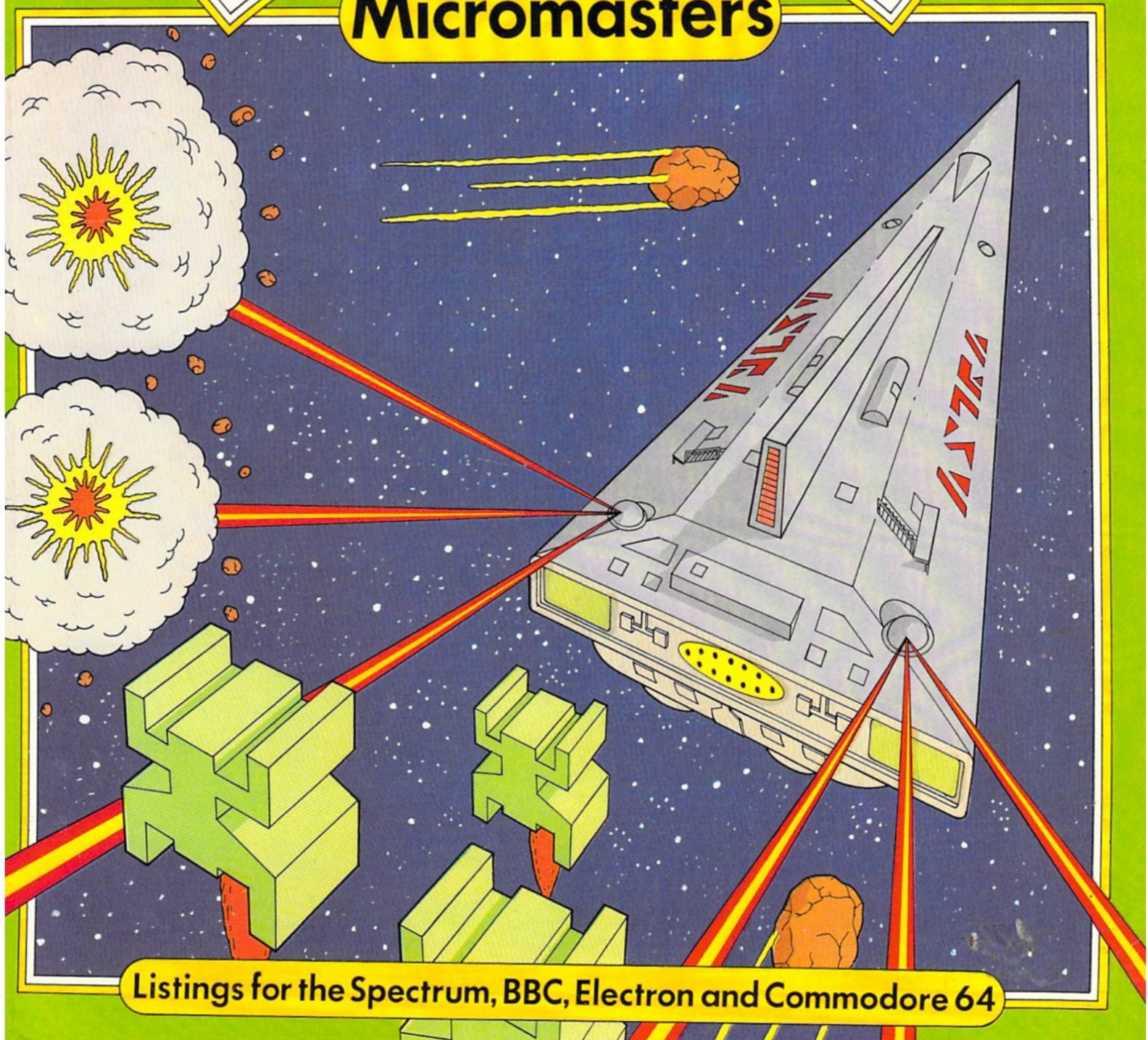


*St Michael*

# COMPUTER GAMES

## Micromasters



Listings for the Spectrum, BBC, Electron and Commodore 64

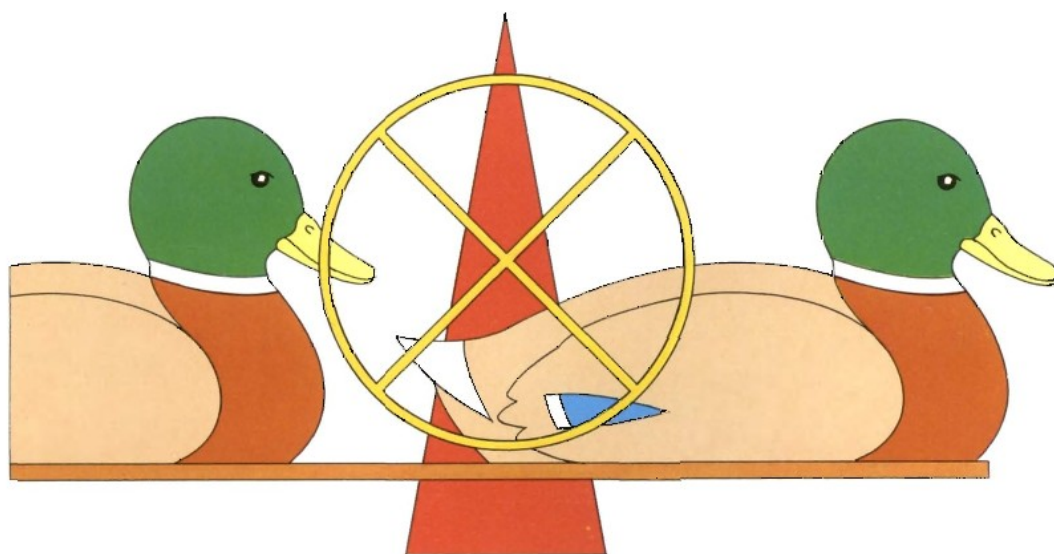




*St Michael*

# COMPUTER GAMES

**Micromasters**



---

---

**Author:** Patricia Grady

**Designer:** Ben White

**Illustrators:** Hayward Art

**Editorial:** Christopher Maynard  
Frances Ferneyhough

Published exclusively for Marks and Spencer p.l.c.  
by Grisewood & Dempsey Limited,  
Elsley Court, 20-22 Great Titchfield Street, London W1

