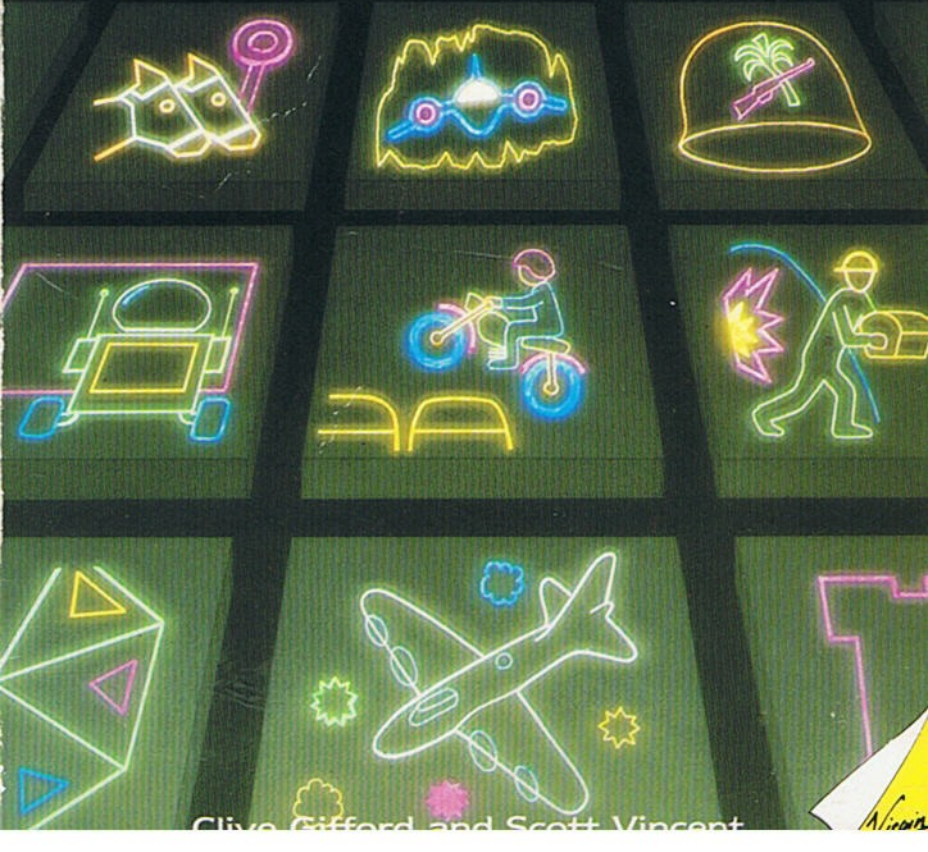
The Computer Games Series

Series Editor: Tim Hartnell

More

GAMES FOR YOUR ZX81

£££££'s of entertaining games for only £3.50



Clive Gifford and Scott Vincent

MORE GAMES FOR YOUR ZX81

By Scott Vincent and Clive Gifford

Virgin Books

First published in Great Britain in 1983 by Virgin Books Ltd, 61-63 Portobello Road, London W11 3DD.

Copyright © 1983 Interface/Virgin Books

ISBN 0 907080 98 7

All rights reserved. No part of this book may be reproduced in any form or by any means without prior permission from the publisher.

Printed and bound in Great Britain by Richard Clay (The Chaucer Press) Ltd, Suffolk.

Production services by Book Production Consultants, Cambridge.

Designed by Ray Hyden.

Illustrated by Sue Walliker.

Typeset by QV Typesetting.

Distributed by Arrow Books.

TIM HARTNELL — SERIES EDITOR

Tim Hartnell is a leading computer expert and journalist, who has contributed extensively to the Technical Consumer Press. He is also the author of several books including *Getting Acquainted With Your ZX81*, *Let Your BBC Micro Teach You to Program* and *Programming Your ZX Spectrum*.

SCOTT VINCENT — AUTHOR

Scott Vincent is a 17-year-old student studying Advanced Maths and Physics at a sixth form college. He has written games for several major software houses and has a thorough knowledge of Z80 machine code. His main pastime, other than computing, is sailing.

CLIVE GIFFORD — AUTHOR

Clive Gifford has previously written *Games for Your Dragon* (also in this series) and *Making the Most of Your Dragon*, and has been involved with computers for nearly three years. He writes for several magazines including *Dragon User*, and formerly wrote for *Interface* magazine. He made several contributions to the first *Games for Your ZX81* book and enjoys playing golf and listening to music.

SUE WALLIKER — THE ILLUSTRATOR

Sue Walliker is a freelance illustrator.

ACKNOWLEDGEMENTS

The authors wish to thank Simon Gould and David Turner for their contributions, Tim and Liz for their support, and our families for their help and advice at all times.

TO TERRY AND CHRISSY, PETER AND SALLY

CONTENTS

Editor's Introduction

Your computer is waiting to challenge you. Moving graphics games, brain stretchers, word games and puzzles are all here and ready to entertain you.

A wide variety of games are included in this book. The programs have been written by some of the most talented young programmers working in this country at the moment, and represent a variety of approaches to solving programming problems.

An examination of the listings should teach you many tricks and techniques to apply to your own programming. And once you have mastered the programs in their present form, you might want to try your hand at improving them. There is no such thing as a 'perfect program', so these games are sure to benefit from your programming skill.

All that now remains is for you to turn the page and enter the programs. I can only hope that you enjoy playing the games as much as we did when preparing this volume.

Tim Hartnell, series editor
London
March 1983

Authors' Introduction

The ZX81 is a remarkable computer still topping the charts for sales over two years after it was orginally launched. This machine, coupling advanced features with low cost, has played a vital part in the computer boom that has recently started and will continue for many years.

We have tried to cover as many aspects as possible of this multi-faceted machine, and games in this book include gambling games, arcade games, strategy games and adventure games.

The programs were not rushed, much time has been taken to check the programs carefully for any errors and to maintain a balance between the computer's and the player's chances of winning. Many programs feature useful routines which you can incorporate into your own programs.

Here, then, is a collection of games to suit people of all ages. We have enjoyed ourselves immensely in the creation and the testing of these games and we hope that you enjoy them as much as we do.

Scott Vincent and Clive Gifford
Ashford, Middlesex
July 1983

Program Notes

There are some programs that use machine code for greater speed or flexibility. The machine code in these programs is all self-contained, that is, it is POKEd into the memory by the main program and no other program is needed to perform this function. The machine code is stored in a REM statement which is always the first line. Care must be taken when typing in the correct number of zeros after the REM.

The machine code takes some time to be POKEd into the memory but after the initial RUN, the program jumps over the machine code part, thus saving a lot of time. The program should be saved when the machine code has been entered into the memory.

DEFENCE COMMAND

You have been appointed Commander in charge of the Earth's defences. Your predecessor has failed and only two cities remain. A fleet of 20 enemy ships are rapidly approaching: man your laser gun and prepare for the war of the planets.

The ships attack individually and move from left to right and vice versa. When a ship reaches a certain height, it will bomb one of your cities. If you still have two cities left when a ship reaches this height, it will bomb the first city and fly over the second. This is because (luckily for you) each ship can only carry one bomb.

Fire your laser either left or right and, if a ship is in your line of fire, it will be destroyed and a new ship will attack.

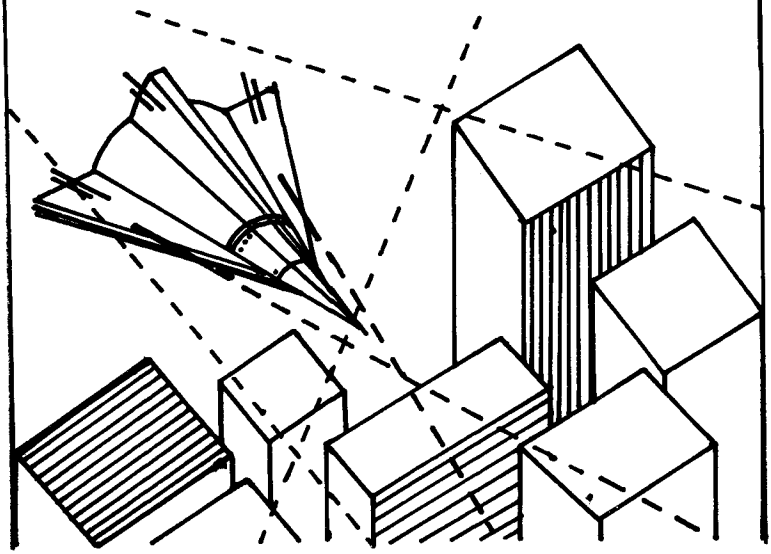
Keys 1 to 5 fire left and keys 6 to 0 fire right.

```
   1 REM 000000000000000000000000
000000000000000000000000000000000
000000000000000000000000000000000
   5 IF PEEK 16597=201 THEN GOTO
 45
  10 LET A$="000 058 130 064 254
 000 040 005 061 050 130 064 201
 205 187 002 125 254 247 032 012
 062 000 050 187 064 062 034 050
 190 064 024 "
  15 LET A$=A$+"013 254 239 192
062 035 050 187 064 062 032 050
190 064 062 003 050 130 064 042
012 064 017 255 001 000 025 017
034 000 006 016 "
  20 LET A$=A$+"126 254 128 040
007 254 155 040 003 054 128 201
054 155 167 237 082 016 237 201
"
  25 FOR A=16514 TO 16597
  30 POKE A,VAL A$( TO 3)
  35 LET A$=A$(5 TO )
```

```
  40 NEXT A
  45 RAND
  48 GOSUB 1500
  50 LET X=0
  55 LET Y=0
  60 LET NO=0
  65 LET SC=0
  67 LET V=1
  69 LET W=1
  70 CLS
  75 FOR A=1 TO 18
  80 PRINT "
                                    "
  85 NEXT A
  90 PRINT AT 16,15;"■ ■";AT 17,1
4;"■   ■"
 100 LET A$="▜ ▛"
 110 PRINT AT 18,4;A$;AT 18,24;A
$
 120 PRINT AT 17,5;"▜▛";AT 17,25
;"▜▛"
 130 PRINT AT 18,0;"▜▛";AT 18,2
 140 LET DF=PEEK 16396+256*PEEK
16397
 150 IF Y<>0 OR X<>0 THEN GOTO 1
90
 164 IF RND<.5 THEN GOTO 172
 166 LET X=DF+67+INT (RND*3)*66
 168 POKE X,146
 170 GOTO 180
 172 LET Y=DF+65+INT (RND*3)*66
 174 POKE Y,147
 180 LET NO=NO+1
 185 LET HT=0
 190 IF X=0 THEN GOTO 310
 200 IF PEEK X<>128 THEN GOTO 24
0
 210 LET X=0
 220 LET SC=SC+13
 230 GOTO 310
 240 POKE X,128
 250 LET X=X+1+34*(PEEK (X+1)=11
8)
 260 POKE X,146
 270 IF X=DF+336 OR X=DF+357 THE
N GOSUB 1000
 280 IF X<>DF+362 THEN GOTO 310
 290 POKE X,128
 300 LET X=0
 310 IF Y=0 THEN GOTO 430
 320 IF PEEK Y<>128 THEN GOTO 36
0
```

```
 330 LET Y=0
 340 LET SC=SC+13
 350 GOTO 430
 360 POKE Y,128
 370 LET Y=Y-1+96*(PEEK (Y-1)=11
8)
 380 POKE Y,147
 390 IF Y=DF+303 OR Y=DF+324 THE
N GOSUB 1000
 400 IF Y<>DF+298 THEN GOTO 430
 410 POKE Y,128
 420 LET Y=0
 430 IF V=0 AND W=0 THEN GOTO 20
00
 440 POKE 16591,155
 450 LET Z=USR 16515
 460 POKE 16591,128
 470 IF PEEK 16514=3 THEN LET Z=
USR 16565
 480 IF NO=20 AND X=0 AND Y=0 TH
EN GOTO 2030
 490 GOTO 160
1000 IF HT=1 THEN RETURN
1005 LET Z=0
1010 IF V=1 AND (PEEK (DF+336)=1
46 OR PEEK (DF+303)=147) THEN LE
T Z=DF+369
1020 IF W=1 AND (PEEK (DF+357)=1
46 OR PEEK (DF+324)=147) THEN LE
T Z=DF+390
1025 IF Z=0 THEN RETURN
1030 IF Z=DF+369 THEN LET V=0
1040 IF Z=DF+490 THEN LET W=0
1050 POKE Z,132
1060 FOR A=0 TO 198 STEP 33
1070 POKE Z+A,128
1080 POKE Z+Z+33,132
1090 NEXT A
1095 LET Z=Z+7*33-(Z=DF+390)
1100 POKE Z-33,128
1110 POKE Z-32,128
1120 FOR A=1 TO 20
1130 POKE Z,0
1140 POKE Z+1,0
1150 POKE Z,128
1160 POKE Z+1,128
1170 NEXT A
1175 LET HT=1
1180 RETURN
1500 CLS
1510 LET A$=" ■EFENCE COMMAN■"
1520 FOR N=0 TO 15
1530 PRINT AT 21,0;A$(16-N TO )
1540 NEXT N
1550 FOR N=1 TO 7
1560 PRINT AT 21,N;A$
```

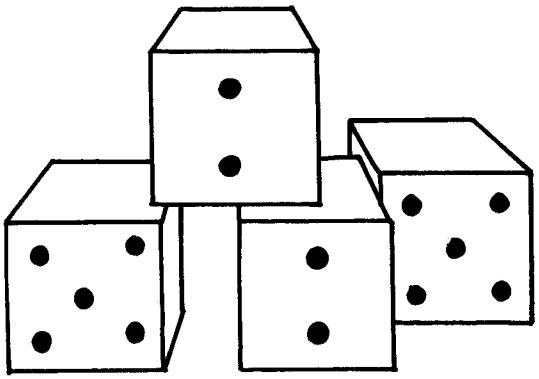
```
1570 NEXT N
1580 FOR N=20 TO 10 STEP -1
1590 PRINT AT N+1,8;"
     ";AT N,7;A$
1600 NEXT N
1610 FOR N=7 TO 0 STEP -1
1620 PRINT AT 10,N;A$( TO 9);AT
10,23-N;A$(9 TO )
1630 NEXT N
1640 PRINT AT 8,13;"
                    "
1650 PRINT AT 10,0;A$( TO 9);AT
10,23;A$(9 TO )
1660 PRINT AT 3,10;"PRESS ANY KE
Y";AT 17,13;" TO PLAY."
1670 IF INKEY$="" THEN GOTO 1670
1680 RETURN
2000 FOR A=1 TO 30
2010 PRINT AT INT (RND*4)+18,RND
*31;"■"
2020 NEXT A
2030 PRINT AT 8,9;"YOU SCORED ";
2040 PRINT AT 10,9;"PRESS ANY KE
Y"
2050 IF INKEY$="" THEN GOTO 2050
2060 RUN
2100 SAVE "DEFENCE COMMAN■"
2110 RUN
```



007 DICEMASTER

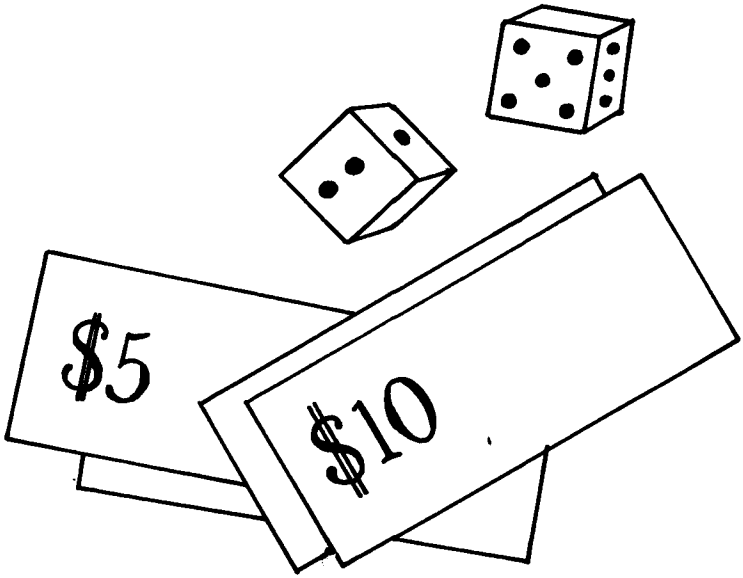
In this multi-player game you must try to throw a seven with your two dice. You win double your bet if you throw a seven and win your money back if you throw an odd number but, if you throw an even number, you lose your money. You have a choice of six game lengths and as many people as you like can play. The computer updates the situation every round and skilfully calculates its bet according to its financial situation and how recent games have gone.

The game stops after one go, but if you require a repeat option, add the following line: 490 GOTO 5.



```
  5 GOSUB 500
  9 LET Z=0
 10 LET V=0
 15 CLS
 16 PRINT '''''"LENGTH OF GAME?
(1-6)  (6=LONG)"
 17 INPUT L
 18 IF L<1 OR L>6 THEN GOTO 15
 19 LET L=L*5
 20 PRINT '''''"HOW MANY PLAYERS
?"
```

```
  30 INPUT P
  33 DIM A$(P+1,12)
  34 DIM M(P+1)
  35 FOR K=1 TO P
  37 PRINT
  40 PRINT "NAME?"
  45 INPUT A$(K)
  47 LET M(K)=40
  50 NEXT K
  55 LET A$(P+1)="COMPUTER"
  57 LET M(P+1)=40
  60 PRINT
  65 FOR K=1 TO P+1
  66 IF M(K)<=0 THEN NEXT K
  67 CLS
  68 PRINT "ENTER YOUR BET, ";A$
(K)
  69 IF K=P+1 THEN LET C=INT (.2
5*M(K))
  70 IF K=P+1 AND Z>2 THEN LET C
=INT (.10*M(K)*(2*Z/3))
  71 IF K=P+1 AND RND>.8 THEN LE
T C=C/INT (M(K)/6)
  72 IF K=P+1 AND RND<.2 THEN LE
T C=C-INT (M(K)/6)
  73 PRINT
  75 IF K=P+1 THEN PRINT "COMPUT
ER BETS ";C
  77 IF K=P+1 THEN GOTO 80
  78 INPUT C
  79 IF C>M(K) THEN GOTO 68
  80 PRINT "","",""
  82 PRINT "HIT N/L TO TO THROW
DICE"
  85 INPUT N$
  90 LET DD=INT (RND*6)+1
 100 LET D=INT (RND*6)+1
 105 PRINT "","",""
 110 PRINT D,DD
 115 LET T=D+DD
 117 PRINT
 120 PRINT "TOTAL OF ";T
 135 IF T/2=INT (T/2) THEN GOSUB
 420
 140 IF T=7 THEN GOSUB 300
 142 PRINT
 145 PRINT "PRESS NEWLINE TO CON
TINUE"
 147 IF K=P+1 AND T<>7 THEN LET
Z=Z+1
 150 INPUT N$
 155 CLS
 160 NEXT K
 162 LET U=U+1
 165 PRINT "END OF ROUND ";U
 167 FOR K=1 TO P+1
```




```
 168 IF M(K)<=0 THEN NEXT K
 170 PRINT ,,,,A$(K);" HAS ";M(K
);
 172 PRINT
 174 NEXT K
 175 IF V=L THEN GOTO 460
 176 PRINT "PRESS NEWLINE"
 178 INPUT N$
 180 GOTO 65
 300 PRINT ,,,,
 310 IF K=P+1 THEN PRINT "((((((
(((I WIN)))))))))"
 320 IF K<>P+1 THEN PRINT "(((((
((YOU WIN))))))"
 330 LET M(K)=M(K)+2*C
 400 RETURN
 420 PRINT ,,,,
 430 IF K<>P+1 THEN PRINT "YOU L
OSE BUDDY"
 432 IF K=P+1 THEN PRINT "I LOST
"
 435 LET M(K)=M(K)-C
 440 RETURN
 460 PAUSE 300
 462 CLS
 464 FOR T=1 TO 20
```

```
 466 PRINT "
"
 468 NEXT T
 470 PRINT AT 8,8;" END OF GAME
"
 480 PAUSE 1000
 500 CLS
 505 PRINT "          007 - DICEMA
STER"
 510 PRINT ,,,,"  YOU HAVE 40 PO
UNDS TO BET ON"
 520 PRINT "THE 2 DICE. YOU WIN
IF THE DICE"
 530 PRINT "COME UP SEVEN. YOU L
OSE IF THE"
 540 PRINT "DICE TOTAL AN EVEN N
UMBER. ANY"
 550 PRINT "OTHER SCORE DOES NOT
 COUNT."
 560 PRINT ,,,,"PRESS A KEY TO S
TART"
 570 IF INKEY$="" THEN GOTO 570
 580 PRINT ,,,,"
"
 590 PAUSE 200
 600 RETURN



END OF ROUND 1

CAROLINE      HAS 40

LAWRENCE      HAS 28

JEREMY        HAS 40

DEREK         HAS 74

COMPUTER      HAS 31
PRESS NEWLINE
```

SPACE CAVERN

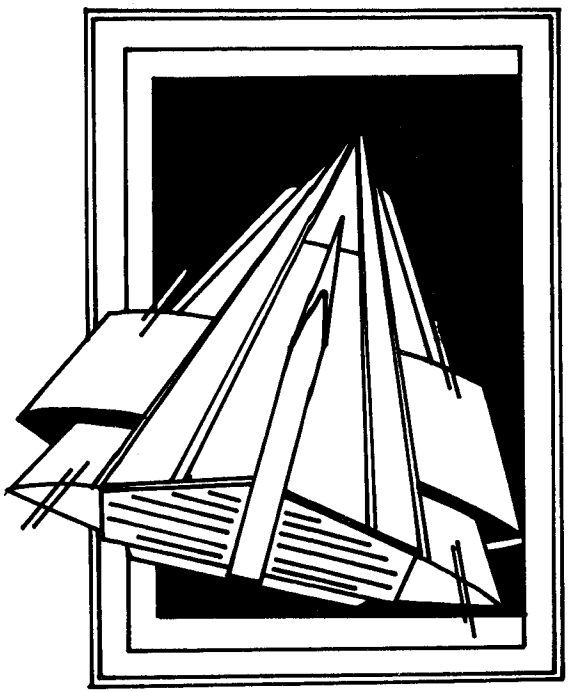
Space Cavern will keep all the arcade games' fans happy for a long, long time. It is a very fast, challenging game written partly in machine code for the extra speed necessary for a good action game on the ZX81.

