=MOVE+1
160 COUNT=0
170 FOR F=11 TO 75
180 IF A(F)=79 THEN COUNT=COUNT+1
190 NEXT F
200 GOSUB 250
210 PRINT "THERE ARE";COUNT;"MARBLES ";
      PRINT "ON THE BOARD"

220 IF COUNT<>1 THEN GOTO 40
230 IFA(44)=79THENPRINT"YOU DID IT, IN
      JUST";MOVE;"MOVES! ";:END

240 PRINT "THE GAME IS OVER AND YOU'VE FAILED!":END
250 REM PRINT OUT
260 PRINT@448,L#:PRINTCHR$(28);
270 PRINT "ENTER SIDE CO-ORDINATE FIRST"
280 PRINT TAB(5);"ENTER 99 TO CONCEDE"
290 PRINT "      1 2 3 4 5 6 7"
300 PRINT TAB(5);
310 FOR D=11 TO 75
320 T=10*(INT(D/10))
330 IF D-T=8 THEN D=D+2:PRINT T/10:PRINT
      TAB(5);:GOTO 350

340 PRINT CHR$(A(D));" ";
350 NEXT D:PRINT "      7"
370 PRINT "MOVES SO FAR:";MOVE
390 RETURN
400 REM INITIALISE
410 CLS
420 DIM A(87)

```

```

430 E=42:L$=""

```

```

":
REM 31 SPACES

```

```

440 FOR D=11 TO 75
450 T=10*(INT(D/10))
460 IF D-T=8 THEN D=D+3
470 READ A(D)
480 NEXT D
490 MOVE=0
500 RETURN
510 REM 42 IS ASC ("*")
520 REM 79 IS ASC ("D")
530 DATA 32,32,79,79,79,32,32
540 DATA 32,32,79,79,79,32,32
550 DATA 79,79,79,79,79,79,79
560 DATA 79,79,79,42,79,79,79
570 DATA 79,79,79,79,79,79,79
580 DATA 32,32,79,79,79,32,32
590 DATA 32,32,79,79,79

```

# CYCLOTRON

In CYCLOTRON, you fire atoms into the machine from the top, by specifying a number, and you have to try and locate particles of antimatter hidden within the machine from the behavior of the atom.

If the atom hits a particle of antimatter it will either be absorbed, or deflected to the right or left. An atom may be effected by more than one particle of antimatter which makes the resolution of the problem even more difficult.

There are always three particles of antimatter in the machine, although they could well all be in the same number cyclotron. You have an extremely limited number of goes in order to locate the antimatter. At each go, you can either enter the number where you want to enter an atom, or be reckless and enter 8, which gives you the chance to guess the location of the antimatter particles.

```
10 REM CYCLOTRON
20 CLS
30 N=0:HS=9:DIM B(7),Q(3),W(3)
80 REM PLACE ANTIMATTER
90 FOR A=1 TO 3
100 B(A)=RND(7)
110 NEXT A
120 FOR H=6 TO 1 STEP -1
130 CLS
140 PRINT:PRINT:PRINT"YOU HAVE"H"SECONDS"
150 PRINT"ENTER CYCLOTRON NUMBER,"
160 PRINT"OR 8 IF YOU THINK YOU KNOW"
170 PRINT"WHERE THE ANTIMATTER LIES"
180 INPUT C
190 IF C=8 THEN 520
200 GOSUB 450
210 D=1
220 IF B(D)=C THEN 350
230 IF D<3 THEN D=D+1:GOTO 220
240 GOSUB 500
```