You control a space cruiser and must try to travel through the dangerous Space Cavern. The path is twisting and turning, and the tunnel into the cavern is getting narrower and narrower. You control your craft with the up and down cursor keys. A game of Space Cavern is a real challenge to your reflexes.

```
   1 REM 000000000000000000000000
000000000000000000000000000000000
000000000000000000000000000000000
0000000000000000
   5 IF PEEK 16615=201 THEN GOTO
 50
  10 LET A$="042 130 064 054 000
 042 012 064 005 024 035 035 126
 254 118 040 004 043 119 024 245
 016 243 205 187 002 017 033 000
 124 254 239 "
  15 LET A$=A$+"032 008 042 130
064 167 237 082 024 008 254 223
042 130 064 032 001 025 126 254
128 032 005 062 001 050 132 064
054 018 034 130 "
  20 LET A$=A$+"064 201 042 012
064 017 033 000 025 043 014 024
006 001 054 128 025 013 016 250
006 001 054 000 025 013 016 250
065 054 128 025 "
  25 LET A$=A$+"016 251 201 "
  30 FOR A=16517 TO 16615
  35 POKE A,VAL A$( TO 3)
  40 LET A$=A$(5 TO )
  45 NEXT A
  50 LET HI=0
```

```
  60 LET DF=PEEK 16396+256*PEEK
16397
  65 LET A=DF+373
  70 LET B=INT (A/256)
  80 POKE 16514,A-B*256
  90 POKE 16515,B
 100 POKE 16516,0
 110 RAND
 115 LET A$="                SPACE CAV
ERN                "
 118 CLS
 119 POKE 16416,0
 120 PRINT A$;AT 23,0;A$
 121 POKE 16418,2
 122 LET SC=0
 125 LET G=6
 128 FOR A=1 TO 20
 130 LET Z=USR 16517
 140 IF PEEK 16516=1 THEN GOTO 1
000
 150 POKE 16594,A
 155 IF A>9 THEN POKE 16594,9
 160 POKE 16602,24-A*2
 165 IF A>9 THEN POKE 16602,6
 170 LET Z=USR 16583
 180 NEXT A
 185 LET A=9
 190 LET B=A-INT (RND*10)
 200 IF B<1 THEN LET B=1
 210 FOR N=A TO B STEP -1
 220 LET Z=USR 16517
 230 IF PEEK 16516=1 THEN GOTO 1
000
 240 POKE 16594,N
 250 POKE 16602,G
 260 LET Z=USR 16583
 270 NEXT N
 280 LET A=B+INT (RND*10)
 290 IF A>23-G THEN LET A=23-G
 300 FOR N=B TO A
 310 LET Z=USR 16517
 320 IF PEEK 16516=1 THEN GOTO 1
000
 330 POKE 16594,N
 340 POKE 16602,G
 350 LET Z=USR 16583
 360 NEXT N
 370 LET SC=SC+13
 380 IF SC/78=INT (SC/78) THEN L
ET G=G-1
 390 IF G<2 THEN LET G=2
 400 GOTO 190
1000 LET A=PEEK 16514+256*PEEK 1
6515
1010 FOR N=1 TO 25
```

```
1030 POKE A,23
1040 POKE A,128
1050 NEXT N
1060 PRINT AT 9,10;"SCORE = ";SC
1070 IF SC>HI THEN LET HI=SC
1080 PRINT AT 13,8;"HIGH SCORE =
 ";HI
1090 FOR A=1 TO 80
1100 NEXT A
1110 GOTO 60
5000 SAVE "SPACE CAVERN"
5010 RUN
```



PONTOON

This is a ZX81 version of the classic card game played all over the world. You and the computer take turns in trying to get as close to 21 as possible without exceeding that figure. You can also achieve five card tricks with this version. The computer is a difficult but not unbeatable opponent who weighs up the situation carefully before making a decision. This game is extremely friendly to play, with all the instructions contained in the program.

```
   5 CLS
  10 LET P$="**********PONTOON**
**********"
  20 PRINT P$
  25 RAND
  26 LET B=0
  28 LET C=0
  30 PRINT ,,,,"  PLEASE ENTER Y
OUR NAME"
  32 INPUT N$
  33 CLS
  34 PRINT ,,N$
  35 PRINT ,,"  I WILL TOSS A CO
IN TO SEE WHO GOES FIRST.
                DO YOU WANT HEAD
S OR TAILS ?          (PRESS H OR
 T)"
  40 LET A$=INKEY$
  50 IF A$<>"H" AND A$<>"T" THEN
 GOTO 40
  60 PRINT ,,A$
  70 LET X=INT (RND*2)
  80 IF X=0 THEN LET B$="HEADS"
  90 IF X=1 THEN LET B$="TAILS"
 100 PRINT ,,B$;", ";
 110 LET A=1
 120 IF X=1 AND A$="H" OR X=0 AN
D A$="T" THEN LET A=0
 130 IF A=0 THEN PRINT "I WIN"
 140 IF A=1 THEN PRINT "YOU WIN
";N$;"."
 142 LET X=0
 144 LET Y=0
 145 FOR F=1 TO 10
 146 PRINT ,,"PRESS ANY KEY FOR
GAME ";F
```

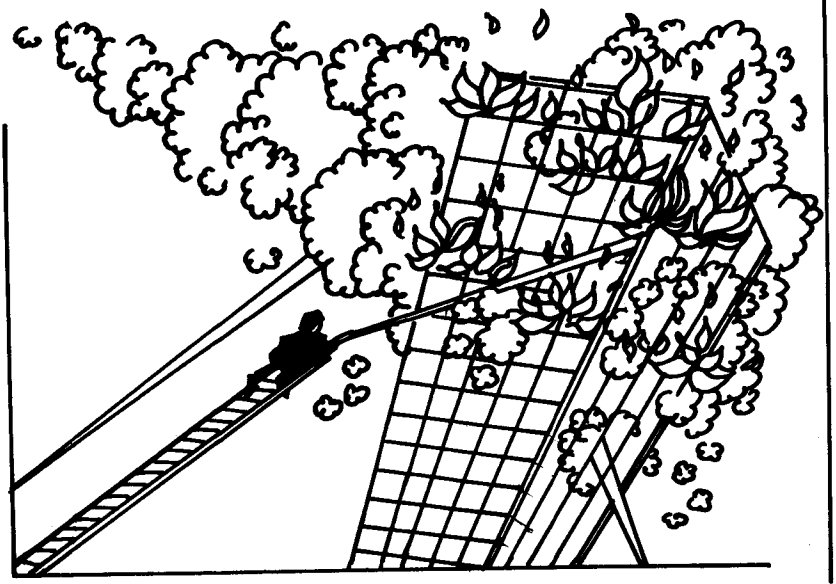
```
 148 IF INKEY$="" THEN GOTO 148
 150 IF A=0 THEN GOSUB 2000
 160 IF A=1 OR A=0 AND X>0 THEN
GOSUB 1000
 170 IF A=1 AND Y<>0 THEN GOSUB
2000
 175 IF Y=0 THEN LET X=1
 177 IF X=0 THEN LET Y=1
 180 IF X>Y THEN LET B=B+1
 190 IF X<Y THEN LET C=C+1
 200 PRINT ,," COMPUTER : ";N$
 210 PRINT ,,"          ";B;" : "
;C
 215 LET A=A+(A=0)-(A=1)
 220 NEXT F
 225 PAUSE 100
 230 PRINT ,,"I ENJOYED THAT"
 240 IF B>C THEN PRINT "MAYBE YO
U WILL BEAT ME NEXT TIME"
 250 IF B<=C THEN PRINT ,,"I WIL
L BEAT YOU NEXT TIME"
 260 PRINT ,," ANOTHER GAME Y/N
 ?"
 270 INPUT A$
 280 IF A$="Y" OR A$="YES" THEN
RUN
 290 PRINT ,,"BYE THEN"
 300 STOP
1000 PRINT ,," YOUR CARDS HAVE
BEEN DEALT.  PRESS ANY KEY TO T
URN THEM OVER."
1010 IF INKEY$="" THEN GOTO 1010
1020 GOSUB 3000
1025 CLS
1026 IF P=1 THEN GOSUB 3500
1030 PRINT ,,"      ";P,,
1040 LET Y=P
1045 LET L=1
1050 GOSUB 3000
1051 IF P=1 THEN GOSUB 3500
1055 IF L=3 THEN CLS
1060 PRINT ,,"      ";P
1070 LET Y=Y+P
1075 LET L=L+1
1080 PRINT ,,L;" CARDS. TOTAL=";
Y
1105 IF L=5 AND Y<=21 THEN GOTO
1200
1106 IF Y>21 THEN GOTO 1400
1108 IF Y=21 THEN GOTO 1600
1110 PRINT ,," TWIST OR STICK ?
(PRESS T OR S)"
1120 LET A$=INKEY$
1125 IF A$<>"T" AND A$<>"S" THEN
 GOTO 1120
1130 PRINT ,,"  ";A$
```

```
1140 IF A$="T" THEN GOTO 1050
1150 CLS
1160 RETURN
1200 CLS
1205 PRINT " YOU HAVE A FIVE CAR
D TRICK.     THIS CAN ONLY BE BEA
TEN BY       PONTOON."
1208 LET Y=22
1210 RETURN
1400 CLS
1405 PRINT " SORRY YOU HAVE BUST
.  I WIN"
1410 LET Y=0
1420 RETURN
1600 CLS
1605 PRINT " WELL DONE YOU HAVE
SCORED 21"
1608 LET Y=21
1610 IF L=2 THEN PRINT ,,P$
1615 IF L=2 THEN LET Y=23
1620 RETURN
2000 PRINT ,," PRESS ANY KEY TO
DEAL ME TWO   CARDS"
2010 IF INKEY$="" THEN GOTO 2010
2020 GOSUB 3000
2030 CLS
2040 IF P=1 THEN GOSUB 4000
2050 PRINT ,,"       ";P
2060 LET X=P
2070 LET M=1
2080 GOSUB 3000
2090 IF P=1 THEN GOSUB 4000
2100 IF M=3 THEN CLS
2110 PRINT ,,"       ";P
2120 LET X=X+P
2130 LET M=M+1
2140 PRINT ,,M;" CARDS. TOTAL=";
X
2145 PRINT ,,"PRESS ANY KEY"
2147 IF INKEY$="" THEN GOTO 2147
2150 IF M=5 AND X<=21 THEN GOTO
2400
2160 IF X>21 THEN GOTO 2500
2170 IF X=21 THEN GOTO 2600
2180 IF X>14+INT (RND*3) AND A=0
 THEN GOTO 2700
2190 IF X>14+INT (RND*3) AND RND
>.3 THEN GOTO 2700
2200 PRINT ,,"I THINK I WILL TWI
ST"
2210 GOTO 2080
2400 CLS
2410 PRINT "I HAVE A FIVE CARD T
RICK"
2420 IF A=0 THEN PRINT ,,"YOU NE
ED TO GET PONTOON TO BEAT ME"
```

```
2430 IF A=1 AND Y=22 THEN PRINT
"YOU STILL BEAT ME THOUGH AS YOU
 GOT PONTOON"
2440 IF A=1 AND Y<22 THEN PRINT
"THAT BEATS YOU. I WIN"
2450 LET X=22
2460 RETURN
2500 CLS
2510 PRINT " OH DEAR I HAVE BUST
. YOU WIN"
2520 LET X=0
2530 RETURN
2600 CLS
2610 PRINT " I HAVE SCORED 21"
2620 IF M=2 THEN PRINT ,,P$
2630 LET X=21
2640 IF M=2 THEN LET X=23
2650 RETURN
2700 CLS
2710 PRINT " I THINK I WILL STIC
K"
2720 RETURN
3000 LET P=INT (RND*13)+1
3010 IF P>10 THEN LET P=10
3020 RETURN
3500 IF Y>10 THEN GOTO 4100
3505 PRINT ,," AN ACE. DO YOU WA
NT IT TO BE  HIGH OR LOW ? (PRE
SS H OR L)"
3510 INPUT A$
3525 PRINT ,,A$
3530 IF A$="H" THEN LET P=11
3540 RETURN
4000 IF X>=7 OR X<=10 THEN LET P
=11
4010 IF X<7 OR X>10 THEN LET P=1
4020 PRINT ,,"I HAVE AN ACE."
4030 IF P=1 THEN PRINT " I THINK
 I WILL MAKE IT LOW"
4040 IF P=11 THEN PRINT "I THINK
 I WILL MAKE IT HIGH"
4050 RETURN
4100 PRINT ,,"AN ACE. IT HAD BET
TER BE LOW OR YOU WILL BUST"
4110 LET P=1
4120 RETURN
4500 SAVE "PONTOO"
4510 RUN
```

TOWERING INFERNO

A 30-storey skyscraper is on fire, you as fire chief must save the ten people still trapped in the burning debris. As they jump from the top of the building, you must catch them in your blanket. Nerves of steel are required. To move the blanket, press 2 for the left and 5 for the right.



```
    2 GOSUB 900
    3 LET P=0
    4 LET S=P
    6 LET F=1
    8 LET G=9
    9 CLS
   10 LET E=10
   12 FOR X=2 TO 20
   15 PRINT AT X,0;"[redacted]"
   18 NEXT X
   19 PRINT AT 21,0;"[redacted]"
   23 FOR X=3 TO 16 STEP 2
   24 PRINT AT X,1;" ";AT X,3;" "
;AT X,5;" ";AT X,7;" "
   26 PRINT AT X,1;" ";AT X,3;" "
;AT X,5;" ";AT X,7;" "
   28 NEXT X
   50 PRINT AT 20,E-1;"[redacted]"
   51 FOR T=1 TO 2
   52 NEXT T
   53 PRINT AT F,G;"X"
   58 IF F<INT (RND*5)+2 THEN GOT
O 80
   59 PRINT AT 20,E-1;"   "
   60 IF INKEY$="5" THEN LET E=E+
1
   62 IF E>31 THEN LET E=31
   70 IF INKEY$="2" THEN LET E=E-
1
   72 IF E<10 THEN LET E=10
   85 PRINT AT F,G;" "
   90 LET F=F+1
   94 IF F>18 THEN GOTO 105
  100 LET G=G+INT (RND*4)-1
  102 IF G<10 THEN LET G=10
  105 IF F=21 THEN GOTO 165
  110 GOTO CODE "C"
  165 LET P=P+1
  170 IF G=E OR G=E-1 THEN GOTO 3
00
  179 PRINT AT 21,G;"[redacted]"
  180 FOR R=1 TO 10
  181 PRINT AT 7,16;"SPLAT"
  182 PRINT AT 7,16;"[redacted]"
  183 NEXT R
  185 PRINT AT 9,12;"SCORE= ";S;"
OUT OF ";P
  186 IF P=10 THEN GOTO 360
  190 CLS
  200 GOTO 6
  300 LET S=S+1
  305 FOR Q=1 TO 10
  310 PRINT AT 7,14;"[redacted]"
  315 PRINT AT 7,14;"WELL CAUGHT"
```

```
 320 NEXT Q
 340 GOTO 185
 360 CLS
 370 FOR X=1 TO 21
 380 PRINT "**********************
**********"
 390 NEXT X
 395 FOR K=7 TO 13
 400 PRINT AT K,4;"
                  "
 410 NEXT K
 460 PRINT AT 8,5;"YOUR FINAL SC
ORE WAS ";S
 465 IF S=10 THEN GOTO 495
 470 PRINT AT 10,5;"YOU ARE DEMO
TED UNLESS"
 480 PRINT AT 11,5;"YOU DO BETTE
R THIS TIME"
 490 GOTO 500
 491 FOR K=1 TO 15
 493 PRINT AT 10,5;"▚▚▚▚▚▚WELL D
ONE▚▚▚▚▚▚"
 495 PRINT AT 10,5;"▚▚▚▚▚▚WELL D
ONE▚▚▚▚▚▚"
 496 NEXT K
 500 PAUSE 250
 510 RUN
 900 CLS
 910 PRINT AT 6,6;"THE TOWERING
INFERNO"
 920 PRINT AT 8,9;"2=LEFT, 5=RIG
HT"
 930 PRINT AT 14,5;"PRESS ANY KE
Y TO START"
 940 IF INKEY$="" THEN GOTO 940
 950 RETURN
1000 SAVE "THE TOWERING INFERN▒"
1010 RUN
```

REPEAT

Repeat is a simple 1K game where you must repeat the selection of characters that appear on the screen. Concentrate now, they only appear for a very short time. Every time you guess the correct combination of symbols, an extra character is added on. When you reach the full amount of characters and you guess them all correctly, you win the game.

```
 100 LET X=6+INT (RND*10)
 105 DIM F$(X)
 110 FOR N=1 TO X
 120 LET F$(N)=CHR$ (28+INT (RND
*10))
 130 NEXT N
 140 PRINT "THIS NUMBER HAS ";LE
N (F$);" FIGURES"
 150 FOR T=1 TO LEN (F$)
 160 PRINT AT 10,10;F$(1 TO T)
 170 FOR Y=1 TO 20
 180 NEXT Y
 190 CLS
 200 PRINT "ENTER GUESS"
 210 INPUT A$
 220 IF A$<>F$(1 TO T) THEN GOTO
 300
 230 PRINT AT 14,7;"▚▚CORRECT▚▚
▚"
 235 PAUSE 150
 240 CLS
 250 NEXT T
 260 PRINT AT 10,10;"YOU WIN"
 270 STOP
 300 PRINT ,,,,,"▚▚WRONG▚▚"
 310 PRINT "END OF GAME"
 320 PRINT ,,,,F$
```

1234567891011 REPEATREPEAT

REPEATREPEAT 1234567891011

LAST HOPE

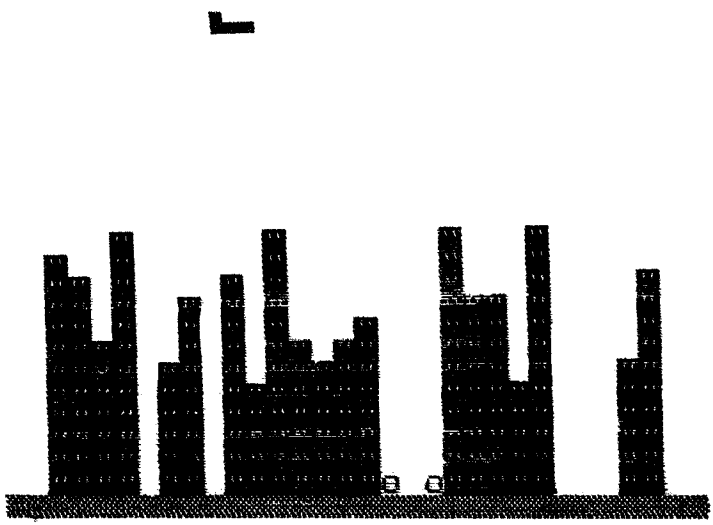
You are the last member of your squadron of B17 bombers. All the others have failed in their task of destroying the enemy's prime city and the success of the whole mission lies in your hands. You fly over the partially destroyed city, gradually getting lower and lower. By pressing the 'P' key you can drop a bomb on the city. You must destroy all of the city before your battered aircraft can land safely. If you hit the fuel dumps (Os) then your height is increased by two lines.