```
250 PRINT"THE ATOM EMERGED FROM"C:SOUND 31,5
260 FOR Z=1 TO 1000:NEXT Z
270 NEXT H
280 PRINT:PRINT:PRINT
290 PRINT"SORRY,TIME IS UP"
300 PRINT:PRINT"THE ANTIMATTER WAS IN":SOUND 6,7:
SOUND 4,8
310 FOR A=1 TO 3
320 PRINT B(A);
330 NEXT A
340 END
350 REM ANTIMATTER ACTS
360 IF RND(10)>9 THEN 420
370 IF C=1 THEN C=2:GOTO 210
380 IF C=7 THEN C=6:GOTO 210
390 F=RND(2)-1
400 C=C+F
410 GOTO 210
420 REM ANTIMATTER ABSORBS ATOM
430 PRINT"THE ATOM HAS BEEN ABSORBED":SOUND 31,1:
SOUND 30,1
440 GOTO 260
450 PRINT:PRINT:PRINT TAB(3+2*C);C
460 FOR Z=1 TO 5
470 PRINT TAB(6);"- - - - -"
480 NEXT Z
490 RETURN
500 PRINT TAB(3+2*C);C
510 RETURN
520 REM LOCATION ATTEMPT
530 PRINT"WHERE DO YOU THINK"
540 PRINT"THE ANTIMATTER LIES?"
550 INPUT Q(1),Q(2),Q(3)
560 FOR T=1 TO 3:W(T)=B(T):NEXT T
570 SR=0
580 FOR X=1 TO 3:FOR Y=1 TO 3
590 IF Q(X)=B(Y) THEN SR=SR+1:B(Y)=0:Q(X)=0
600 NEXT Y,X
610 GOTO 650
620 FOR P=1 TO 3
630 B(P)=W(P)
640 NEXT P
650 IF SR>2 THEN 700
```

```

660 IF SR>0 THEN PRINT"YOU LOCATED"SR"CORRECTLY"
670 IF SR=0 THEN PRINT"YOU ARE COMPLETELY WRONG!"
680 FOR T=1 TO 3:B(T)=W(T):NEXT T
690 GOTO 260
700 PRINT"WELL DONE. YOU LOCATED THE"
710 PRINT"ANTIMATTER WITH"H"SECONDS"
720 PRINT,"TO SPARE":GOSUB 770
730 IF 6-H<HS THEN HS=6-H
740 PRINT:PRINT"YOUR BEST TIME SO FAR IS"HS
750 FOR Z=1 TO 1000:NEXT Z
760 GOTO 90
770 SOUND 21,4:SOUND 16,2:SOUND 16,1:SOUND 18,4:
      SOUND 16,4
780 :SOUND 0,1:SOUND 20,4:SOUND 21,4:RETURN

```

# FLIPPER

This intriguing game can provide you with a considerable degree of mental stretching. When you run the game, you'll see a mix of X's and \*'s on a three by three grid. You have to end up with an X in the middle, and eight \*'s surrounding it.

You enter the number of the place you want to hit with your flipper. Flipping a corner piece causes those adjoining it to change into their opposites (that is an X becomes an \*, and an \* becomes an X). Hitting the middle piece on a side with your flipper causes the two on either side of it to change, and hitting the middle piece changes the middle pieces on each of the sides.

The piece you hit always changes in each case.

```

10 REM FLIPPER
20 DIM A(10),F(4):CLS
30 M=-1:Q=42:X=88:P=0
50 FOR C=1 TO 9
60 A(C)=Q
70 IF INT(RND(100)/100+.5)=0 THEN A(C)=X
80 NEXT C
90 GOSUB 270
100 M=M+1
110 N=0
120 FOR C=1 TO 9
130 IF A(C)=X THEN N=N+1
140 NEXT C
150 IF N=1 AND A(5)=X THEN 350
160 IF M>0 THEN PRINT:PRINT"THAT WAS MOVE" M
170 PRINT:PRINT"NUMBER OF X IS" N
190 PRINT:PRINT"WHICH ONE DO YOU WANT TO FLIP?"
200 IF INKEY#<>" THEN 200
210 A#=INKEY#
220 N=VAL(A#):IF N<1 OR N>9 THEN 210
225 IF A(VAL(A#))=88 THEN 210
230 P=N:SOUND 31,1
240 GOSUB 380
250 GOTO 90

```

```

260 END
270 PRINT CHR$(28)
280 IF P<>0 THEN PRINT"YOU FLIPPED"P ELSE PRINT
290 PRINT:PRINT"1 2 3",CHR$(A(1));" ";CHR$(A(2));" "
295 PRINT CHR$(A(3))
300 PRINT
310 PRINT"4 5 6",CHR$(A(4));" ";CHR$(A(5));
      " "CHR$(A(6))