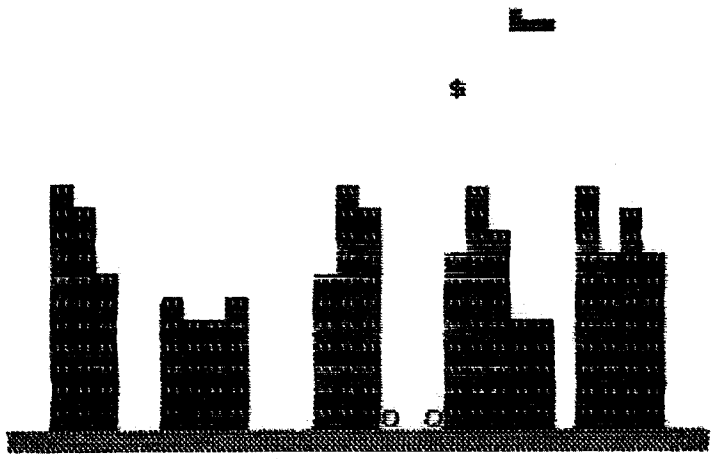
You must not let your squadron down, good luck.

```
   5 RAND
  10 LET X=0
  15 LET DF=PEEK 16396+256*PEEK
16397
  20 LET Y=0
  25 LET SC=0
  30 CLS
  40 FOR L=2 TO 30
  50 LET A=INT (RND*8)+10
  55 IF RND>.8 OR L>15 AND L<20
THEN GOTO 90
  60 FOR B=A TO 21
  70 PRINT AT B,L;"■"
  80 NEXT B
  90 NEXT L
  95 POKE 16418,0
 100 PRINT AT 22,0;"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";AT 21,17;"
0 0"
 210 PRINT AT X,Y;"▄▄";
 215 IF PEEK (PEEK 16398+256*PEE
```

```
K 16399)=139 THEN GOTO 800
  220 IF INKEY$="P" THEN GOTO 500
  225 LET SC=SC+17
  260 PRINT AT X,Y;"  "
  270 LET Y=Y+1
  280 IF Y=31 THEN LET X=X+1
  285 IF Y=31 THEN LET Y=0
  287 IF X=21 AND Y=13 THEN GOTO
900
  290 GOTO 210
  500 LET XX=X
  505 LET YY=Y
  510 LET BC=DF+33*(X+1)+Y+1
  512 LET CB=BC
  515 PRINT AT XX,YY;"  "
  518 IF PEEK BC=52 THEN LET X=X-
2+(X=1)+(X=0)
  519 PRINT AT X,Y;"▙▄";
  520 IF PEEK (PEEK 16398+256*PEE
K 16399)=139 THEN GOTO 800
  521 LET XX=X
  522 LET YY=Y
  524 POKE CB,0
  525 IF BC>DF+726 THEN GOTO 225
  526 POKE BC,13
  530 LET CB=BC
  540 LET Y=Y+1
  550 IF Y=31 THEN LET X=X+1
  560 IF Y=31 THEN LET Y=0
  565 LET BC=BC+33
  580 GOTO 515
  800 CLS
  810 PRINT AT 14,8;"YOU SCORED "
;SC
  815 PRINT AT 18,2;"PRESS ANY KE
Y TO PLAY AGAIN"
  820 PRINT AT 8,8;"CCCRRRAAASSSH
HH."
  830 PRINT AT 8,8;"▀▀▀▀▀▀▀▀▀▀▀▀▀
▀▀▀"
  840 IF INKEY$="" THEN GOTO 820
  850 RUN
  900 PRINT AT X,Y;"▙▄"
  905 PRINT AT 10,2;"PRESS ANY KE
Y TO PLAY AGAIN"
  910 PRINT AT 8,8;"▀▀▀▀ATULATIO
NS"
  915 LET N=9*9*9*9*9*9
  920 PRINT AT 8,8;"CONGR▀▀▀▀TIO
NS"
  925 LET N=9*9*9*9*9*9
  930 PRINT AT 8,8;"CONGRATULA▀▀▀
▀▀"
  935 LET N=9*9*9*9*9*9
  960 IF INKEY$="" THEN GOTO 910
  970 RUN
```

```
 999 STOP
1000 SAVE "LAST HOP"
1100 RUN
```

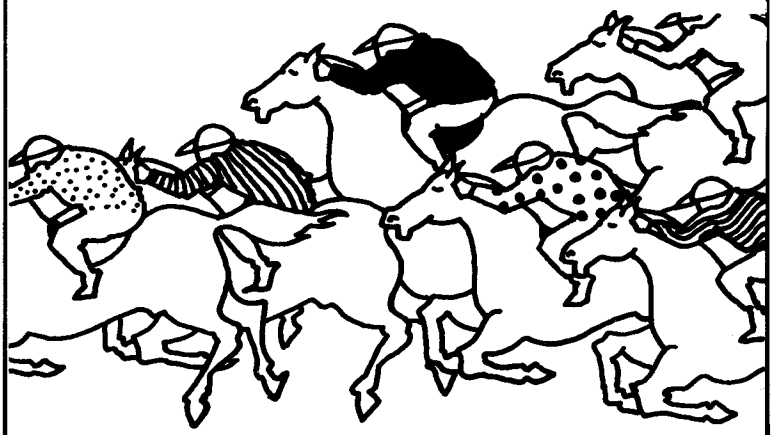




A DAY AT THE RACES

Why bother to travel to a racetrack and lose all your money when you can gamble to your heart's content in the comfort of your own home? This game puts you on a course of your choice, betting a portion of your £200 on one of eight horses. There are different odds, a payout for winning and a graphic display of the race in progress with the horses jumping the fences and galloping to the finishing post.

The game is simple to follow. To place a bet, just enter the number of the required horse and the amount you wish to bet and gamble away!



```
  20 GOSUB 1470
  30 DIM H(8)
  35 LET M=200
  36 DIM D(8)
  38 GOSUB 1000
  40 FOR P=1 TO 8
  41 LET H(P)=0
  42 LET D(P)=0
  43 NEXT P
  44 CLS
  45 FOR L=4 TO 20
  50 PRINT AT L,29;"▚";AT L,8;"/
";AT L,16;"/";AT L,24;"/"
  55 NEXT L
  60 PRINT AT 0,0;"
                                  "
  62 PRINT "          A DAY AT THE
RACES"
  63 PRINT "
                                  "
  64 PRINT "
                                  "
  80 FOR P=1 TO 8
  82 IF H(P)=1 THEN PRINT AT P*2
+3,D(P);"▟▙"
  84 IF H(P)=1 THEN GOTO 150
  85 LET R=RND
  87 PRINT AT P*2+3,D(P);" "
  90 IF R>.9 THEN LET D(P)=D(P)+
2
  95 IF R>.45 AND R<.9 THEN LET
D(P)=D(P)+1
  98 IF R<.35 THEN LET D(P)=D(P)
+3
  99 IF P<4 AND RND<.5 THEN LET
D(P)=D(P)+INT (RND*3)
 100 IF P<7 AND P>3 AND RND<.34
THEN LET D(P)=D(P)+1
 108 IF D(P)=8 OR D(P)=16 OR D(P
)=24 THEN GOSUB 350
 110 IF D(P)>=29 THEN GOTO 490
 120 PRINT AT P*2+3,D(P);P
 150 NEXT P
 152 LET N=0
 155 FOR P=1 TO 8
 156 IF H(P)=1 THEN LET N=N+1
 158 NEXT P
 170 IF N<>8 THEN GOTO 80
 180 PRINT AT 10,10;"
"
 190 PAUSE 150
 195 CLS
 200 GOTO 640
 350 LET J=RND
 360 IF J>A THEN PRINT AT P*2+3,
D(P);"▟▙"
```

```
 370 IF J>A THEN LET H(P)=1
 380 RETURN
 490 CLS
 500 FOR T=1 TO 10
 510 NEXT T
 520 FOR S=1 TO 21
 530 PRINT "
"
 540 NEXT S
 545 PRINT AT 7,6;"
"
 550 PRINT AT 8,6;"THE WINNER W
AS ";P;"."
 560 PRINT AT 9,6;"
"
 570 PAUSE 100
 580 CLS
 600 IF P=Z AND Z<4 THEN LET M=M
+(MO*2)
 610 IF P=Z AND Z>6 THEN LET M=M
+(MO*8)
 620 IF P=Z AND Z<7 AND Z>3 THEN
 LET M=M+(MO*4)
 630 IF P=Z THEN PRINT "WELL DON
E"
 640 IF Z<>P THEN PRINT "HARD LU
CK, TRY AGAIN."
 650 IF M<1 THEN GOTO 800
 700 PAUSE 100
 710 GOSUB 1000
 720 GOTO 40
 800 CLS
 810 PRINT "YOU HAVE RUN OUT OF
MONEY"
 820 PRINT "
"
1000 REM ********ALL SETTING INS
TRUCTIONS HERE********
1005 CLS
1010 PRINT "     A DAY AT THE
RACES    "
1015 PRINT "          **********
*****          "
1020 PRINT ,,,,"WELCOME PUNTER."
1025 PRINT ,,"CHOOSE YOUR HORSE;
"
1030 PRINT AT 8,4;"1    WAYWARD S
ON     2 TO 1"
1040 PRINT AT 9,4;"2    LITTLE GE
NIE    2 TO 1"
1050 PRINT AT 10,4;"3    BLUE MIS
T       2 TO 1"
1060 PRINT AT 11,4;"4    MILL ROU
GH      4 TO 1"
1070 PRINT AT 12,4;"5    CRYSTAL
CLEAR   4 TO 1"
1075 PRINT AT 13,4;"6    FRISKY F
```

```
ILLY    4 TO 1"
1080 PRINT AT 14,4;"7    BROWN BE
AUTY    8 TO 1"
1085 PRINT AT 15,4;"8    LUCKY LO
SER     8 TO 1"
1090 INPUT Z
1095 IF Z<1 OR Z>8 THEN GOTO 109
0
1120 CLS
1130 PRINT ,,,,"HOW MUCH DO YOU
WANT TO BET?"
1140 PRINT ,,"YOU HAVE £ ";M
1150 INPUT MO
1160 IF MO<1 OR MO>M THEN GOTO 1
150
1170 LET M=M-MO
1180 PRINT ,,"YOU NOW HAVE £ ";M
1200 PAUSE 100
1300 RETURN
1470 CLS
1480 PRINT "CHOOSE COURSE; AINTR
EE=1"
1484 PRINT "                LEOPA
RDSTOWN=2"
1486 PRINT "                ASCOT
=3"
1488 PRINT "                CHELT
ENHAM=4"
1490 PRINT "                NEWBU
RY=5"
1500 INPUT A
1510 LET A=((A/10)*2)-0.05
1790 CLS
1800 RETURN
2000 SAVE "A.DAY AT THE RACES"
2010 RUN
```

DOWNHILL SKIER

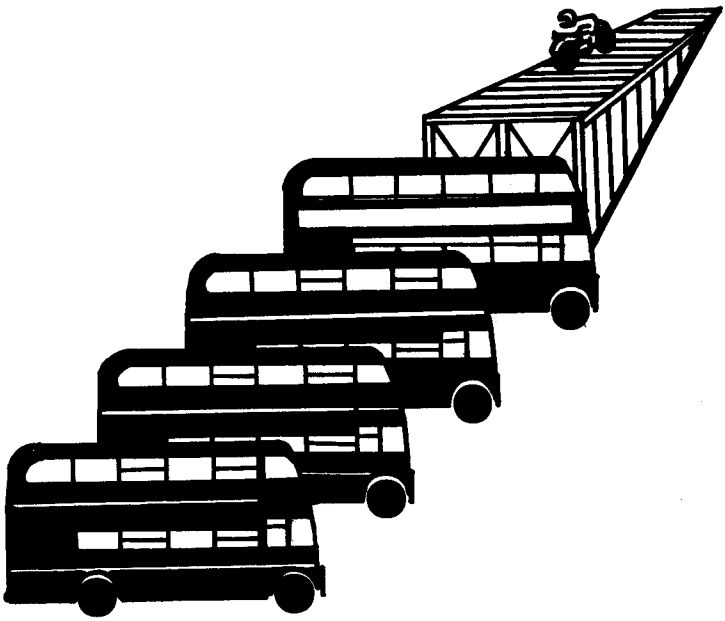
You are competing on the slopes of the famous Mount Zwegan. You must ski through the gates to score points while avoiding the rocks and ice patches. You move your skier with the left and right cursor keys in this action game. Full instructions are included in the program.

```
    1 REM 00000000000000000000000
0000
    2 GOSUB 700
    3 RAND
    4 CLS
    5 IF PEEK 16540=201 THEN GOTO
 60
   10 LET A$="042 012 064 017 033
 000 025 006 022 035 126 254 118
 040 009 079 167 237 082 121 119
 025 024 241 016 239 201"
   20 FOR A=16514 TO 16540
   30 POKE A,VAL A$( TO 3)
   40 LET A$=A$(5 TO )
   50 NEXT A
   60 LET DF=PEEK 16396+256*PEEK
16397
   65 LET SC=0
   70 LET X=DF+147
   75 LET CT=9
   80 LET Y=X
   85 LET GT=0
   90 LET A=DF+694
  100 REM ****MAIN ROUTINE****
  110 POKE Y,0
  120 LET Q=USR 16514
  130 IF PEEK X=27 THEN LET SC=SC
+17
  140 IF PEEK X=133 OR PEEK X=5 O
R PEEK X=8 THEN GOTO 500
  150 POKE X,59
  160 IF CT/3=INT (CT/3) THEN POK
E A+INT (RND*15)*2,8
  165 IF CT<>9 THEN GOTO 220
  166 IF GT=22 THEN GOTO 570
  168 LET CT=1
  169 LET B=INT (RND*27)
  170 POKE A+B,133
  180 POKE A+B+1,27
  190 POKE A+B+2,27
  200 POKE A+B+3,27
  210 POKE A+B+4,5
  215 LET GT=GT+1
  220 LET Y=X
  230 LET X=X+2*(INKEY$="8" AND X
<DF+163)-2*(INKEY$="5" AND X>DF+
134)
  235 LET CT=CT+1
  240 GOTO 110
  500 FOR A=1 TO 30
  510 POKE X,RND*27
  515 POKE X+33,RND*27
  520 NEXT A
  530 CLS
  540 PRINT AT 8,13;"CRUNCH"
  545 LET R=INT (RND*6)
```

```
548 IF R=0 THEN LET B$="NECK"
550 IF R=1 THEN LET B$="LEG"
552 IF R=2 THEN LET B$="ARM"
554 IF R=3 THEN LET B$="NOSE"
556 IF R=4 THEN LET B$="ANKLE"
558 IF R=5 THEN LET B$="SHOULDER"
560 PRINT AT 10,1;"YOU HAVE BROKEN YOUR ";B$
570 PRINT AT 12,9;"YOU SCORED ";SC
580 PRINT AT 14,2;"PRESS ANY KEY TO TRY AGAIN"
590 IF INKEY$="" THEN GOTO 590
600 RUN
700 CLS
710 PRINT "*********DOWNHILL SKIER*********"
720 PRINT ,,,," YOU ARE ON THE TRICKY SLOPES  OF MOUNT ZWEGAN, COMPETING WITH EUROPES FINEST SKIERS."
730 PRINT " YOU MUST SKI THROUGH THE GATES(▌..▐) TO SCORE. YOU CRASH IF YOU EITHER HIT THE POSTS OR THE ICE PATCHES (▒)."
740 PRINT " THERE ARE 20 GATES AND THE    MORE YOU SKI THROUGH THE HIGHER YOUR SCORE."
750 PRINT " YOUR SKIER (V) IS MOVED BY    THE S AND 8 KEYS."
760 PRINT ,,,,,,,,"▚▚▚▚▚▚▚▚▚▚▚HAPPY SKIING▚▚▚▚▚▚▚▚▚▚▚"
770 PRINT ,,,,"                 PRESS A KEY."
780 IF INKEY$="" THEN GOTO 780
790 RETURN
800 SAVE "DOWNHILL SKIER"
810 RUN
```

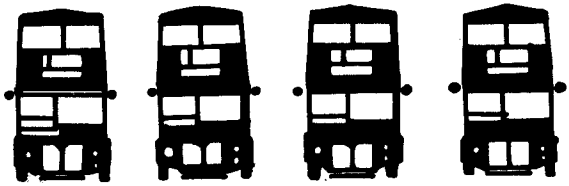
MOTORCYCLE STUNTMAN

In this game, it's on with your crash helmet and away you go. You are trying to jump over a line of buses, which change in number randomly every game. Hold down any key, watch your speed increase, let go of the brakes (take your finger off the key) and watch your motorcycle fly over the ramps. This great game uses accurate formulae for thrust and velocity and is excellent fun.



```
  10 RAND
  15 LET BUS=INT (RND*17)+2
  20 GOSUB 1000
  30 LET M=0
  32 PRINT AT 1,11;"0 M.P.H."
  35 IF INKEY$="" THEN GOTO 35
  40 LET M=M+5+INT (RND*6)
  45 IF M>170 THEN LET M=170
  50 PRINT AT 1,11;M;" M.P.H."
  60 IF INKEY$<>"" THEN GOTO 40
  65 LET MP=M
  70 LET Z=INT (RND*11)
  80 LET M=M+(Z-5)-40
  90 LET J=9+INT (M/7.5+.5)
 100 IF M<-5 THEN LET J=6
 110 FOR N=1 TO 4
 120 PRINT AT 10,N-1;" ";AT 10,N;"_"
 130 NEXT N
 140 PRINT AT 10,4;" ";AT 9,5;"/"
 150 PRINT AT 9,5;" ";AT 8,6;"/"
 160 IF J=6 THEN GOTO 200
 170 FOR N=7 TO J
 180 PRINT AT 8,N-1;" ";AT 8,N;"/"
 190 NEXT N
 200 PRINT AT 8,J;" ";AT 9,J+1;"/"
 210 PRINT AT 9,J+1;" ";AT 10,J+2;"/"
 220 IF J<>BUS+7 THEN GOTO 2000
 230 FOR N=J+3 TO 31
 240 PRINT AT 10,N-1;" ";AT 10,N;"_"
 250 NEXT N
 260 PRINT AT 10,31;" "
 270 CLS
 280 PRINT AT 4,11;"GREAT JUMP"
 290 PRINT AT 6,2;"YOU CLEARED ALL ";BUS;" BUSES AT"
 300 PRINT AT 8,6;"A SPEED OF ";MP;" M.P.H."
 310 FOR N=1 TO 150
 320 NEXT N
 330 RUN
1000 CLS
1010 PRINT AT 11,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1020 PRINT AT 10,5;"▟█";AT 9,6;"▖"
1030 FOR N=7 TO BUS+6
1040 PRINT AT 10,N;"A"
1050 NEXT N
1060 PRINT AT 10,BUS+7;"█▙";AT 9,BUS+7;"▗"
```

```
1070 PRINT AT 10,0;"  "
1080 RETURN
2000 FOR N=1 TO 20
2010 PRINT AT 10,J+2;"■"
2020 PRINT AT 10,J+2;" "
2030 NEXT N
2040 CLS
2045 IF J<BUS+7 THEN GOTO 2100
2050 PRINT AT 4,3;"YOU WENT TOO
FAST AND YOU"
2060 PRINT AT 6,4;"MISSED THE LA
NDING RAMP"
2070 PRINT AT 8,7;"TRY A SLOWER
SPEED"
2080 FOR N=1 TO 120
2085 NEXT N
2090 GOTO 20
2100 PRINT AT 4,1;"YOU WENT TOO
SLOW AND YOUR BIKE"
2110 PRINT AT 6,9;"IS A WRITE OF
F"
2120 PRINT AT 8,3;"TRY AGAIN ON
A NEW BIKE"
2130 FOR N=1 TO 120
2140 NEXT N
2150 GOTO 20
5000 SAVE "MOTORCYCLE STUNTMA■"
5010 RUN
```



MISSION APOLLO

You are the Commander of Apollo 24. You must try to land your craft on the lunar surface at a speed of less than five miles per second.

Skilful manoeuvering is required, although computer reports indicate that your landing site is good. To control the rate of descent use a key between 1 and 9. This indicates the amount of thrust you wish to use which slows down the rate of descent but uses fuel. Key 9 gives you the most thrust. Be careful, as your fuel line can block at any time, particularly when you are in the final stages of your descent. The middle mark on the speed graphic display is the balance between descent and ascent.

The whole game revolves around a graphic display of the craft's controls which shows the height, fine height, fuel, speed and time and is based on real formulae scaled down for this game. Depending on how well you do, the game rewards you with one of six cartoon displays at the end.

This excellent game was written by D.N. and M.D. Turner of Ashford, Middlesex.

```
    1 REM 0000000000000000000000
    2 REM 0000000000000000000000000000
    3 REM 0000000000000000000000000000
0000000000000000000000000000000000000
0
    4 REM 0000000
    6 GOSUB 6000
    8 CLS
    9 FOR F=1 TO 25
   10 PRINT AT 8,8;"MISSION APOLL
O"
   12 PRINT AT 8,8;"[illegible]
[illegible]"
```

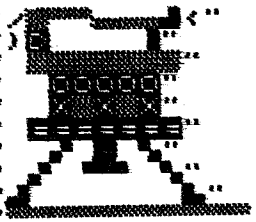
```
  14 NEXT F
  24 RAND
  25 LET E=0
  27 LET Q=0
  30 FOR I=1 TO 75
  40 NEXT I
  90 CLS
  95 LET R=INT (SQR (RND*5000))
 100 LET X=31
 110 LET S=0
 120 LET V=100
 130 LET U=500
 140 LET H=1000
 150 LET M=2.5
 160 LET A$="▌▀▀▀▀▌▀▀▀▀▌▀▀▀▀▌▀▀▀▀▌▀▀▀▀▌"
 164 PRINT AT 0,1;"TIME=0000";AT
 3,5;"O";TAB 16;"FUEL";TAB 29;"5
00";TAB 5;A$;AT 8,5;"O";TAB 15;"
HEIGHT";TAB 28;"1000";TAB 5;A$;A
T 13,5;"O";TAB 13;"FINE HEIGHT";
TAB 29;"100";AT 14,5;A$;AT 18,16
;"SPEED";TAB 5;"▄▄▄▄▄ ";A$(3 TO 1
9);" ▄▄"
 265 FOR I=10 TO 60
 266 PLOT I,32
 267 PRINT AT 0,14;("RE-FUELLING"
 AND I/2<>INT (I/2));("RE-FUELIN
G" AND I/2=INT (I/2))
 269 NEXT I
 270 PRINT AT 0,14;"           "
 271 POKE 16418,0
 273 PRINT AT 23,0;"HIT ""0"" TO
 DROP OUT OF ORBIT...."
 274 IF INKEY$<>"" THEN GOTO 274
 275 IF INKEY$<>"0" THEN GOTO 27
5
 276 PRINT AT 23,0;"
                           "
 280 IF H<=1000 THEN PLOT H/20+1
0,22
 290 IF H>1000 THEN PRINT AT 10,
31;"█"
 300 IF H<=100 THEN PLOT H/2+10,
12
 305 IF S=R THEN GOSUB 4000
 307 IF E=1 THEN LET Q=Q-1
 308 IF Q=0 AND E=1 THEN GOSUB 5
000
 310 LET F$=INKEY$
 330 IF F$<"0" OR F$>"9" OR E=1
THEN LET F$="0"
 340 LET F=2*VAL F$
 350 IF F>U THEN LET F=U
 357 IF F<>0 THEN LET XX=USR 165
69
```

```
 360 LET U1=INT (U/10+10)
 370 LET U=U-F
 380 LET U2=INT (U/10+11)
 390 FOR P=U1 TO U2 STEP -1
 400 UNPLOT P,32
 410 NEXT P
 420 PRINT AT 3,21;("[redacted]" AND U<
=50);AT 3,21;("OUT" AND U=0)
 450 LET V=V-F/M+1
 470 LET S=S+1
 480 UNPLOT X,2
 490 LET X=36-V/LN (2+ABS V)
 500 PLOT X,2
 510 PRINT AT 10,31;" "
 520 IF H<=1000 THEN UNPLOT H/20
+10,22
 530 IF H<=100 THEN UNPLOT H/2+1
0,12
 540 LET H=H-V/2
 545 LET K=USR 16597+USR 16540
 550 IF H>0 THEN GOTO 0280
 560 CLS
 570 PRINT "SPEED= ";.1*INT (10*
V);" M/S, TIME= ";S;" SECS"
 580 PRINT ,,"FUEL REMAINING= ";
.1*INT (2*U);" PERCENT",,,,,
 590 LET D=INT ((V-0.1)/5)
 600 IF D<0 THEN LET D=0
 610 IF D=0 THEN PRINT AT 5,5;"S
AFE LANDING.";AT 6,5;"
"
 620 IF D=1 THEN PRINT AT 5,5;"B
UMPY LANDING.";AT 6,5;"
"
 630 IF D=2 THEN PRINT AT 5,5;"D
ANGEROUS LANDING.";AT 6,5;"
"
 640 IF D=3 THEN PRINT AT 5,5;"C
RASH LANDING.";AT 6,5;"
"
 650 IF D>3 THEN PRINT AT 5,5;"C
ATASTROPHIC LANDING.";AT 6,5;"
"
 660 IF D>10 THEN GOTO 1460
 670 IF D>6 THEN PRINT AT 7,1;"Y
OU HAVE MADE A ";INT (V/2);" FOO
T CRATER";TAB 5;"IN THE MOON""S
SURFACE."
 680 GOSUB 1930
 690 IF D=0 THEN GOTO 1650
 700 PRINT AT 12,14;" ";TAB 14;"
"
 720 FOR Z=12 TO 20
 730 PRINT AT Z-1,12;"  "
 740 PRINT AT Z,12;")/"
 750 NEXT Z
```

```
 753 IF D<>1 AND D<>2 THEN GOTO
758
 755 PRINT AT 15,5;"**OUCH**"
 756 FOR Z=1 TO 10
 757 NEXT Z
 758 PRINT AT 15,5;"          "
 760 IF D=1 THEN GOTO 2040
 770 PRINT AT 18,18;" ";TAB 17;"
 ";TAB 17;" ";AT 12,21;"
 810 LET Y=22
 820 FOR Z=13 TO 20
 830 PRINT AT Z-1,Y-.3;"   ";AT
Z,Y;"
 850 LET Y=Y+0.3
 860 NEXT Z
 880 IF D=2 THEN GOTO 2040
 890 LET Z$="          "
 895 LET K=14
 900 PRINT AT 12,K;Z$;TAB K;"
     ";AT 13,K;Z$;TAB K;"
 ";TAB K;"
 ";TAB K;"
 ";TAB K;"
 990 IF D=3 THEN GOTO 2040
1010 RAND USR 16514
1020 PRINT AT 18,K;"= =  === ="
1030 RAND USR 16514
1040 PRINT AT 18,K;"
 ";AT K,12;"
1060 PRINT TAB USR 16631;AT 13,1
1;" ";AT 10,13;" ";AT 13,11;"
AT 10,13;"";AT K,12;"";AT K,2
3;"";TAB USR 16631;AT 20,10;"
";TAB 21;"";AT K,17;"  ";AT 18
,11;"";AT K,22;"";AT 18,11;"
;AT K,22;"  ";TAB USR 16631;AT 15
,12;"";AT 17,24;"
AT 19,13;"  ";AT 17,24;"";AT 19,
13;"";TAB USR 16631;AT 15,12;"
";TAB 20;"  ";AT 15,12;"";
AT 19,23;"";AT 19,
23;"
1070 PRINT TAB USR 16631;AT 13,1
2;" ";AT 19,11;" ";AT 13,12;"
AT 19,11;"";AT 16,18;"";AT 2
0,13;"OO";AT 15,12;"
TAB USR 16631;AT 15,15;"";AT 18
,25;"";AT 15,15;" ";AT 18,25;"
";AT 16,K;"  ";TAB USR 16631;AT
 17,15;"";AT 18,15;"";AT 17,1
8;"";AT 20,18;"X";AT 18,18;"
";AT 17,20;"";AT 18,20;"";AT
20,20;"O";TAB USR 16631;AT 16,17
;" ";TAB 17;" ";TAB 16;" ";TAB 1
6;" ";TAB 16;"";AT 18,17;"  ";AT
 20,17;""
```

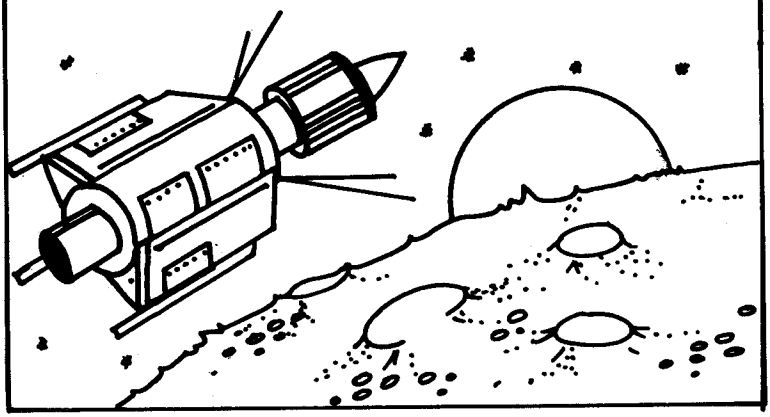
```
1350 FOR Z=1 TO 30
1360 NEXT Z
1370 FOR Z=1 TO 5
1380 PRINT AT 16,16;"R.I.P."
1390 FOR L=1 TO 8
1400 NEXT L
1410 PRINT AT 16,16;"       "
1420 FOR L=1 TO 8
1430 NEXT L
1440 NEXT Z
1450 GOTO 90
1460 PRINT AT 21,4;"[redacted]
[redacted]"
1470 PRINT AT 20,5;"[redacted]
[redacted]"
1480 PRINT AT 19,6;"[redacted]
[redacted]"
1490 PRINT AT 18,8;"[redacted]
"
1500 FOR Z=13 TO 18
1510 PRINT AT Z,13;":"
1520 FOR Y=1 TO 3
1530 NEXT Y
1540 NEXT Z
1550 FOR Z=1 TO 10
1560 PRINT AT 19,13;"■"
1570 PRINT AT 19,13;"*"
1580 NEXT Z
1600 PRINT AT 21,4;"[redacted]
[redacted]"
1610 PRINT AT 20,5;"[redacted]
[redacted]"
1620 PRINT AT 19,6;"[redacted]
[redacted]"
1630 PRINT AT 18,8;"[redacted]
"
1631 PRINT AT 7,1;"YOU HAVE MADE
 A ";INT (V/2);" FOOT CRATER
  IN THE MOON""S SURFACE."
1632 FOR Z=18 TO 13 STEP -1
1634 PRINT AT Z,13;" "
1635 FOR Y=1 TO 5
1636 NEXT Y
1637 NEXT Z
1638 PRINT AT 16,11;"R.I.P."
1640 GOTO 2040
1650 FOR Z=1 TO 30
1651 NEXT Z
1653 PRINT AT 13,16;"O"
1654 PRINT AT 13,16;"[redacted]"
1655 PRINT AT 13,16;"[redacted]";AT 11,1
5;" O "
1656 PRINT AT 10,15;" O ";AT 11,
15;"[redacted]";AT 13,16;"  "
1657 PRINT AT 9,15;" O ";AT 10,1
5;"[redacted]";AT 11,15;"[redacted]"
```

```
1670 FOR Z=1 TO 6
1680 PRINT AT 9,15;" O ";AT 10,1
5;"▀▀▀";AT 11,15;" ▀ "
1690 FOR Y=1 TO 2
1700 NEXT Y
1710 PRINT AT 9,15;"▝O▘";AT 10,1
5;" ▀ ";AT 11,15;"▀ ▀"
1720 FOR Y=1 TO 2
1730 NEXT Y
1740 NEXT Z
1750 PRINT AT 9,15;" O ";AT 10,1
5;"▀▀▀";AT 11,15;" ▀ "
1760 FOR Z=1 TO 20
1770 NEXT Z
1775 PRINT AT 10,15;" ";AT 10,15
;" "
1780 FOR Z=11 TO 9 STEP -1
1790 PRINT AT Z,21;"▌"
1800 FOR Y=1 TO 4
1810 NEXT Y
1820 NEXT Z
1830 FOR Z=1 TO 5
1840 NEXT Z
1850 PRINT AT 9,22;"▒";AT 10,22;
"▒"
1860 FOR Z=1 TO 5
1870 NEXT Z
1880 PRINT AT 9,23;"▒";AT 10,23;
"▒"
1890 FOR Z=1 TO 2
1900 NEXT Z
1905 RAND
1910 PRINT AT 9,24;"▒";AT 10,24;
"▒"
1911 IF RND>.5 THEN GOTO 3000
1913 FOR Z=1 TO 10
1914 PRINT AT 9,15;"▖";AT 10,15;
" "
1915 FOR Y=1 TO 3
1916 NEXT Y
1917 PRINT AT 9,15;" ";AT 10,15;
"▄"
1918 FOR Y=1 TO 3
1919 NEXT Y
1920 NEXT Z
1921 GOTO 90
1930 PRINT AT 12,14;"
1940 PRINT AT 13,14;"
1950 PRINT AT 14,14;"
1960 PRINT AT 15,14;"
1970 PRINT AT 16,14;"
1980 PRINT AT 17,14;"
1990 PRINT AT 18,14;"
2000 PRINT AT 19,14;"
2010 PRINT AT 20,14;"
2020 PRINT AT 21,10;
▒▒▒▒▒▒▒▒▒▒"
```



```
2030 RETURN
2040 FOR Z=1 TO 50
2050 NEXT Z
2060 GOTO 90
3000 LET X=22
3005 LET Y=9
3008 FOR Z=1 TO 27
3010 PRINT AT 10,Z;" (**)"
3015 IF Z=3 THEN PRINT AT 8,16;"
?"
3020 IF Z=5 THEN PRINT AT 8,15;"
GULP"
3030 IF Z=8 THEN PRINT AT 8,15;"
    ";AT 9,15;"   ";AT 10,15;" O
  "
3040 IF Z=9 THEN PRINT AT 10,15;
"   ";AT 11,15;" O ";AT 13,16;"
 "
3045 IF Z=16 THEN PRINT AT 8,15;
"PHEW"
3050 IF Z=17 THEN PRINT AT 8,15;
"   ";AT 10,15;" O ";AT 11,15;"
▀▀";AT 13,16;"  "
3060 IF Z=18 THEN PRINT AT 9,15;
"  O ";AT 10,15;"▀▀▀";AT 11,15;"▀
▀ "
3070 IF Z=24 THEN PRINT AT 9,21;
"
3072 IF Z>24 THEN PRINT AT Y,X;"
   "
3075 IF Z>24 THEN LET X=X+0.3
3080 IF Z>24 THEN LET Y=Y+1.3
3090 PRINT AT Y,X;"▒▒▒▒"
3100 NEXT Z
3105 CLS
3110 FOR Y=1 TO 20
3115 NEXT Y
3120 GOTO 0090
4000 LET 0=INT (RND*6)+3
4010 PRINT AT 0,11;"FUEL FLUSH";
AT 1,11;"FUEL OUT FOR ";0;" SECS
"
4020 LET E=1
4040 RETURN
5000 PRINT AT 0,11;"           ";
AT 1,11;"           "
5005 LET E=0
5010 RETURN
6000 IF PEEK 16637=201 THEN RETU
RN
6010 LET A$="042 012 064 006 022
 197 006 032 035 126 198 128 119
 016 249 035 193 016 242 201 "
6020 LET B$="042 012 064 017 010
 000 026 126 060 119 254 038 204
 172 064 201 054 028 043 205 163
 064 201 "
```

```
6030 LET C$="042 012 064 017 201
 000 025 017 066 000 054 057 025
 054 045 025 054 055 025 054 058
 025 064 056 025 054 057 201 042
 012 064 017 "
6040 LET C$=C$+"201 000 025 017
066 000 054 000 025 054 000 025
054 000 025 054 000 025 054 000
025 054 000 201 "
6060 LET D$="205 130 064 001 000
 000 201 "
6070 FOR N=16514 TO 16533
6080 POKE N,VAL A$( TO 3)
6090 LET A$=A$(5 TO )
6100 NEXT N
6110 FOR N=16540 TO 16562
6120 POKE N,VAL B$( TO 3)
6130 LET B$=B$(5 TO )
6140 NEXT N
6150 FOR N=16569 TO 16624
6160 POKE N,VAL C$( TO 3)
6170 LET C$=C$(5 TO )
6180 NEXT N
6190 FOR N=16631 TO 16637
6200 POKE N,VAL D$( TO 3)
6210 LET D$=D$(5 TO )
6220 NEXT N
6230 RETURN
9000 SAVE "MISSION APOLL"
9010 RUN
```

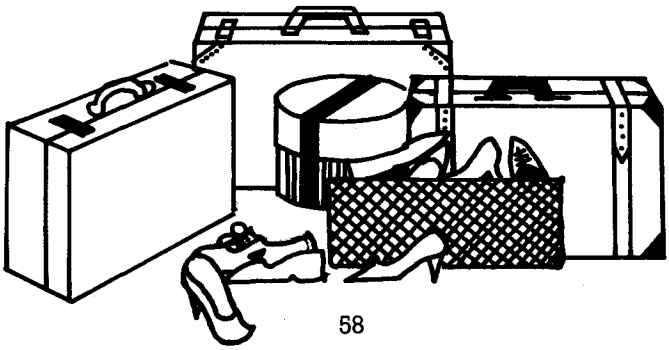


JUNKSHOP

Why not grab anything that you don't want and take a stroll down to your local junk shop?

This program, which fits into 1K, allows you to try to sell anything that you wish to get rid of to the owner of the junk shop. You must haggle over the price with him, he is a very stubborn man and will not accept anything valued at over £200.

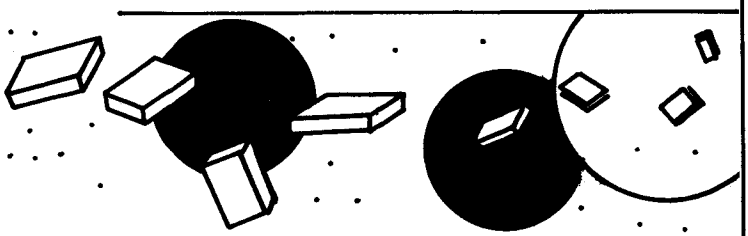
```
   10 RAND
   20 CLS
   30 LET D=0
   40 PRINT "WHAT DO YOU WANT TO
SELL?"
   50 INPUT A$
   60 IF RND<.6 THEN GOTO 90
   70 PRINT ,,"SORRY, I AM NOT IN
TERESTED IN   ";A$+("S" AND A$(L
EN A$)<>"S")
   75 FOR A=1 TO 80
   80 NEXT A
   85 RUN
   90 CLS
  100 PRINT "HOW MUCH ARE YOU ASK
ING?"
  110 INPUT A
  120 CLS
  130 IF A>100+RND*100 THEN GOTO
70
  140 LET C=A
  150 LET B=D+INT (RND*(A-D))+2
  170 IF B>=A THEN GOTO 1000
  180 PRINT "I WILL GIVE YOU £";B
  190 PRINT "WHAT DO YOU SAY?"
  200 INPUT B$
  210 CLS
  220 IF B$="YES" THEN GOTO 1010
  230 IF VAL B$>C-RND*C/3 OR VAL
B$>A THEN GOTO 70
  240 LET A=VAL B$
  250 LET D=B
  260 GOTO 150
1000 LET B=A
1010 CLS
1020 PRINT "I WILL TAKE YOUR ";A
$
1025 PRINT "FOR £";B
1030 FOR A=1 TO 80
1040 NEXT A
1050 RUN
1100 SAVE "JUNKSHOP"
1110 RUN
```



SPACE SALVAGE

A huge explosion threw a cruiser's valuable cargo of 2,500 energy modules into deep space. You, as Commander of the humble Space Debris Collector SDC81 must collect as many of these modules as you can. Avoid the black holes which contain anti-matter and are attracted to the energy modules. Some of the units have been thrown into different time warps. You must fly through as many of these time warps as possible: the length of your mission will be increased as a result. You control your ship with the up and down cursor keys; to brake hold down the 'O' key (which loses you a few points). To pause for a long time put the Time-Lock into operation by pressing 'P'. To continue, press any key.