320 PRINT
330 PRINT"7 8 9",CHR$(A(7));" ";CHR$(A(8));
      " ";CHR$(A(9))

340 RETURN
350 PRINT:PRINT
360 PRINT:PRINT:PRINT"YOU SOLVED IT IN JUST"M"MOVES"
370 END
380 IF A(N)=X THEN RETURN
390 IF N=1 THEN F(1)=2:F(2)=4:F(3)=5:F(4)=10
400 IF N=2 THEN F(1)=1:F(2)=3:F(3)=10:F(4)=10
410 IF N=3 THEN F(1)=2:F(2)=5:F(3)=6:F(4)=10
420 IF N=4 THEN F(1)=1:F(2)=7:F(3)=10:F(4)=10
430 IF N=5 THEN F(1)=2:F(2)=4:F(3)=8:F(4)=6
440 IF N=6 THEN F(1)=3:F(2)=9:F(3)=10:F(4)=10
450 IF N=7 THEN F(1)=4:F(2)=5:F(3)=8:F(4)=10
460 IF N=8 THEN F(1)=7:F(2)=9:F(3)=10:F(4)=10
470 IF N=9 THEN F(1)=8:F(2)=5:F(3)=6:F(4)=10
480 FOR G=1 TO 4
490 F=0
500 IF A(F(G))=X THEN F=1
510 IF F=1 THEN A(F(G))=0
520 IF F=0 AND A(F(G))=0 THEN A(F(G))=X
530 NEXT G
540 A(N)=X
550 RETURN

```

# Creating your own Games



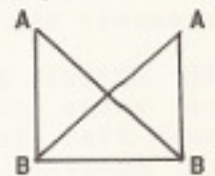
# Creating Games

You may well find, after you've been programming for some time, that although you've got many of the skills you need to write games of your own, you're a little short on ideas. If that's the case, you should find this section of the book of interest to you. I'm going to outline a number of games which seem to me to be ideal for conversion to computer games.

## GAMES TO ADAPT

**SHUFFLEBOARD:** Players slide flat, circular pieces along a board towards a triangular 'target' which contains painted circles with numbers on them. A piece ending up entirely within the numbered circle gives the player the score of that number. There are many possible variations of this, including darts-like target games. Once you get a shuffleboard program running, you may well find that only cosmetic changes are needed to convert it into a ten-pin bowling game.

**HORSESHOE:** Starting with a board like this, with one player's pieces at the positions marked A, and one at those marked B, the players take it in turns to move along a line to a vacant spot:



The game ends when a player discovers he or she (or it) cannot move. Despite its simplicity, this is an interesting game, as you'll discover when you play it.

**MU-TORERE:** This game, which originated among the Maoris in New Zealand, is played on a board shaped like an eight-pointed star, with each point of the star joined to a circle in the middle. This circle

is called the putahi. The players start with four pieces each, which are placed on the outside points, with all of one player's pieces occupying points next to each other. The aim of the game, as in the last two we've discussed, is to make it impossible for a player to move. You can move from one point to an adjacent empty point, or from the putahi to any vacant point. The third possible move is from a point to the putahi, but this move can only be made if at least one of the points immediately adjacent to the one from which you intend to move is occupied by an opponent's piece.

**BINGO:** This game is usually played with cards marked with numbers, which are crossed off as numbers are drawn at random from a pool, the winner being the player who first crosses off five in a row on their card, in any direction. It can be greatly simplified for computer use, with the computer generating two 'cards' at random, using numbers from zero to 99, and printing them on a five by five grid. The computer can then choose numbers at random from zero to 99, and do the crossings off, and look for possible wins. The program can be made more interesting to look at by, instead of just generating a random number, including two 'dials' with spinning arrows. The first dial, the 'tens' is marked from zero to nine, and this is spun first. The second dial is marked the same as the first one, except that it represents the 'ones'.

**DICE GAMES:** As I said in the notes regarding the dice games in the dice section of this book, you should find dice games relatively easy to program on your computer, especially if the computer is not expected to exhibit too much intelligence when playing. There are a number of dice games which you may wish to try and adapt for computer play, including:

**DROP DEAD:** You need five dice for this. The dice are thrown by the first player, and any which show two or five are removed. The total of the rest is written down, and the dice which fell with a two or

a five are thrown again. After this throw, the dice landing two and five are removed, and the total of the remaining dice is added to the total from the first throw. This process continues until there are no dice left, at which point that particular player's move ends. It is then player two's go, and the same procedure is followed. The winner is the player with the biggest total after both have had their throws. You can make the game the best of, say, three rounds, or limit the total number of throws any player can have per go, regardless of how many dice he or she has left.

**MARTINETTI:** You need three dice for this, and a score sheet marked with a row of boxes marked from one to 12, and two tokens, one each for you and the computer. You are aiming to traverse the numbers in order, before your opponent manages to do so. Each player takes it in turn to throw the dice. If the throw contains a one, the player's token can be placed in the one square. If the first throw also contained a two, the token could be moved into the two. After box one, players can either take the pip shown on an individual die for a move, or can add together the pips showing to get a desired number. A variation of this game (often called EVEREST) is to 'climb down' through the numbers, after one through to 12 has been achieved. The winner of EVEREST is the first player getting back to base camp on one.

**INDIAN DICE:** This game is similar to POKER DICE and is played with five dice. The aim is simply to get the best possible 'poker hands'. There is no provision for getting additional 'cards' in one deal. You throw the five dice at once, and that is your hand. The ranking of the hands is as follows [with six highest and ones wild]:

- five of a kind
- four of a kind
- full house (three of a kind plus a pair)
- two pairs
- one pair
- highest die

BIDOU: This game is also similar to POKER DICE, however, you aim to get rid of all your chips, rather than win them. The winner of the game is paradoxically, the person who has lost the most rounds. You need three dice for this game, and a supply of chips which are distributed equally between the two players. A bet is made, and matched, before the players take it in turns to throw the three dice. The throws are ranked as follows, with 'winning throws' coming first:

2 1 1; 2 2 1; 4 2 1; 6 6 6; 5 5 5; 4 4 4;  
3 3 3; 2 2 2; 1 1 1; 3 3 6; 3 3 5; 3 3 4;  
3 3 2; 3 3 1; 1 1 6; 1 1 5; 1 1 4; 1 1 3;  
3 2 1; 4 3 2; 5 4 3; 6 5 4

If the combination thrown is not in the list, the pips are added, with the highest total winning.

BARBOOTH: Also known as BARBUDI, this game is played with two dice, and features winning and losing throws. It is suggested you start with a bank of 30 chips, and give both players a starting stake of 15 chips each, charging two chips per player per round. The players take it in turns to throw the dice. Winning throws are 3 3; 5 5; 6 5; and 6 6. Losing throws are 1 1; 2 2; 2 1 and 4 4. If the players both get winning throws, the one closest to the start of the list wins (that is, 3 3 is better than 6 6). If both throw the same winning throw, the bank gets the money. If one player throws a combination which is not a winning one nor a losing one, and the other throws a losing throw, the loser's chips go into the bank, and the other player retains his or her chips.

If one player throws a winning throw, and the other a non-ranking combination, the winner gets his own chips back, as well as those bet by the loser. If, however, one player throws a winning combination,

and the other a losing combination, the bank contributes the same number of chips as have been bet in total. The game continues until either player, or the bank, runs out of chips. The player holding the maximum number of chips at this point wins. Note that this way of distributing the wins and losses is not the same as BARBOOTH when played as a 'professional' gambling game, but is a pay-off scheme which is relatively easy to computerise, and makes for an interesting game.

FIVE THOUSAND: There are no prizes for guessing that the first player to score over 5000 points in this game is the winner. Played with as many players as you like, all you'll need in the way of game equipment is five six-sided dice. Each player takes it in turn to throw the five dice, and scoring is carried out in the following way:

- a 5 is worth 100
- a 2 is worth 50
- three 5's thrown in one go are worth 1000
- three 2's thrown in a go score 500
- three of a kind (apart from 2's and 5's, are worth 100 times their value (that is three 1's are worth 100 points, and three 6's are worth 600
- If the sequence 1, 2, 3, 4 and 5 is thrown in a single go, this scores 1500
- 2, 3, 4, 5 and 6 in one throw are worth 2000 points

Note that 4's and 6's do not score in this game, except as part of a sequence. Once you've thrown, you can -- if you like -- throw the non-scoring dice again. However, if the next throw gains no score, the turn ends, and the overall score for the throw is zero. So long as a scoring throw has been made, any dice 'left over' can be thrown.