The energy modules are shown as Os and the black holes are represented as inverse Os. The warp gate is an inverse space and your ship is shown as a 'greater than' sign.



```
  10 LET X=2
  15 LET Y=2
  20 LET SC=0
  30 LET N=1
  40 LET W=1
  45 CLS
  50 PRINT AT 6,8;"SPACE SALVAGE
"
  55 PRINT ,,,,"7 TO GO UP,
  6 TO GO DOWN,"
  58 PRINT ,,"0 FOR RETROS    P
FOR TIME-LOCK"
  59 PRINT ,,,,"PRESS S TO START
"
  60 FOR T=30 TO 1 STEP -1
  65 PRINT AT 16,12;"TIME: ";T
  67 IF T<10 THEN PRINT AT 16,16
;" ";T
  68 IF INKEY$="S" THEN GOTO 90
  70 NEXT T
  90 CLS
 155 PRINT AT 0,0;"
"
 160 FOR S=1 TO 18
 165 PRINT AT S,0;"■"
 170 NEXT S
 175 PRINT AT 19,0;"
"
 177 REM
 180 FOR T=1 TO 20
 185 LET A=INT (RND*22)+5
 190 LET B=INT (RND*17)+2
 195 PRINT AT B,A;"O"
 197 PRINT AT B,A+INT (RND*4);"■
"
 198 NEXT T
 199 PRINT AT B,A;"■"
 205 IF INKEY$="P" THEN GOSUB 40
0
 210 IF INKEY$="0" THEN LET Y=Y-
2
 212 IF INKEY$="0" THEN LET SC=S
C-4
 213 LET Y=Y+1
 215 LET X=X+(INKEY$="6")-(INKEY
$="7")
 217 IF X<1 THEN LET X=1
 219 IF X>18 THEN LET X=18
 222 IF Y>31 THEN GOTO 300
 228 PRINT AT X,Y;">";
 229 IF INKEY$="Z" THEN COPY
 230 PRINT AT X,Y;" ";
 240 IF X=B AND Y=A THEN GOSUB 5
00
 250 IF PEEK (PEEK 16398+256*PEE
K 16399)=52 THEN GOSUB 700
```

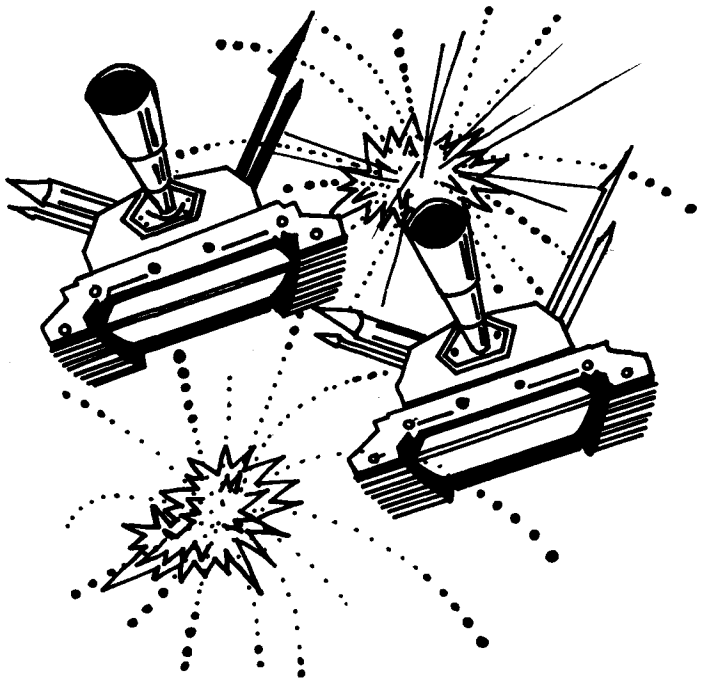
```
 260 IF PEEK (PEEK 16398+256*PEE
K 16399)=180 THEN GOTO 1000
 280 GOTO 205
 300 LET Y=1
 305 CLS
 310 PRINT AT 0,0;"
"
 315 LET N=N+1
 317 IF N=4*W THEN GOTO 1000
 320 PRINT AT 19,0;"
"
 330 PRINT AT 20,0;"
";N
;"
";SC;"
";W;"
"
 345 PRINT AT 21,0;"
"
 360 GOTO 177
 400 PRINT AT 21,5;"
"
 405 PRINT AT X,Y;"
"
 410 PAUSE 4E4
 415 PRINT AT X,Y;" "
 420 PRINT AT 21,5;"
"
 430 RETURN
 500 LET W=W+1
 510 LET SC=SC+40*W
 520 FOR T=1 TO 50
 530 PRINT AT 21,8;"FOUND WARP G
ATE"
 540 PRINT AT 21,8;"
"
 550 NEXT T
 560 RETURN
 700 PRINT AT X,Y+1;"
"
 710 LET SC=SC+11+INT (RND*8)
 715 PRINT AT X,Y+1;"
"
 730 PRINT AT X,Y+1;" "
 740 RETURN
1000 CLS
1050 LET S$="
                    
                  
                    
"
1065 LET T$="
"
1070 FOR T=1 TO 12
1075 LET A=INT (RND*16)
1077 LET B=INT (RND*26)
1080 PRINT AT A,B;S$
1082 FOR G=1 TO 4
1083 NEXT G
1085 PRINT AT A,B;T$;T$;T$;T$;T$
1090 NEXT T
1100 CLS
1110 PRINT AT 6,10;"YOU SCORED "
```

```
;SC
1115 PRINT ,,,,"YOUR COMMANDER R
ATING IS;"
1120 IF SC<200 THEN LET R$="POOR
 AMATEUR"
1125 IF SC>200 AND SC<900 THEN L
ET R$="NOT BAD"
1130 IF SC>900 THEN LET R$="**EX
CELLENT**"
1135 PRINT ,R$
1140 PRINT ,,,,,,"PRESS ANY KEY
TO TRY AGAIN"
1150 IF INKEY$="" THEN GOTO 1150
1160 RUN
1500 SAVE "SPACE SALVAGE"
1510 RUN
```

BAZOOKA!

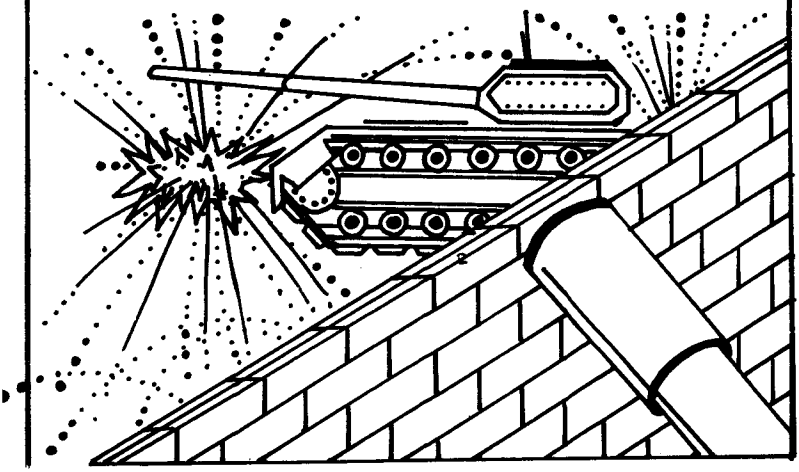
You lie in wait behind a pile of sandbags with your anti-tank gun loaded and ready for use. An enemy tank appears on the horizon, but a large wall is in your line of fire. You raise the turret of your gun and FIRE! The shell arcs over the wall and penetrates the tank's armour reducing it to a mass of mangled metal. There is little time for praise, however, as another tank comes into view.

In this game you must judge the angle and velocity needed for your shell to clear the wall and hit the tank. The shell is fired and plotted according to genuine formulae as it tries to hit the realistic-looking tank on the other side of the wall.



```
  10 RAND
  15 CLS
  18 LET TRY=0
  20 DIM A$(32)
  30 LET WALL=5+INT (RND*18)
  40 LET HGT=3+INT (RND*6)
  50 LET TANK=WALL+2+INT (RND*(2
7-WALL))
  60 PRINT AT 19,TANK;"--▄";AT 2
0,TANK;"(▒▒▒)";AT 21,TANK;" OO"
  70 FOR A=1 TO HGT
  80 PRINT AT 22-A,WALL;"Z"
  90 NEXT A
 100 PRINT AT 20,1;"▬▬";AT 21,0;
"▒)"
 110 PRINT AT 8,3;"ENTER ANGLE O
F PROJECTION"
 120 INPUT ANG
 130 IF ANG<1 OR ANG>90 THEN GOT
O 120
 140 PRINT AT 8,0;A$
 150 LET X=1+(ANG>10)+(ANG>33)+(
ANG>55)+(ANG>77)
 160 PRINT AT 20,1;("▬▬" AND X=1
)+("▄▬" AND X=2)+("▄ " AND X=3)+
("▌ " AND X>3)
 170 PRINT AT 19,1;(" ▄" AND X=3
)+(" ▌" AND X=4)+("▌ " AND X=5)
 180 PRINT AT 8,6;"ENTER SHELL V
ELOCITY"
 190 INPUT VEL
 200 IF VEL<10 THEN GOTO 190
 202 LET TRY=TRY+1
 205 LET VEL=VEL/4
 210 PRINT AT 8,0;A$
 220 LET ANG=ANG*PI/180
 230 LET HORI=COS ANG*VEL
 240 LET VERT=SIN ANG*VEL
 243 LET V=63
 246 LET W=43
 250 LET T=0
 260 LET X=6+HORI*T
 270 LET Y=6+VERT*T-4.9*T**2
 273 IF Y>43 THEN UNPLOT V,W
 275 IF Y>43 THEN GOTO 320
 277 IF X>63 THEN GOTO 410
 280 PLOT X,Y
 290 UNPLOT V,W
 300 LET V=X
 310 LET W=Y
 320 LET T=T+.2
 330 IF Y<2 THEN GOTO 360
 340 IF X>WALL*2-2 AND X<WALL*2+
3 AND Y<=HGT*2+1 THEN GOTO 360
 350 GOTO 260
 360 UNPLOT V,W
```

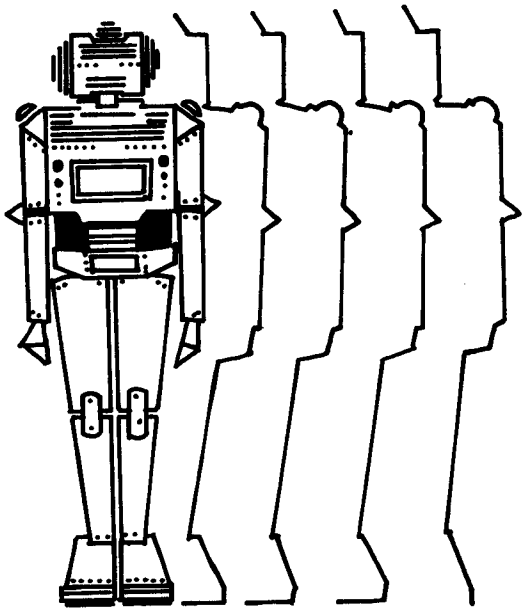
```
 370 IF Y<2 THEN GOTO 400
 380 PRINT AT 8,5;"OOPS, YOU HIT
 THE WALL"
 390 GOTO 420
 400 IF X>=TANK*2 AND X<=TANK*2+
7 THEN GOTO 470
 410 UNPLOT V,W
 415 PRINT AT 8,9;"YOU MISSED IT
"
 420 PRINT AT 10,4;"HIT ANY KEY
TO TRY AGAIN"
 430 IF INKEY$="" THEN GOTO 430
 440 PRINT AT 19,1;"          "
 450 PRINT AT 8,0;A$;AT 10,0;A$
 460 GOTO 100
 470 FOR N=1 TO 15
 480 PRINT AT 20,TANK+1;"■ ";AT
20,TANK+1;" ■"
 490 NEXT N
 495 CLS
 500 PRINT AT 8,4;"GREAT SHOT. T
HAT TOOK YOU"
 505 PRINT AT 10,12;TRY;" GOES."
 508 PRINT AT 12,9;"PRESS ANY KE
Y."
 510 IF INKEY$="" THEN GOTO 510
 520 RUN
 560 SAVE "TANK ATTAC■"
 570 RUN
```



DROID

Here is your chance to program a real robot, a small droid, to reach the top of the maze. You must enter the correct commands for the droid, using L for left, R for right, U for up and D for down. Every instruction you give the droid will move him one space in the appropriate direction and if you make any mistakes in the initial entry just type O to rub out. You can have a maximum of 60 instructions, as the droid's memory is very small, and after every 20 instructions the screen clears so you cannot see what you have just put in. You have to rely on your mental dexterity and memory to get the droid to the top of the maze safely. When you have finished typing in your instructions, press the '9' key and watch him move. If he hits a wall along the way, then he is destroyed.

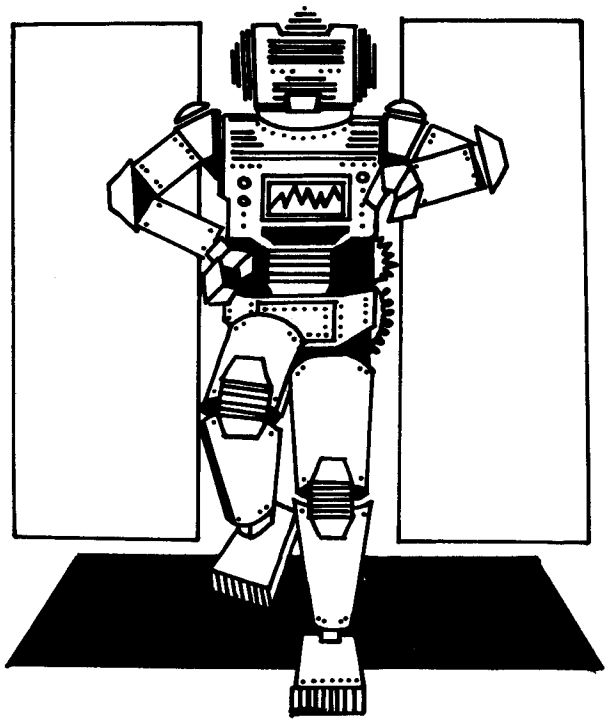
Happy programming!



```
    5 DIM S$(60)
   10 CLS
   15 RAND
   20 GOSUB 1000
   22 LET SC=0
   24 LET I=0
   25 PRINT AT 21,5;"><"
   30 FOR N=1 TO 20
   40 IF INKEY$<>"" THEN GOTO 40
   42 LET A$=INKEY$
   43 IF A$="9" THEN GOTO 90
   44 IF A$="0" THEN GOTO 500
   46 IF A$<>"L" AND A$<>"R" AND
A$<>"U" THEN GOTO 42
   50 LET S$(I+N)=A$
   60 PRINT AT 21,5+N;S$(I+N);"<"
   70 NEXT N
   80 PRINT AT 21,5;"
                     "
   85 LET I=I+20
   86 IF I=60 THEN GOTO 2150
   88 GOTO 25
   90 PRINT AT 21,5;" ";AT 21,5+N
;" "
   91 POKE 16418,0
   92 LET I=0
   93 PRINT AT 23,11;"SCORE = 0"
   94 PRINT AT 21,6;S$(I+1 TO I+2
0)
  100 LET N=1
  105 IF S$(I+N)=" " THEN GOTO 20
0
  110 LET X=X+(S$(N+I)="R")-(S$(N
+I)="L")-33*(S$(N+I)="U")
  120 IF S$(N+I)="U" THEN LET SC=
SC+6
  122 LET SC=SC-(SC<>0)
  125 PRINT AT 23,19;SC
  130 PRINT AT 20,4+N;"■";AT 22,
4+N;"■"
  140 IF PEEK X<>0 THEN GOTO 2000
  150 POKE X,52
  160 LET N=N+1
  165 IF INKEY$<>"" THEN GOTO 165
  170 IF N<>21 THEN GOTO 105
  175 LET I=I+20
  180 PRINT AT 20,25;"■";AT 22,25
;" "
  190 GOTO 94
  200 FOR N=1 TO 50
  210 NEXT N
  220 CLS
  230 PRINT AT 8,2;"YOU DID NOT G
IVE THE ROBOT"
  240 PRINT AT 10,2;"ENOUGH INFOR
```

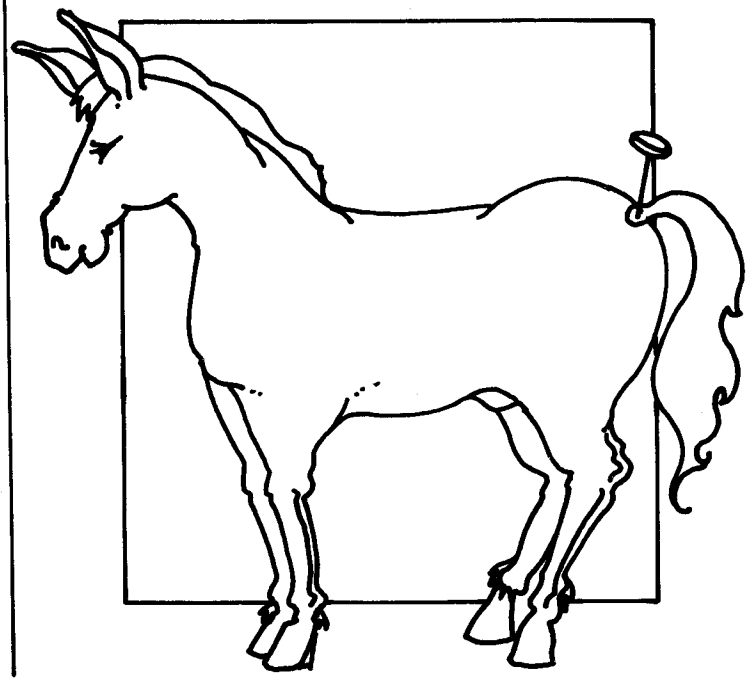
```
MATION TO REACH"
 250 PRINT AT 12,5;"THE TOP OF T
HE SCREEN."
 255 PRINT AT 12,7;"YOUR SCORE W
AS ";SC
 260 PRINT AT 16,5;"ANOTHER GAME
 ? (Y/N)"
 270 LET A$=INKEY$
 280 IF A$="" OR A$<>"Y" AND A$<
>"N" THEN GOTO 270
 290 IF A$="Y" THEN RUN
 300 STOP
 500 IF N=1 THEN GOTO 40
 502 LET N=N-1
 505 PRINT AT 21,5+N;"<  "
 510 GOTO 40
1000 PRINT AT 0,0;"
1010 PRINT AT 20,0;"
1020 FOR N=1 TO 17 STEP 2
1030 PRINT AT N,0;"";AT N,31;"
"
1040 PRINT AT N+1,0;"
1050 NEXT N
1060 PRINT AT 19,0;"";AT 19,31;
""
1070 FOR N=2 TO 18 STEP 2
1080 LET Y=1
1090 LET Y=Y+INT (RND*15)+3*(Y<>
1)
1100 LET Z=INT (RND*2)
1110 IF Y>29 THEN GOTO 1140
1120 PRINT AT N,Y;(" " AND Z=0)+
(" " AND Z=1)
1130 GOTO 1090
1140 NEXT N
1150 LET DF=PEEK 16396+256*PEEK
16397
1160 LET X=DF+642
1170 POKE X,52
1180 RETURN
2000 IF PEEK X=131 THEN GOTO 210
0
2005 FOR N=1 TO 10
2010 POKE X,133
2015 LET Z=9*9*9*9*9*9
2020 POKE X,3
2025 LET Z=9*9*9*9*9*9
2030 POKE X,5
2035 LET Z=9*9*9*9*9*9
2040 POKE X,131
2045 LET Z=9*9*9*9*9*9
2050 NEXT N
2060 CLS
```

```
2070 PRINT AT 8,2;"THE ROBOT HAS
 HIT A WALL AND"
2080 PRINT AT 10,9;"IS NOW IN BI
TS"
2090 GOTO 255
2100 LET L=I+N
2110 CLS
2120 PRINT AT 8,2;"YOUR ROBOT GO
T THROUGH THE"
2130 PRINT AT 10,8;"MAZE IN ";L;
" MOVES."
2140 GOTO 255
2150 CLS
2160 PRINT AT 8,1;"YOU HAVE BURN
T OUT THE ROBOT""S"
2170 PRINT AT 10,13;"MEMORY."
2180 GOTO 260
5000 SAVE "DROID"
5010 RUN
```

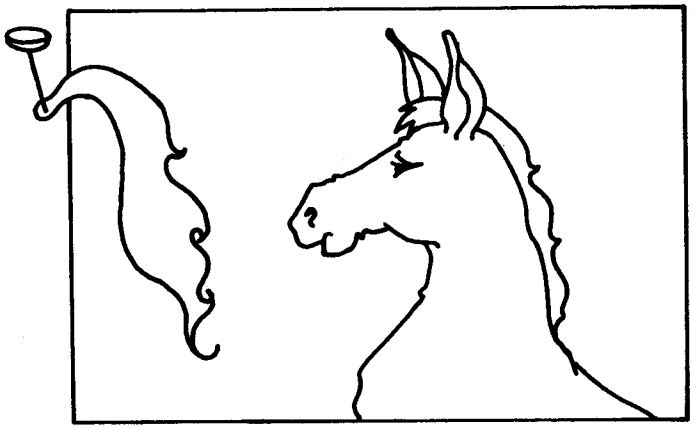


PIN THE TAIL ON THE DONKEY

In this simple, but fun game you must guess the position of the donkey's tail. The donkey will appear on the screen for a short space of time and then disappear. You will be asked to enter the horizontal and the vertical co-ordinates and where you think the donkey's tail should be. You can have up to four guesses. This games fits into 1K.