The game rewards the slow and careful player with a reasonable score, but gives him or her no chance against a player who decides to take a chance and try to get a massive score in a single throw.



**CARD GAMES:** Once you've worked out a routine to get the computer hold, shuffle and deal the cards (and such a routine is in my version of Concentration in this book) you have the raw bones of a host of games. Any book of card games will give you more ideas than you can possibly cope with. Here are a few to start you off:

**TRENTE AND QUARANTE:** This game, which originated in 17th century Europe, is also known as ROUGE ET NOIR. Six decks of cards are used in the casino version but it works well with a single deck. The cards are dealt into two rows, with the total of each row being examined after each row has had an additional card. Court cards (jack, queen, king) count as ten with all other cards (including the ace) counting as their face value. The moment the total of a row equals or exceeds 31, no more cards are added to that row. However, the dealer keeps adding to the other row until it, too, equals or exceeds 31. The first row is called 'black', regardless of the suit of the cards involved, and the second row is 'red'.

You can bet that, at the end, the red row or the black row total will be closer to 31. You can also bet that the colour of the first card to be dealt will match the colour name of the winning row. All bets are made against the bank, and winners receive their bet back, plus that amount again (odds of one to one). If both rows equal 31, the bank takes half the stake and returns the rest to the player.

**CARD CRAPS:** This game was devised to get around laws passed by some US States which outlawed craps played with dice. You use a special deck of 48 cards, made up from the aces, 2's, 3's, 4's, 5's and 6's, from two decks of cards. The game follows the standard rules of craps (which are similar to those used in the game SEVEN/ELEVEN in this book).

In its simplified form, without the somewhat complex betting combinations, you play as follows:

- a total of 2, 3 or 12 is 'craps' and the dealer loses, passing the cards to the next player to deal
- a total of 7 or 11 is a 'natural' and the dealer wins
- any other number (4, 5, 6, 8, 9 or 10) is the player's 'point' and he or she must continue dealing in pairs until the point is dealt again, at which point the dealer wins. However if a 7 is dealt before the point is obtained, the dealer loses

**THREE CARD BRAG:** This game was the forerunner of poker, and as its name suggests, it is played with hands of three, rather than five, cards. Here are the winning combinations, ranked from highest to lowest:

- 'pryle', three of a kind
- 'on a bike', three in sequence from the same suit
- 'run', three cards in sequence
- 'flush', three cards from the same suit
- 'pair', two of a kind
- 'high card'

A normal deck is used, and aces rank high, except for 'on a bike' and 'run', when 3, 2, ace beats ace, king, queen. You can vary the game by making it SEVEN CARD BRAG, in which the players split their seven cards into two three-card hands of their choice, or NINE CARD BRAG, when three cards are created.

**ACE-DEUCE-JACK:** This is a simple game, heavily loaded in the dealer's favor, in which players bet on the likelihood that the next card which appears will be an ace, a two (a 'deuce') or a jack.

**PUT AND TAKE:** This gambling game uses an eight-sided top, which is spun. The game begins with all players having 10 chips, with 30 in the pot. The players take it in turn to spin the top, and -- obeying the instructions which come up on the top -- put chips into the pot, or take them out. A player is out when he or she runs out of chips, and the game ends when the pot is empty. The instructions written on the sides of the top are:

- PUT ONE
- TAKE ONE
- PUT THREE
- TAKE THREE
- PUT FOUR
- TAKE FOUR
- PUT ALL
- TAKE ALL

**BOULE:** This is a simplified form of roulette which appeared in the 18th century. It is often found in European casinos. The wheel is divided into 18 segments, numbered from one to nine...twice. You can bet on the winning number, whether it will be a red or a black number, even or odd, high or low. Except for the fives on the wheel, which are usually yellow, the reds and blacks alternate, so that there is, for example, a red six and a black six, a red nine and a black nine, and so on. A 'high' number is above five, while 'low' numbers are below it. If you bet successfully on a single number, you'll be paid at seven to one, with a one to one return on the 'characteristic' of a number (such as it being 'high' or 'odd').

## BOOKS TO READ

As you've probably gathered by reading through this book, I make a point of reading any book on games I can get my hand on. It doesn't matter whether the book is about computer games or the old-fashioned 'play against another human' type. So long as it is a book of games, I want to see it.