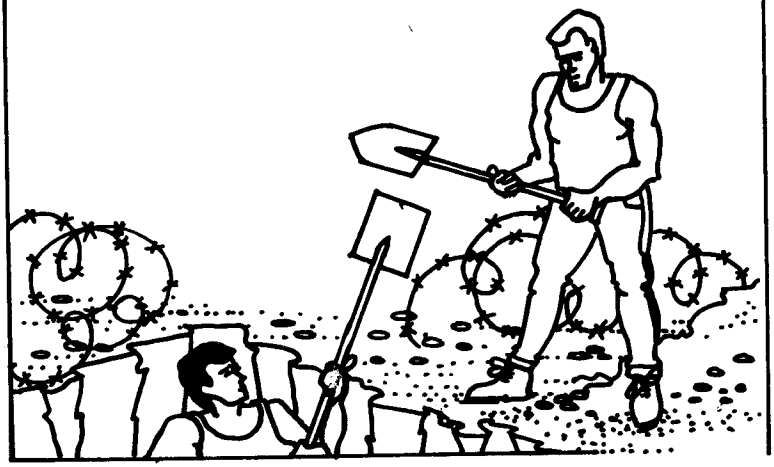
```
  5 RAND
 10 LET A=0
 20 LET B=A
 30 LET N=A
 40 LET Y=INT  (RND*25)+1
 50 LET X=INT  (RND*18)+2
 60 CLS
 70 LET A$="▄▀"
 80 LET B$="▀▄"
 90 PRINT AT X,Y;A$;AT X+1,Y;B$
;AT A,B;"/"
100 IF A=X AND B=Y+4 THEN GOTO
300
105 IF N=4 THEN GOTO 250
110 FOR T=1 TO 20
120 NEXT T
130 CLS
140 PRINT "TAIL POSITION"
150 PRINT ,,"VERTICAL POSITION
2-19"
155 INPUT A
160 PRINT ,,"HORIZONTAL POSITIO
N 5-30"
165 INPUT B
170 LET N=N+1
180 GOTO 60
250 PRINT AT X-2,2;"YOU LOSE"
260 PRINT ,,X;"  ";Y+4
270 STOP
300 PRINT AT X-2,2;"CORRECT, YO
U TOOK ";N;" GOES";AT X-2,2;"▒▒▒
▒▒▒▒"
310 IF INKEY$="" THEN GOTO 300
320 RUN
500 SAVE "DONKE▒"
510 RUN
```



DIGGER

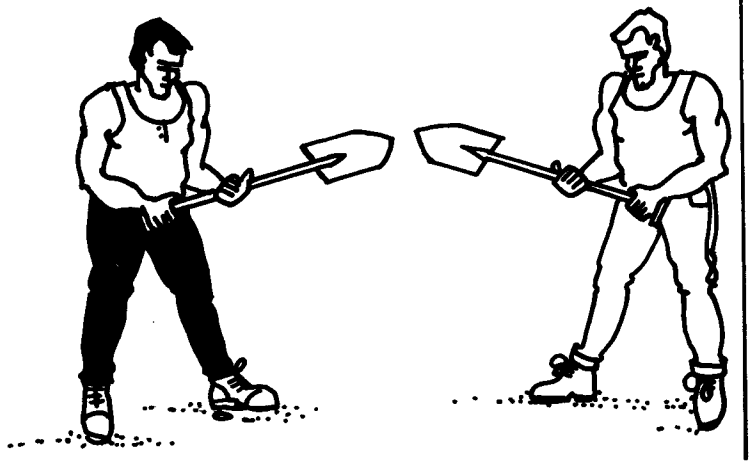
In this exciting two-player game, you and your opponent are digging trenches all over the screen. You must try and make your opponent fall into your trench or indeed fall back into his own trench. To make matters more difficult, there is a boundary around the edge of the screen which is electrified and if you hit this boundary you lose the point to your opponent. There is no time or score limit on this game. You use either end of the keyboard (this is easier if you own a professional keyboard) with the left half of the top row used to move the left digger up and the right half to move the right digger up. The second row is similarly split with the keys used to move the diggers down. The third row of keys moves the diggers to the right while the bottom row moves the diggers to the left.

Because of the number of keys involved, the program uses a machine code routine to scan the keyboard.



```
   1 REM @@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@
   5 IF PEEK 16579=201 THEN GOTO
 32
  10 LET A$="205 187 002 237 075
 130 064 203 069 032 002 006 000
 203 077 032 002 006 001 203 085
 032 002 006 002 203 093 032 002
 006 003 203 "
  12 LET A$=A$+"101 032 002 014
000 203 109 032 002 014 001 203
117 032 002 014 002 203 125 032
002 014 003 121 050 130 064 120
050 131 064 201 "
  15 FOR N=16516 TO 16579
  20 POKE N,VAL A$( TO 3)
  25 LET A$=A$(5 TO )
  30 NEXT N
  32 LET LFT=0
  34 LET RGT=0
  36 POKE 16418,0
  40 FOR N=0 TO 31
  50 PRINT AT 0,N;"■";AT 23,N;"■
"
  60 IF N<24 THEN PRINT AT N,0;"
■";AT N,31;"■"
  70 NEXT N
  80 POKE 16418,2
  90 LET DF=PEEK 16396+256*PEEK
16397
 100 LET X=DF+373
 110 LET Y=DF+384
 120 POKE 16514,3
 130 POKE 16515,1
 140 LET B=PEEK 16514
 150 LET A=PEEK 16515
 160 LET X=X+(A=1)-(A=0)+33*(A=2
)-33*(A=3)
 170 LET Y=Y+(B=2)-(B=3)+33*(B=1
)-33*(B=0)
 180 IF PEEK X=128 OR PEEK Y=128
 THEN GOTO 1000
 190 POKE X,128
 200 POKE Y,128
 210 RAND USR 16516
 220 GOTO 140
1000 IF PEEK X=128 THEN GOTO 105
0
1010 POKE Y,23
1020 LET A$="LEFT PLAYER WINS"
1030 LET LFT=LFT+1
1040 GOTO 1150
1050 IF PEEK Y=128 THEN GOTO 110
0
1060 POKE X,23
```

```
1070 LET A$="RIGHT PLAYER WINS"
1080 LET RGT=RGT+1
1090 GOTO 1150
1100 POKE X,23
1110 POKE Y,23
1120 LET A$="THAT WAS A DRAW"
1130 LET LFT=LFT+1
1140 LET RGT=RGT+1
1145 FOR N=1 TO 60
1150 NEXT N
1152 LET B$="■
                ■"
1155 FOR N=6 TO 16
1160 PRINT AT N,0;B$
1162 NEXT N
1164 PRINT AT 8,8;A$
1170 PRINT AT 12,8;"LEFT";AT 12,
17;"RIGHT"
1180 PRINT AT 14,9;LFT;AT 14,19;
RGT
1190 FOR N=1 TO 100
1200 NEXT N
1210 CLS
1220 GOTO 38
```



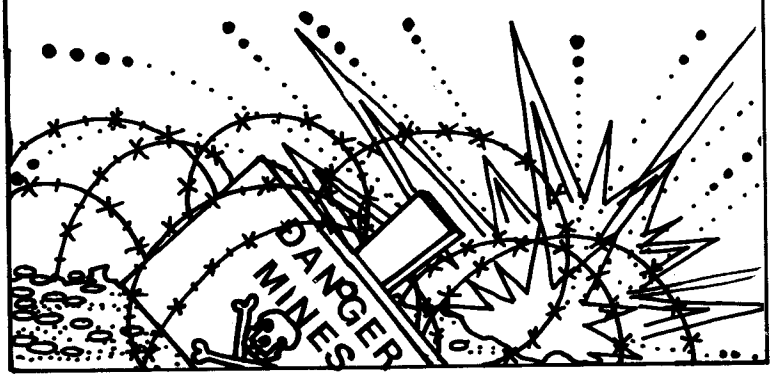
MINEFIELD

You must bravely try to escape from the minefield in this strategy game written for the ZX81 by Simon Gould. Use the cursor keys to move carefully around the screen but take note of the warning sign.

At the beginning of the game, you must input the number of mines that will be planted in the minefield. Any number less than 35 can be used. If you tread on a mine you will explode and the locations of the other mines will be shown. There is a timer and an on-screen scoring feature.

```
  5 LET SC=0
 10 LET A$=" O
                  "
 20 RAND
 30 CLS
 35 POKE 16418,2
 40 PRINT "  INPUT NO. OF MINES
";
 50 INPUT MNS
 55 PRINT "=";MNS
 60 DIM M(MNS,2)
 70 FOR N=1 TO MNS
 80 LET M(N,1)=INT (RND*20)+1
 90 LET M(N,2)=INT (RND*32)
100 NEXT N
105 CLS
106 POKE 16418,0
110 PRINT "TIMER=          EXIT
  SCORE=";SC
120 LET T=0
130 LET X=23
140 LET Y=INT (RND*28)
150 PRINT AT X-2,Y-1;A$
160 LET T=T+1
170 PRINT AT 0,6;T
180 IF T=100 THEN GOTO 1000
190 GOSUB 300
191 LET A=X
192 LET B=Y
195 LET B$=INKEY$
196 IF B$="" THEN GOTO 150
200 LET X=X+(B$="6" AND X<23)-(
```

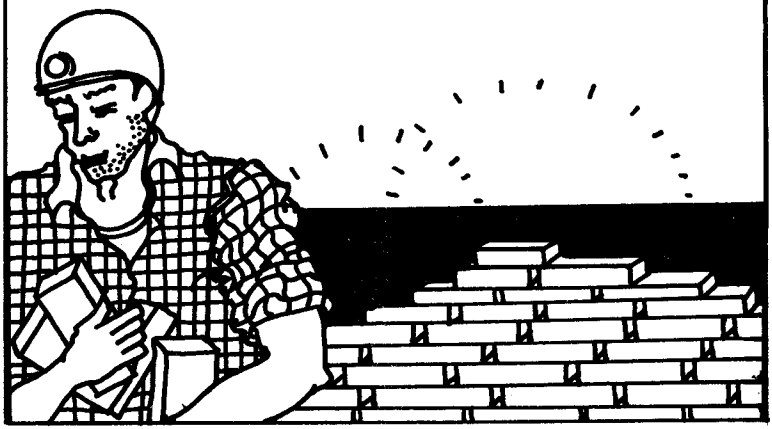
```
B$="7" AND X>3)
 210 LET Y=Y+(B$="8" AND Y<30)-(
B$="5" AND Y>0)
 220 PRINT AT A-2,B-1;"
                                  "
 230 GOTO 150
 300 PRINT AT 0,11;"┌─ EXIT ─┐"
 310 IF X=3 AND Y>11 AND Y<20 TH
EN GOTO 2000
 320 LET N=1
 330 LET XC=M(N,1)-X
 340 LET YC=M(N,2)-Y
 350 IF ABS XC<3 AND ABS YC<3 TH
EN PRINT AT 0,11;"┌──────┐"
 360 IF XC=0 AND YC=0 THEN GOTO
1000
 365 LET N=N+1
 370 IF N=MNS THEN RETURN
 380 GOTO 330
1000 PRINT AT X-2,Y-1;"■■■
                       ■■■
                       ■■■"

1010 FOR N=1 TO MNS
1020 PRINT AT M(N,1),M(N,2);"#"
1030 NEXT N
1040 FOR N=0 TO 100
1050 NEXT N
1060 RUN
2000 LET SC=SC+100
2010 PRINT AT 10,5;"└──────────┘
└──────┘"
2020 PRINT AT 12,5;"STAND BY FOR
 NEXT FIELD"
2030 FOR N=0 TO 80
2040 NEXT N
2050 GOTO 105
```



GREEDY GRANNIS

Up in the hills lives an old miner who missed the gold rush but still believes that there is gold nearby. One day he finds a passageway that leads into a large maze. He cannot believe his luck, on the floor lie lots of gold bars! He hurriedly picks up the bars nearest him and starts to limp his way round the twisting maze. But trouble lurks nearby. Greedy Grannis, the Troll of the Maze, is hungry for human flesh. He smells the miner from some distance away, and although he too is getting old his knowledge of the maze makes up for his lack of speed as he pursues the aged man. Can the old man survive...?

In this game, you control the old miner as he is chased around the maze, trying to collect as many gold bars as he can. You use the cursor keys to move and if you manage to clear a complete screenful, you will be awarded a bonus — the appearance of another screenful of gold.

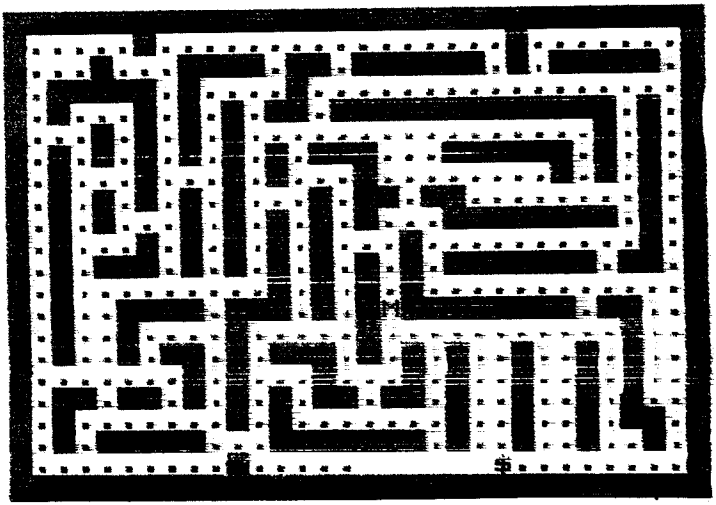


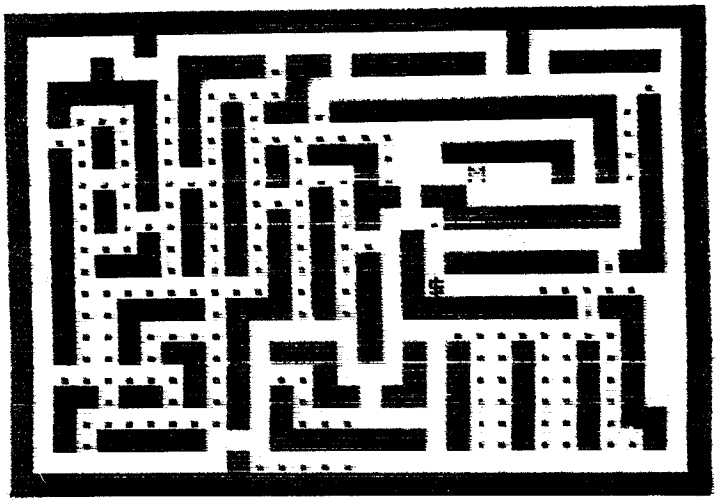
```
   1 REM 000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000
   5 IF PEEK 16637=201 THEN GOTO
 48
  10 LET A$="062 000 050 130 064
 042 012 064 017 033 000 058 131
 064 071 005 025 016 253 058 132
 064 071 035 016 253 126 254 128
 032 004 006 "
  15 LET A$=A$+"001 024 002 006
000 025 229 025 126 254 128 032
004 014 001 024 002 014 000 225
120 185 192 043 126 254 128 032
006 062 000 185 "
  20 LET A$=A$+"192 024 004 062
001 185 192 035 035 126 254 128
032 006 062 000 185 192 024 004
062 001 185 192 062 001 050 130
064 201 042 012 "
  25 LET A$=A$+"064 017 035 000
025 001 146 002 126 254 027 200
035 011 120 177 032 246 062 001
050 133 064 201 "
  30 FOR A=16518 TO 16637
  35 POKE A,VAL A$( TO 3)
  40 LET A$=A$(5 TO )
  45 NEXT A
  48 LET SC=0
  50 POKE 16514,1
  60 POKE 16517,0
  90 LET A=-1
 100 LET DF=PEEK 16396+256*PEEK
16397
 110 LET B=DF+347
 120 LET C=DF+677
 130 LET D=27
 140 LET J=20
 150 LET K=17
 160 LET X=10
 170 LET Y=17
 180 GOSUB 3000
 190 RAND
 200 IF PEEK B<>50 THEN GOTO 100
0
 210 POKE B,D
 220 LET B=B+A
 230 LET D=PEEK B
 240 POKE B,50
 250 IF ABS A=1 THEN LET Y=Y+A
 260 IF ABS A=33 THEN LET X=X+A/
33
 270 IF PEEK C<>13 THEN GOTO 100
0
```

```
 280 LET L=C
 290 LET A$=INKEY$
 300 IF A$<>"5" OR PEEK (C-1)=12
8 THEN GOTO 340
 310 LET C=C-1
 320 LET K=K-1
 330 GOTO 450
 340 IF A$<>"8" OR PEEK (C+1)=12
8 THEN GOTO 380
 350 LET C=C+1
 360 LET K=K+1
 370 GOTO 450
 380 IF A$<>"6" OR PEEK (C+33)=1
28 THEN GOTO 420
 390 LET C=C+33
 400 LET J=J+1
 410 GOTO 450
 420 IF A$<>"7" OR PEEK (C-33)=1
28 THEN GOTO 450
 430 LET C=C-33
 440 LET J=J-1
 450 POKE L,0
 455 IF PEEK C=27 THEN LET SC=SC
+3
 460 POKE C,13
 470 POKE 16515,X
 480 POKE 16516,Y
 490 LET Z=USR 16518
 500 LET Z=USR 16612
 510 IF PEEK 16517=1 THEN GOTO 2
000
 520 IF PEEK 16514=1 THEN GOTO 2
00
 530 LET Z=RND
 540 IF Z<.5 THEN GOTO 580
 550 LET A=(J>X)-(J<X)
 555 IF A<>0 AND PEEK (B+A*33)<>
128 THEN GOTO 568
 560 IF Z>=.5 THEN GOTO 580
 565 LET A=(PEEK (B+33)<>128)-(P
EEK (B-33)<>128)
 568 LET A=A*33
 570 GOTO 200
 580 LET A=(K>Y)-(K<Y)
 585 IF A<>0 AND PEEK (B+A)<>128
 THEN GOTO 200
 590 IF Z<.5 THEN GOTO 550
 595 LET A=(PEEK (B+1)<>128)-(PE
EK (B-1)<>128)
 600 GOTO 200
1000 CLS
1010 PRINT AT 8,2;"YOU HAVE JUST
 BEEN EATEN BY"
1020 PRINT AT 10,9;"GREEDY GRANN
IS."
1025 PRINT AT 12,10;"YOU SCORED
```

```
";SC
1030 PRINT AT 14,2;"HIT ANY KEY
TO MAKE HIM SPIT"
1040 PRINT AT 16,12;"YOU OUT."
1045 IF INKEY$<>"" THEN GOTO 104
5
1050 IF INKEY$="" THEN GOTO 1050
1060 RUN
2000 CLS
2005 LET SC=SC+500
2010 PRINT AT 8,1;"YOU HAVE MANA
GED TO COLLECT ALL"
2020 PRINT AT 10,6;"OF THE GOLD
NUGGETS."
2030 PRINT AT 12,6;"YOU HAVE SCO
RED ";SC
2040 PRINT AT 14,2;"PRESS ANY KE
Y IF YOU DARE TO"
2050 PRINT AT 16,11;"CONTINUE."
2055 IF INKEY$<>"" THEN GOTO 205
5
2060 IF INKEY$="" THEN GOTO 2060
2070 GOTO 50
3000 CLS
3010 PRINT "
"
3020 PRINT "
"
3030 PRINT "
"
3040 PRINT "
"
3050 PRINT "
"
3060 PRINT "
"
3070 POKE B,50
3080 POKE C,13
3090 IF INKEY$="" THEN GOTO 3090
3100 RETURN
```

```
5000 SAVE "GREEDY GRANNIE"
5010 RUN
```