Games books are superb as idea-starters. Read about a game and you've got the seed of a program. And it doesn't matter if the program doesn't end up playing precisely the game you read about. Many great computer games started their lives as an attempt to computerise a more traditional game, and somehow got side-tracked.

I hope some of the following list of books prove as valuable to you as they have to me.

\*DISCOVERING OLD BOARD GAMES - R C Bell (Shire Publications, Aylesbury, UK, 1980)

\*DISCOVERING DICE AND DOMINOES - R C Bell (Shire Publications, Aylesbury, UK, 1980)

\*DICE GAMES NEW AND OLD - William E Tredd (The Oleander Press, London and New York, 1981)

\*GAME PLAYING WITH BASIC - Donald D Spencer (Hayden Book Co., Inc., Rochelle Park, New Jersey, 1977)

\*GAMES ANCIENT AND ORIENTAL AND HOW TO PLAY THEM - Edward Falkener (Dover Publications, Inc., New York, 1961 - a reprint of an original work published by Longmans, Green and Co. in 1892)

\*BASIC COMPUTER GAMES - Ed. David H Ahl (Creative Computing Press, Morristown, New Jersey, 1978)

\*MORE BASIC COMPUTER GAMES - Ed. David H Ahl (Creative Computing Press, Morristown, New Jersey, 1979)

\*26 BASIC PROGRAMS FOR YOUR MICRO - Derrick Dainoff  
(Newnes Technical Books, Butterworths & Co., London  
and Boston, 1982)

\*TABLE GAMES OF GEORGIAN AND VICTORIAN DAYS - F R B  
Whitehouse (Priory Press, Royston, Herts., UK, 1971)

\*PET FUN AND GAMES - Ron Jeffries and Glen Fisher  
(Osborne/McGraw-Hill, Berkeley, California, 1981)

\*COMPUTER GAMES FOR BUSINESSES, SCHOOLS AND HOMES -  
J Victor Nahigian and William S Hodges (Winthrop  
Publishers, Inc., Cambridge, Massachusetts, 1979)

\*GAMES YOU MAKE AND PLAY - Pia Heiso, Neil Lorimer  
and Nick Williams (Macdonald and Jane's, London,  
1975)

\*WHAT TO DO AFTER YOU HIT RETURN - People's  
Computer Company (Hayden Book Co., Inc., Rochelle  
Park, New Jersey, 1980)

\*EVERYMAN'S INDOOR GAMES - Gyles Brandreth (Dent  
Everyman's Library, London, Melbourne and Toronto  
1982)

\*A PLAYER'S GUIDE TO TABLE GAMES - John Jackson  
(Stackpole Books, Harrisburg, Pennsylvania, 1975)

\*GAMES FOR TWO - John Wesley (Proteus, London and  
New York, 1981)

\*THE ADDISON-WESLEY BOOK OF APPLE COMPUTER SOFTWARE  
1982 - Jeffrey Stanton and John Dickey, editors (The  
Book Company, Lawndale, California, 1982)

\*DICING WITH DRAGONS - Ian Livingstone (Routledge  
& Kegan Paul, London, Melbourne and Henley, 1982)

\*FANTASY ROLE PLAYING GAMES - J Eric Holmes (Hippo  
crene Books Inc., New York, 1981)

Here's a host of great programs, just waiting to be  
fed into your Dick Smith VZ200 Personal  
Colour Computer. No matter what kind of games you're  
interested in, you'll find them here, to entertain  
you in the weeks and months ahead. Now you can take  
full advantage of the colour, sound and memory  
available on the VZ200 to build yourself a library  
of worthwhile and challenging games programs.

Tim Hartnell went to London from Melbourne in 1977  
to work on the Australasian Express newspaper. While  
in London, he wrote his first computer book, which  
proved immediately successful. He has now written  
more than 30 books on microcomputers, and brings the  
experience gained from the books together in  
this ultimate games book for your VZ200.

Tim was joined in the writing of this book by Glen  
Pringle, a schoolboy from Lilydale, Victoria who  
completed his major contributions to the work a week  
before his fourteenth birthday. Glen intends to  
make his career in computing, probably as a  
programmer. As well as being interested in  
computers, Glen reads a lot (mostly war books) and  
is interested in international cricket.

You'll find much to entertain you in this volume,  
including board games such as GOMOKU and CHECKERS.  
As well, there are two adventure programs:  
STRONGHOLD OF THE DWARVEN LORDS and  
THE DUKE OF DRAGONFEAR. You can take part in  
simulation programs like MISTRESS OF XENOPHOBIA or  
run a company in CHAIRMAN OF THE BOARD. There is a  
generous selection of dice games, such as ONE AND  
TWENTY and CHEMIN DE COMPUTER.

If you're looking for raw entertainment, jump right  
into ROBOT MINEFIELD and INNER SPRING. Out in space,  
your VZ200 is waiting to entertain you with  
MOONLANDER and HYPERWAR.

This is a just a sample of the action waiting for  
you in this great book by Glen and Tim. Come  
on in, the games are fine!