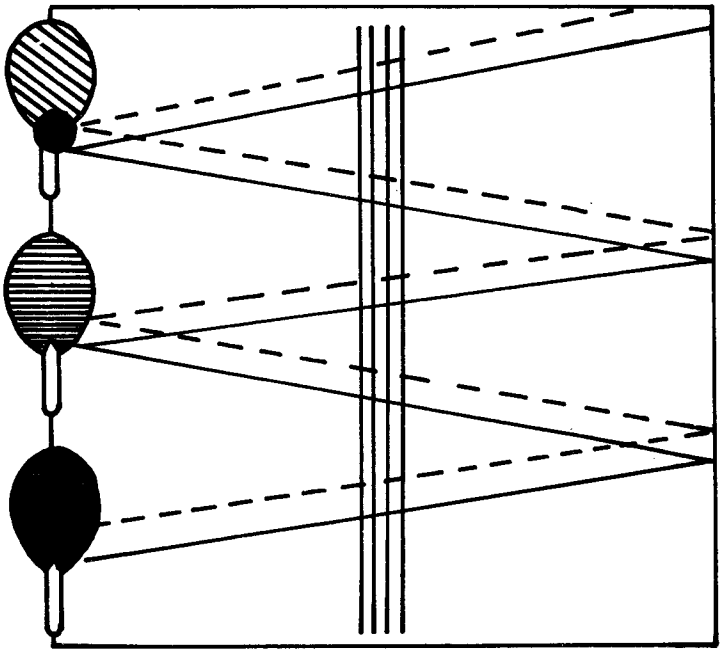




GAME, SET AND MATCH

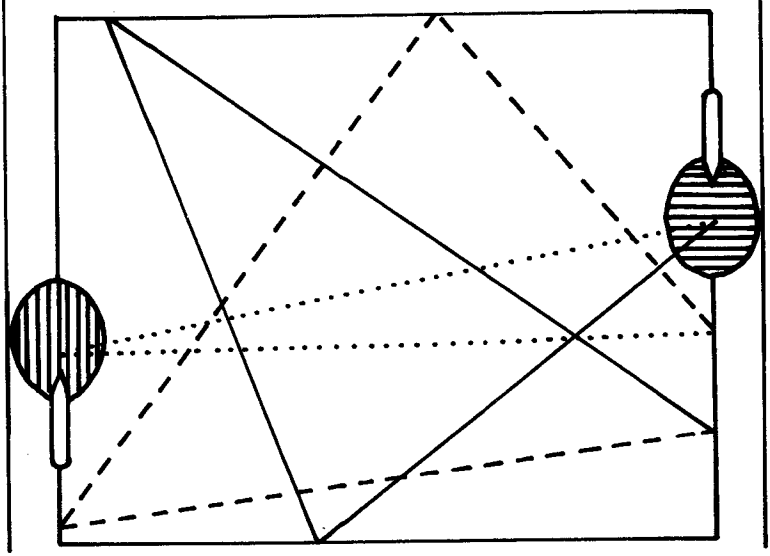
In this two-player game you each control a bat which can be moved up and down. You must try to hit the ball back to your opponent. If you miss the ball, your opponent scores a point — there is no limit to the score you can attain.

When the ball is hit by the bat it is deflected along any of three paths. The left player uses any key from 1 to 5 to move the bat up and any key from Q to T to move down. The right player uses any key from 6 to 0 to move up and any key between Y and P for down.



```
   1 REM 000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
000000000
   5 IF PEEK 16638=201 THEN GOTO
 50
  10 LET A$="205 187 002 203 085
 192 042 130 064 054 000 006 033
 035 016 253 126 254 022 032 005
 006 033 043 016 253 054 133 034
 130 064 201 "
  15 LET A$=A$+"062 085 050 138
064 062 130 050 141 064 050 163
064 062 035 050 147 064 062 043
050 157 064 062 133 050 161 064
205 134 064 062 "
  20 LET A$=A$+"093 050 138 064
062 043 050 147 064 062 035 050
157 064 205 134 064 062 101 050
138 064 062 132 050 141 064 050
163 064 062 005 "
  25 LET A$=A$+"050 161 064 205
134 064 062 109 050 138 064 062
035 050 147 064 062 043 050 157
064 205 134 064 201 "
  30 FOR N=16518 TO 16638
  35 POKE N,VAL A$( TO 3)
  40 LET A$=A$(5 TO )
  45 NEXT N
  50 LET DF=PEEK 16396+256*PEEK
16397
  52 LET LFT=0
  54 LET RGT=0
  58 RAND
  60 LET A=DF+332
  70 LET B=DF+361
  80 LET C=INT (A/256)
  85 CLS
  90 POKE 16514,A-C*256
 100 POKE 16515,C
 110 LET C=INT (B/256)
 120 POKE 16516,B-C*256
 130 POKE 16517,C
 140 LET A$="--------------------
--------------"
 150 PRINT AT 0,0;A$;AT 21,0;A$
 152 POKE 16418,0
 160 POKE A,133
 170 POKE B,5
 175 PRINT AT 23,0;"LEFT SCORE =
";LFT;"  RIGHT SCORE = ";RGT
 180 LET X=DF+345
 190 LET Z=RND
 200 LET D=(Z(.5)-(Z>=.5)
 205 LET E=INT (RND*3)-1
```

```
 210 LET Y=X
 215 LET X=X+D+33*E
 220 IF PEEK X=118 THEN GOTO 100
0
 230 IF PEEK X=0 AND PEEK Y=23 T
HEN GOTO 320
 240 IF PEEK X<>22 THEN GOTO 280
 250 LET E=-E
 260 LET X=Y
 270 GOTO 215
 280 LET D=-D
 290 LET E=INT (RND*3)-1
 300 LET X=Y
 304 IF PEEK Y<>23 THEN LET X=X+
D
 305 POKE X,23
 310 GOTO 210
 320 POKE Y,0
 330 POKE X,23
 340 LET Z=USR 16550
 350 GOTO 210
1000 IF D=-1 THEN LET RGT=RGT+1
1010 IF D=1 THEN LET LFT=LFT+1
1020 POKE Y,0
1030 FOR Z=1 TO 30
1040 NEXT Z
1050 GOTO 175
```



RULER OF LOS SINCLAROS

No, not the 12-inch kind, the kind that rules a country. You are the dictator of a Banana Republic in Central America, and your peasants have demanded that you rule them for the next 12 years. The incentive to win is enormous — a large pension, payable on abdication. However, managing the country is not as easy as it sounds. Only your grovelling friend, the ZX81, gives you any information whatsoever. You must try to increase your lands without starving the peasants — if you starve them, they will rise up in revolution. Each peasant requires two sacks of corn for nourishment a year, and an extra acre of land costs ten sacks of corn.

Only enter the number of extra acres of land, *not* the amount of corn that the extra land will cost. I will leave the rest of the strategy up to you, there is still a lot to find out. Happy Ruling!

N.B. When the program pauses, press a key to continue the game.

```
   5 REM ******WHEN PROGRAM PAUS
ES, PRESS ANY KEY******
  10 LET C=1000
  20 LET A=100
  30 LET R=500
  35 LET N=1
  50 LET L$="
```

```
"
  52 CLS
  55 PRINT L$
  60 PAUSE 4E4
  70 CLS
```

```
 100 PRINT "IN YEAR ";N;" OF OUR
 GLORIOUS RULERS REIGN"
 120 PRINT ,,,,"YOU HAVE OH MAST
ER "
 130 PRINT ,,"    A POPULATION OF
 ";R;" LOYAL          SUBJECTS"
 140 PRINT ,,"    A CORN STORE OF
 ";C;" SACKS"
 150 PRINT ,,"    AND ";A;" ACRES
 OF LAND"
 190 PAUSE 300
 200 CLS
 210 PRINT "HOW MUCH EXTRA LAND,
 MASTER?"
 220 INPUT L
 225 LET A=A+L
 230 LET C=C-L*10
 240 IF A<0 THEN GOTO 1000
 250 PRINT ,,,,"HOW MANY SACKS A
S FOOD MASTER?"
 260 INPUT F
 262 IF L=0 AND F=R THEN PRINT ,
,"DO SOMETHING CONSTRUCTIVE"
 264 IF L=0 AND F=R THEN GOTO 19
0
 265 IF F>R+100 THEN PRINT ,,,,"
YOUR PEASANTS WILL BE VERY FAT"
 267 IF F>R+100 THEN PAUSE 80
 270 LET C=C-2*F
 280 IF A<0 THEN GOTO 1000
 290 IF R-F>A/4 THEN GOTO 1500
 300 LET R=R+(F-R)*RND
 310 IF R<50 THEN GOTO 1000
 315 LET R=INT (R)
 320 LET C=C+INT (R*A/50)
 330 LET N=N+1
 340 IF N=12 THEN GOTO 500
 345 IF RND>.7 THEN GOSUB 705
 350 GOTO 52
 500 CLS
 520 FOR G=1 TO 91
 530 PRINT "******YOU DID IT OH
MASTER******"
 540 IF G/18=INT (G/18) THEN CLS

 550 NEXT G
 555 PRINT L$
 560 PRINT ,,,,"  [redacted]. THE
 LOYAL SUBJECTS OF LAS SINCLAROS
 ARE VERY HAPPY"
 570 PRINT ,,"THEY WILL GIVE YOU
 A PEACEFUL   RETIREMENT LEAVING
 YOU ALONE"
 580 PRINT "WITH YOUR ZX81. I HO
PE YOU ARE  HAPPY."
 590 PAUSE 4E4
```

```
 600 CLS
 610 PRINT ,,"OH, BEFORE YOU GO,
 I THOUGHT    THAT YOU WOULD LIK
ETO KNOW THAT"
 620 PRINT "YOU AND YOR PEASANTS
 HARVESTED A TOTAL OF ";C;" SACK
S OF CORN"
 630 PRINT AT 19,4;"PRESS ANY KE
Y TO START"
 635 IF INKEY$<>"" THEN RUN
 640 PRINT AT 17,8;"ANOTHER GAME
"
 650 GOSUB 800
 660 PRINT AT 17,8;"ANOTHER GAME
"
 670 GOSUB 800
 680 PRINT AT 17,8;"ANOTHER GAME
"
 690 GOSUB 800
 700 GOTO 635
 705 PRINT ,,,,"
"
 710 LET C=C-INT (RND*(C/4))
 720 LET C=C+A
 730 PRINT ,,"PRESS KEY TO RETUR
N"
 740 IF INKEY$="" THEN GOTO 740
 750 RETURN
 800 FOR T=1 TO 8
 810 NEXT T
 820 RETURN
1000 CLS
1010 PRINT "                YOU FAILED,
YOU GREEDY                  BARBARIAN"
1020 PRINT "                AND WERE OVE
RTHROWN IN A                VIOLENT REVO
LUTION."
1030 PRINT "                YOU WILL BE
EXECUTED AT                 SUNRISE TOMO
RROW."
1050 FOR T=1 TO 100
1060 NEXT T
1070 PRINT ,,,,,,"  THE PEOPLE O
F LAS SINCLAROS  HAVE DECIDED T
HAT LEAVING THEIR"
1080 PRINT "COUNTRYS ECONOMY TO
THE HANDS OFIGNORANT FOOLS IS A
BAD THING"
1090 PRINT ,,"    THEY HAVE THERE
FORE DECIDED  TO LET A COMPUTER
LOOK AFTER"
1100 PRINT "THEIR AFFAIRS....."
1110 PAUSE 4E4
1120 PRINT ,,"
"
1130 STOP
```

```
1500 CLS
1505 FOR T=1 TO 20
1510 PRINT AT 10,11;"REVOLUTION"
1520 PRINT AT 10,11;"■■■■■■■■■■"
1530 NEXT T
1540 PRINT ''''"PRESS A KEY AND
HOPE TO SURVIVE"
1542 IF INKEY$="" THEN GOTO 1542
1545 PRINT ''
1550 FOR T=1 TO 16
1560 PRINT ".";
1570 FOR V=1 TO 20
1575 NEXT V
1580 NEXT T
1600 IF RND(.25 THEN GOTO 1000
1605 LET D=INT (RND*(3*N))+10
1615 LET E=INT (RND*(8*N))+40
1620 CLS
1630 PRINT ''''"■■■■■■■REVOLUTIO
N STOPPED■■■■■■■"
1640 PRINT ''"BUT LET THAT BE A
LESSON TO YOU"
1650 PRINT ''"YOU ALSO LOST ";D;
" ACRES IN"
1660 PRINT "THE UPRISING AS WELL
 AS ";E;
1670 PRINT " SACKS OF CORN BURNT
 IN A FIRE"
1680 LET R=R-D
1690 LET A=A-E
1700 PAUSE 250
1710 GOTO 300
2000 SAVE "LOS SINCLARO■"
2010 RUN
```

How To Write Better Programs

By Tim Hartnell, series editor

There are a number of fine programs in this book, and many of the regular computer magazines contain other such ones. But no matter how good the programs from published sources are, you are certain to get more pleasure from running them if they have been partially or completely written by you. Putting your personal stamp on programs, altering them to reflect your wishes and creativity, is an excellent way to improve the programs, and eventually, of course, you'll become a better and more imaginative programmer.

Programs in magazines, and in books like this one, are ideal as starting points for your own developments. You may also find that advertisements for software packages can be fruitful 'idea-starters'. You only need to read the description of what the commercially available program does, and you will have the first step towards creating your own program. You have to be careful, of course, not to infringe copyright either in the screen displays, in the name of the program, or the names of the 'characters' within the program. However, you will probably find that at a certain point in its development the program will take on a life of its own, growing and evolving away from the original scenario, until you eventually have a completely new game concept and implementation.

Whatever you do, be careful not to pass off other people's work as your own. By all means adapt and im-

prove published programs, but do not then present them to magazines as if they were originals. I have lost count of the number of times one of my own programs, from one of my books, has been submitted to me for publication.

Always watch out for new ideas as you look through books, game and computer magazines, or wander through video game arcades. It may be worth keeping notes of ideas you come across for games, for character shapes, for sounds, for dramatic endings and so on. Thus you will never be short of ideas, and you will also be able to merge the material together to produce better games which hold the player's attention for longer.

Games tend to fall into one of three categories, and it is worth making sure of the category into which your proposed program will fall *before* you start to program, since the category of game materially alters the programming approach. This is not to say that, as you develop a program, it will not move from one category into another, nor that a particular game might not extend across two categories, but it is nevertheless useful to keep the various groups separate in your mind, just to clarify your thoughts. The three categories are:

1. Board games
2. 'Arcade' (that is, highly visual, fast moving, noisy, real time) games
3. Games of chance (such as Roulette and Snap).

In board games, the quality of play is more important than lightning-fast response, while the arcade-type programs must be kept moving at all costs, even if some 'intelligence' from your Martian intruders must be sacrificed to achieve this. Games of chance depend more on their ease of play ('user-friendly' inputs), and an approach to true randomness, than do either of the other categories.

You will find that games programs tend to fall into types, which are subdivisions of the three above mentioned categories. Many board games are variants of chess or checkers; many arcade games started off life as Space Invader-type games; and games of chance

started off in the 'real world' of dice and cards. Looking at a program description, or a games machine, and trying to categorise the game you see can help trigger new ideas which fit within that particular game's genre.

There is a school of thought within programming — generally called 'structured programming' — which believes that discipline at the beginning of the games-writing process is essential. While less interesting than sitting down at the computer right away, a much better program is produced in the end. I once wrote a program called Dome Dweller, a simulation program in which the player is in charge of a 'lunar dome' and must decide which products to manufacture and sell in order to buy oxygen and food for the station's inhabitants. (This program was used in my book *The Book of Listings*, written with Jeremy Ruston, and published by the BBC.)Once I had decided the overall scenario, I worked out the screen display, and came up with an idea as follows:

> Oxygen supplies are low
> There are 96 people living within your dome in year 3
> Money credit is $5,693
> Annual maintenance charge is $226
> Oxygen tanks hold 811 units
> Oxygen costs $8 per unit
> Each dome dweller needs 5 units a year
> Food stocks stand at 2122
> Each dweller needs 3 units a year ($6 each, $576 for dome. This will last 7 years at present population.)
> You can trade your unique lunar sculptures with the people who live in other domes. You use up 2 units of oxygen making each one, and sell them for $30.

As you can probably guess from this 'sample printout', the idea of the program is to decide how many 'unique lunar sculptures' you must make and sell in order to buy oxygen and food, and to pay the 'annual maintenance' charge. The problem with this particular program is that

making each sculpture uses up oxygen, so you must balance your wish to make money against the need to use the oxygen intelligently.

You may well wish to try writing such a program yourself. You should end up with an enjoyable program, and writing it will do much to help you develop your programming skills. The first thing to do is to make a list of what the program has to do:

Set up the needed variables
Tell the player the 'state of the dome'
Ask how much oxygen to be bought
Check if can afford this, if so buy it, if not go back and ask again
Ask how much food to be bought
Check if can afford this, if so buy it, if not go back and ask again
Update oxygen quantity
Update food quantity
Reduce money left total
Ask how many items of sculpture to be made
Check if there is enough oxygen to make this many, if not go back and ask again
Reduce oxygen quantity by amount needed to make the number of sculptures specified, increase money total to reflect value of sculptures made
Increase the population total slightly, add one to the 'current year'
Check if there is enough food in stocks to feed whole population
Check if there is enough oxygen for whole population
Check if there is any money
If any of these conditions are negative (eg not enough food) send action to an 'end of game' routine
If all are positive, loop back to tell the player the state of the dome, and continue to circle

You could probably write a Dome Dweller program

using the list above, together with the 'sample printout' information. There is, however, a secret I should like to share with you which unlocks programming problems almost instantly. You can actually write all the vital parts of a program in minutes, so you can see the raw framework of a program like this running long before you fill in the details. And once you have a framework you can work on it for as long as you like, knowing as you do so that — at every moment in program development — you have a working program. You do not have to wait until the end until you can run it to see how you are going. The 'secret' is to hold the entire program within a series of subroutine calls, all held within a perpetual loop. Here's how it could work with this program. The very first lines you enter in your computer are as follows:

```
10 REM DOME DWELLER
20 GOSUB 1000: REM ASSIGN VARIABLES
30 GOSUB 2000: REM PRINT OUT STATE OF
DOME
40 GOSUB 3000: REM OXYGEN
50 GOSUB 4000: REM FOOD
60 GOSUB 5000: REM SCULPTURE
70 GOSUB 6000: REM UPDATE POPULATION
80 GOSUB 7000: REM CHECK ON STATE OF
 DOME
90 IF (all conditions positive, from GOSUB 7000)
THEN GOTO 30
100 REM End of game ...
```

As you can see once you have the 'master loop' set up in this way, it is relatively simple to fill in each of the subroutines one by one, testing each as you do so, and elaborating each one so that you end up eventually with a very good program. The only thing you need now is a list of the variables which you will use with the program.

I find the best way to do this is to use explicit names for variables so that when you are programming you do not have to spend time checking, for example, whether AA stands for the population, or the number of units of oxygen used up in making each item of sculpture. To make

programs as easy as possible to transfer between different computers you can stick to two letter variable names, or you can take advantage (if your computer allows it) of long names (such as OXYUSE for the amount of oxygen used) for variables. Then you have no doubts whatsoever as to the meaning of each variable name. To show how this can work, and to illustrate a further advantage of explicit variable names, here are the variables used in Dome Dweller:

FOLK — population of dome
CASH — money in treasury
FOOD — food stocks on hand
FOODCOST — how much each unit of food costs
FOODNEED — how many units of food were consumed per person per year
ARTCOST — how much oxygen was used up making each piece of sculpture
ARTPAY — how many dollars each piece of sculpture was sold for
OXY — oxygen stocks on hand
OXYNEED — how many units of oxygen were consumed per person per year
OXYCOST — how much each unit of oxygen cost to buy
REPAIR — the cost of annual repairs to the dome
YEAR — the year of the dome's life

Using explicit variable names in this way — although they use up more memory than do single or double-letter variable names — makes it very simple to follow through a program, working out what each section of the program actually does. Moreover, and this is the further advantage mentioned, it is very easy when writing the program to insert the formulae required for calculations. By this I mean that if, for example, you wished to include (as I do in this program) an indication of how much oxygen is needed for each year, you simply multiply the number of people in the dome (FOLK) by the number of oxygen units each person needs each year (OXYNEED). You can then

include within the printouts for the state of the dome a line like:

```
PRINT "THERE ARE ";FOLK;" IN THE DOME"
PRINT "IN YEAR ";YEAR
PRINT "EACH PERSON NEEDS ";OXYNEED;"
      UNITS OF"
PRINT "OXYGEN EACH YEAR,";
      OXYNEED*FOLK;" NEEDED"
PRINT "FOR THE WHOLE DOME"
```

It also makes it very easy to check on whether purchases are possible. For example, to buy food, you could say:

```
PRINT "HOW MUCH FOOD WILL YOU BUY?"
INPUT A
IF A*FOODCOST > CASH THEN GOTO (get
another A)
```

So the suggestions given here for improving your programs by the use of 'structured programming' include the following:

- draw up a sample printout, or mock-up of the final screen display
- draw up a list of what the program has to do each time through a 'master control loop'
- change this list to a series of subroutine calls
- use explicit variable names if possible

It is useful if you are designing programs for others to use to ensure that it is quite clear what the player should do when running the program. There is little point, especially when memory is limited, in including a long set of instructions within the program, but you should certainly write such instructions down. In addition, user prompts should be explicit (such as ENTER THE NUMBER OF GOES YOU WANT) and should include warnings of the limits which will be placed on the input (HOW MANY CARDS WILL YOU START WITH: 1, 2 OR 3 ?, for instance).

You cannot assume that you will be present every time a program is run, so you should do your best to make it as foolproof as possible. If you can, add error-trapping routines to the program to ensure that a mistake in enter-

ing a choice earlier on in the program will not cause it to crash or come up with stupid results later on.

If you read through this section of the book several times and try to apply the ideas to your own programming work, you should find your work quality improves significantly, and also that you can spend more time improving and embellishing a program and less in the raw mechanical task of getting the thing running.

GLOSSARY

A

Accumulator — the place within the computer in which arithmetic computations are performed and where the results of these computations are stored.

Algorithm — the series of steps the computer follows to solve a particular problem.

Alphanumeric — this term is usually used in relation to a keyboard, as in 'it is an alphanumeric keyboard', which means that the keyboard has letters as well as numbers. It is also used to refer to the 'character set' of the computer. The character set comprises the numbers and letters the computer can print on the screen.

ALU (Arithmetic/Logic Unit) — the part of the computer which does arithmetic (such as addition, subtraction) and where decisions are made.

AND — a Boolean logic operation that the computer uses in its decision-making process. It is based on Boolean algebra, a system developed by mathematician George Boole (1815-64). In Boolean algebra the variables of an expression represent a logical operation such as OR and NOR.

ASCII — stands for American Standard Code for Information Exchange, the most widely used encoding system for English language alphanumerics. There are 128 upper and lower case letters, digits and some special characters. ASCII converts the symbols and control instructions into seven-bit binary combinations.

Assembler — a program which converts other programs written in assembly language into machine code

(which the computer can understand directly). Assembly language is a low level programming language which uses easily memorised combinations of two or three letters to represent a particular instruction which the assembler then converts so the machine can understand it. Examples of these are ADD (add), and SUB (subtract). A computer programmed in assembly language tends to work more quickly than one programmed in a higher level language such as BASIC.

B

BASIC — an acronym for Beginners All-Purpose Symbolic Instruction Code. It is the most widely used computer language in the microcomputer field. Although it has been criticised by many people, it has the virtue of being very easy to learn. A great number of BASIC statements resemble ordinary English.

Baud — named after Baudot, a pioneer of telegraphic communications. Baud measures the rate of transfer of information and is approximately equal to one bit per second.

BCD — an abbreviation for Binary Coded Decimal.

Benchmark — a test against which certain functions of the computer can be measured. There are a number of so-called 'standard Benchmark tests', but generally these only test speed. This is rarely the aspect of a microcomputer that is most of interest to the potential buyer.

Binary — a numbering system that uses only zeros and ones.

Bit — an abbreviation for Binary Digit. This is the smallest unit of information a computer circuit can recognise.

Boolean Algebra — the system of algebra developed by mathematician George Boole which uses algebraic notation to express logical relationships (see AND).

Bootstrap — a short program or routine which is read into the computer when it is first turned on. It orients the computer to accept the longer, following program.

Bug — an error in a computer program which stops the program from running properly. Although it is generally used to mean only a fault or an error in a program, the term bug can also be used for a fault in the computer hardware.

Bus — a number of conductors used for transmitting signals such as data instructions, or power in and out of a computer.

Byte — a group of binary digits which make up a computer word. Eight is the most usual number of bits in a byte.

C

CAI — Computer Assisted Instruction.

CAL — Computer Assisted Learning. The term is generally used to describe programs which involve the learner with the learning process.

Chip — the general term for the entire circuit which is etched onto a small piece of silicon. The chip is, of course, at the heart of the microcomputer.

Clock — the timing device within the computer that synchronises its operations.

COBOL — a high level language derived from the words Common Business Orientated Language. COBOL is designed primarily for filing and record-keeping.

Comparator — a device which compares two things and produces a signal related to the difference between the two.

Compiler — a computer program that converts high level programming language into binary machine code so the computer can handle it.

Complement — a number which is derived from another according to specified rules.

Computer — a device with three main abilities or functions:
1) to accept data
2) to solve problems
3) to supply results

CPU — stands for Central Processing Unit. This is the heart of the computer's intelligence, where data is handled and instructions are carried out.

Cursor — a character which appears on the TV screen when the computer is operating. It shows where the next character will be printed. On a computer there are usually 'cursor control keys' to allow the user to move the cursor around the screen.

D

Data — information in a form which the computer can process.

Debug — the general term for going through a program and correcting any errors in it, that is, chasing down and removing bugs (see Bug).

Digital Computer —a computer which operates on information which is in a discrete form.

Disk/Disc — this is a magnetically sensitised plastic disk, a little smaller than a single play record. This is used for storing programs and for obtaining data. Disks are considerably faster to load than a cassette of the same length program. The disk can be searched very quickly while a program is running for additional data.

Display — the visual output of the computer, generally on a TV or monitor screen.

Dot Matrix Printer — a printer which prints either the listing of a program or that which is displayed on the TV screen. Each letter and character is made up of a number of dots. The higher the number of dots per character the finer the resolution of the printer.

Dynamic Memory — a memory unit within the computer which 'forgets' its contents when the power is turned off.

E

Editor — this term is generally used for the routine within the computer which allows you to change lines of a program while you are writing it.

EPROM — stands for Erasable Programmable Read-Only Memory. This is like the ROM in the computer, except that it is fairly easy to load material into an EPROM and it doesn't disappear when you turn the power off. EPROMs must be placed in a strong ultra violet light to erase them.

Error Messages — the information given by a computer where there is a fault in the coding during a part of a program, usually shown by the computer stopping, and printing a word, or a word and numbers, or a combination of numbers only, at the bottom of the screen. This tells you what mistake has been made. Common mistakes include using the letter O instead of zero in a line, or leaving out a pair of brackets, or one of the brackets, in an expression, or failing to define a variable.

F

File — a collection of related items of information organised in a systematic way.

Floppy Disk — a relatively cheap form of magnetic disk used for storing computer information, and so named because it is quite flexible (see Disk/Disc).

Flow Chart — a diagram drawn up before writing a program, in which the main operations are enclosed within rectangles or other shapes and connected by

lines, with arrows to represent loops, and decisions written at the branches. It makes writing a program much easier because traps such as infinite loops, or non-defined variables can be caught at an early stage. It may not be worth writing a flow chart for very short programs, but generally a flow chart aids in creating programs.

Firmware — there are three kinds of 'ware' in computers: software 'temporary' programs; hardware like the ROM which contains permanent information; and firmware in which the information is relatively permanent, as in an EPROM (see EPROM).

Flip-Flop — a circuit which maintains one electrical condition until changed to the opposite condition by an input signal.

FORTRAN — an acronym for FORmula TRANslation, this is a high level, problem orientated computer language for scientific and mathematical use.

G

Gate — an electrical circuit which, although it may accept one or more incoming signals, only sends out a single signal.

Graphics — pictorial information as opposed to letters and numbers.

H

Hard Copy — computer output which is in permanent form.

Hardware — the physical parts of the computer (also see software and firmware).

Hexadecimal (Hex) — a numbering system to the base sixteen. The digits zero to nine are used, as well as the letters A, B, C, D, E and F to represent numbers. A

equals 10, B equals 11, C equals 12, and so on. Hex is often used by microprocessor users.

Hex Pad — a keyboard designed specifically for entering hexadecimal notation.

High Level Language — a programming language which allows the user to talk to the computer more or less in English. In general, the higher the level of the language (that is, the closer it is to English), the longer it takes for the computer to translate it into a language it can use. Lower level languages are far more difficult for human operators but are generally executed far more quickly.

I

Input — the information fed into the computer via a keyboard, a microphone, a cassette or a disk.

Input/Output (I/O Device) — a device which accepts information or instructions from the outside world, relays it to the computer, and then, after processing, sends the information out in a form suitable for storing, or in a form which could be understood by a human being.

Instruction — data which directs a single step in the processing of information by the computer (also known as a command).

Integrated Circuit — a complete electronic circuit imprinted on a semiconductor surface.

Interface — the boundary between the computer and a peripheral such as a printer.

Interpreter — a program which translates the high level language fed in by the human operator, into a language which the machine can understand.

Inverter — a logic gate that changes the signal being fed in, to the opposite one.

Interactive Routine — part of a program which is repeated over and over again until a specified condition is reached.

J

Jump Instruction — an instruction which tells the computer to go to another part of the program, when the destination of this move depends on the result of a calculation just performed.

K

K — this relates to the size of the memory. Memory is usually measured in 4K blocks. 1K contains 1,024 bytes.

Keyword — the trigger word in a line of programming, usually the first word after the line number. Keywords include STOP, PRINT and GOTO.

L

Language — computer languages are divided into three sections: high level languages, such as BASIC, which are reasonably close to English and fairly easy for humans to use; low level languages, such as Assembler, that use short phrases which have some connection with English (ADD for add and RET for return, for instance); and machine code which communicates more or less directly with the machine.

LCD — this stands for Liquid Crystal Diode. Some computers such as the TRS-80 Pocket Computer use an LCD display.

LED — this stands for Light Emitting Diode. The bright

red numbers which are often used on watch or clock displays are made up of LEDs.

Logic — the mathematical form of a study of relationships between events.

Loop — a sequence of instructions within a program which is performed over and over again until a particular condition is satisfied.

M

Machine Language or Machine Code — an operation code which can be understood and acted upon directly by the computer.

Magnetic Disk — see Disk and Floppy Disk.

Mainframe — computers are generally divided into three groups, and the group a computer falls into depends more or less on its size. The computer you are thinking of buying is a microcomputer; medium sized computers are known as minicomputers; and the giant computers that you sometimes see in science fiction movies are mainframe computers. Until 15 years ago mainframe computers were, in practical terms, the only ones available.

Memory — there are two types of memory within a computer. The first is called ROM (read-only memory); this is the memory that comes already programmed on the computer, which tells the computer how to make decisions and how to carry out arithmetic operations. This memory is unaffected when you turn the computer off. The second type is RAM (random access memory). This memory holds the program you type in at the keyboard or send in via a cassette or disk. In most computers the computer 'forgets' what is in RAM when you turn the power off.

Microprocessor — the heart of any computer. It requires peripheral unit interfaces, such as a power supply and input and output devices, to act as a microcomputer.

MODEM — stands for Modulator Demodulator. This is a device which allows two computers to talk to each other over the telephone. The computers usually use a cradle in which a telephone receiver is placed.

Monitor — this has two meanings in computer terms. One meaning is a television-like display. A monitor has no facility for tuning television programs, and usually the picture produced on a monitor is superior to that produced by an ordinary television. The second meaning of a monitor relates to ROM. The monitor of a computer is described as the information it has built in when you buy it. This information allows it to make decisions and carry out arithmetic computations.

Motherboard — a framework to which extra circuits can be added. These extra circuits often give the computer facilities which are not built-in, such as that of producing sound or of controlling a light pen.

MPU — an abbreviation for Microprocessor Unit.

N

Nano-second — a nano-second is one thousand billionth of a second, the unit of speed in which a computer or a memory chip is often rated.

Non-Volatile Memory — memory which is not lost when the computer is turned off. Some of the smaller computers such as the TRS-80 Pocket Computer have non-volatile memory. The batteries hold the program you enter for several hundred hours.

Not — a Boolean logic operation that changes a binary digit into its opposite.

Null String — a string which contains no characters. It is shown in the program as two double quote marks, without anything between them.

Numeric — pertaining to numbers as opposed to letters (that is, alphabetic). Many keyboards are described

as being alphanumeric which means both numbers and letters are provided.

O

Octal — a numbering system which uses eight as the base, and the digits 0, 1, 2, 3, 4, 5, 6 and 7. The Octal system is not used very much nowadays in microcomputer fields. The Hexadecimal system is more common (see Hexadecimal).

Operating System — the software or firmware generally provided with the machine that allows you to run other programs.

OR — an arithmetic operation that returns a 1, if one or more inputs are 1.

Oracle — a method of sending text messages with a broadcast television signal. A teletext set is required to decode the messages. Oracle is run by Independent Television Service in the UK, and a similar service — Ceefax — is provided by the BBC.

Output — information or data fed out by the computer to such devices as a TV-like screen, a printer or a cassette tape. The output usually consists of the information which the computer has produced as a result of running a program.

Overflow — a number too large or too small for the computer to handle.

P

Pad — see Keypad.

Page — often used to refer to the amount of information needed to fill one TV screen, so you can talk about seeing a page of a program, the amount of the listing that will appear on the screen at one time.

PASCAL — a high level language.

Peripheral — anything which is hooked onto a computer, for control by the computer, such as a disk unit, a printer or a voice synthesiser.

Port — a socket through which information can be fed out of or in to a computer.

Prestel — the British telecom name for a system of calling up pages of information from a central computer via the telephone and displaying them on a television screen. A similar commercial version in the United States is known as The Source.

Program — in computer terms program has two meanings. One is the list of instructions that you feed into a computer, and the second is used as a verb, as in 'to program a computer'.

PROM — stands for Programmable Read Only Memory. This is a device which can be programmed, and once it is then the program is permanent (also see EPROM and ROM).

R

Random Access Memory (RAM) — the memory within a computer which can be changed at will by the person using the computer. The contents of RAM are usually lost when a computer is turned off. RAM is the memory device that stores the program that you type in and also stores the results of calculations in progress.

Read-Only Memory (ROM) — in contrast to RAM, information in ROM cannot be changed by the user of the computer, and the information is not lost when the computer is turned off. The data in ROM is put there by the manufacturers and tells the computer how to make decisions and how to carry out arithmetic computations. The size of ROM and RAM is given in the unit K (see K).

Recursion — the continuous repetition of a part of the program.

Register — a specific place in the memory where one or more computer words are stored during operations.

Reserved Word — a word that you cannot use for a variable in a program because the computer will read it as something else. An example is the word TO. Because TO has a specific computer meaning, most computers will reject it as a name for a variable. The same goes for words like FOR, GOTO and STOP.

Routine — this word can be used as a synonym for program, or can refer to a specific section within a program (also see Subroutine).

S

Second Generation — this has two meanings. The first applies to computers using transistors, as opposed to first generation computers which used valves. Second generation can also mean the second copy of a particular program; subsequent generations are degraded by more and more noise.

Semiconductor — a material that is usually an electrical insulator but under specific conditions can become a conductor.

Serial — information which is stored or sent in a sequence, one bit at a time.

Signal — an electrical pulse which is a conveyor of data.

Silicon Valley — the popular name given to an area in California where many semiconductor manufacturers are located.

SNOBOL — a high level language.

Software — the program which is entered into the computer by a user which tells the computer what to do.

Software Compatible — this refers to two different computers which can accept programs written for the other.

Static Memory — a non-volatile memory device which retains information so long as the power is turned on, but does not require additional boosts of power to keep the memory in place.

Subroutine — part of a program which is often accessed many times during the execution of the main program. A subroutine ends with an instruction to go back to the line after the one which sent it to the subroutine.

T

Teletext — information transmitted in the top section of a broadcast television picture. It requires a special set to decode it to fill the screen with text information. The BBC service is known as Ceefax, the ITV service as Oracle. Teletext messages can also be transmitted by cable, for example the Prestel service in Britain or The Source in the United States.

Teletype — a device like a typewriter which can send information and also receive and print it.

Terminal — a unit independent of the central processing unit. It generally consists of a keyboard and a cathode ray display.

Time Sharing — a process by which a number of users may have access to a large computer which switches rapidly from one user to another in sequence, so each user is under the impression that he or she is the sole user of the computer at that time.

Truth Table — a mathematical table which lists all the possible results of a Boolean logic operation, showing the results you get from various combinations of inputs.

U

UHF — Ultra High Frequency (300-3000 megaHertz).

Ultra Violet Erasing — Ultra violet light must be used to erase EPROMs (see EPROM).

V

Variable — a letter or combination of letters and symbols which the computer can assign to a value or a word during the run of a program.

VDU — an abbreviation for Visual Display Unit.

Volatile — refers to memory which 'forgets' its contents when the power is turned off.

W

Word — a group of characters, or a series of binary digits, which represent a unit of information and occupy a single storage location. The computer processes a word as a single instruction.

Word-Processor — a highly intelligent typewriter which allows the typist to manipulate text, to move it around, to justify margins and to shift whole paragraphs if necessary on a screen before outputting the information onto a printer. Word-processors usually have memories, so that standard letters and the text of letters, written earlier, can be stored.

BIBLIOGRAPHY

Compiled by Tim Hartnell

The A to Z Book of Computer Games (McIntire, Thomas C, Tab Books, Blue Ridge Summit, Pa.).

This is a fine Tab book to give you program ideas and ready-to-run programs, although some of the games are a disappointment, such as the overly long Othello program which does not even play, but simply records the moves made by two human players. Others, however, such as Fivecard and Hotshot, are well written, and well worth entering into your microcomputer.

BASIC Computer Games (ed. Ahl, David, Creative Computing Press, Morristown, New Jersey).

This is a classic work, the source of more programming ideas than any other computer games book ever published. I had a meal with David Ahl one night in London after a PCW show and discussed the book. He said that he'd been in the personal computer field almost before there were personal computers, and while many of the games in this book do not seem startling now, the fact that people could write and play games for computer interaction at all seemed quite incredible in the late seventies. The Checkers program, and Life for Two are just a couple of the treasures you will find in this splendid program and idea source book.

BASIC Computer Programs for the Home (Sternberg, Charles D, Hayden Book Company, Inc., Rochelle Park, New Jersey).

Traditionally, home computers (when first purchased) have been used for playing games. One reason why they have not been used for more serious applications