© Grisewood & Dempsey Limited 1984

Printed by Graficas Reunidas SA, Madrid, Spain

---

---



# CONTENTS

Meteor Storm . . . . .	4-5	Blockbuster . . . . .	20-21
Alien Invaders . . . . .	6-7	Mu Torere . . . . .	22-23
Bananas . . . . .	8-9	Cops and Robbers . . . . .	24-25
Spider's Web . . . . .	10-11	Sorcerer's Steps . . . . .	26-27
Target Practice . . . . .	12-13	Parcel Drop . . . . .	28-29
Fortune-Teller . . . . .	14-15	Forgery . . . . .	30-31
Grand Slalom . . . . .	16-17	Subroutine Notes . . . . .	32
Laser Locksmith . . . . .	18-19		

## BEFORE YOU START PROGRAMMING

1. All the programs in this book were written, initially, for the Spectrum. However, they can also be made to work with a BBC, Electron or Commodore 64.

If using one of the latter, you will have to replace some of the lines in the programs with new ones:

2. If you are **NOT** using a Spectrum, leave out any line with the symbol **•** in front of it. When you have finished typing in the main listing, type in all the variation lines given for your machine. Your computer will insert them in the right order.

3. The program listings sometimes refer to subroutines (with the instruction GOSUB) numbered 1000, 2000, 3000 and so on. All these are found on page 32. Add all the subroutines that are called in the main listing onto the end of each program when you have typed it in.

4. All the listings should be typed into your computer in capital letters

(CAPS). When you run a program, also be sure to put your computer into CAPS mode. Otherwise, some of the programs will not work.

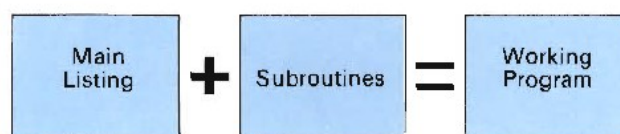
5. When a program comes to the end a 'stop message' usually appears on screen. After a program has halted, or if it crashes for some reason as you are running it, you can re-start by typing and entering RUN (unless otherwise instructed).

6. There are many ways to change

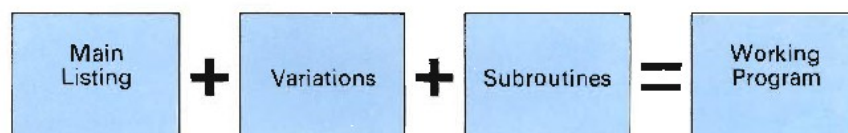
or improve the programs in this book. The obvious place is with new screen messages, or with other colours or by changing the speed of play. Experiment for yourself.

7. Spectrum users are reminded that the keywords FN and DEF FN are entered in E mode. To enter the instruction FNR(6), for example, first obtain FN, then type R(6) normally. To get DEF FNR(N), enter DEF FN and then type R(N).

### FOR SPECTRUM



### FOR OTHER COMPUTERS





## GAME NOTES

During a routine cruise through the Sirius system, the sirens of the Starship *Pluto* sound 'Red Alert'. The scanners have picked up an unusual storm of metallic meteors. The shipboard computers are not programmed to deal with anything like this. You will have to fire the photon blasters manually.

You have one blaster for each of the numbered sectors shown on the screen. Use the keys 1 to 8 to fire. The sooner you hit the meteors, the more points you score.

The ship has a force field which will vaporise any meteors you miss, but will reduce in strength each time. The numbers near the centre of the display show the strength of this field. If meteor hits reduce the strength to zero in any sector the ship is destroyed.

At the start of the game you choose a level of difficulty from 1 (easy) to 20 (impossible). To win you must survive 30 meteors.

## PROGRAM NOTES

Lines 100-130: Define two graphics characters and set up display colours.

Lines 140-160: Input the level of difficulty and set the starting value of the variables.

Lines 170-200: Divide the screen into sectors.

Lines 210-240: Print the 'force field'.

Lines 250-330: Complete the scanner display.

Lines 400-440: Select at random one of 16 starting positions for the next meteor.

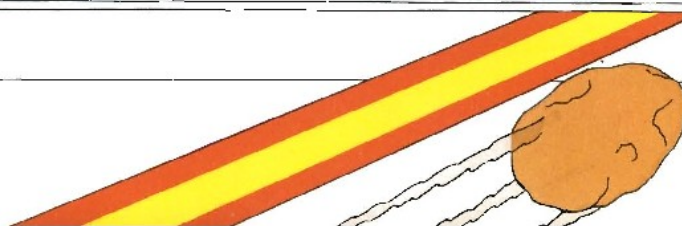
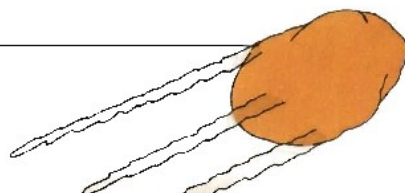
Lines 450-490: Move the meteor towards the ship. The motion stops

## SPECTRUM LISTING

```

100 REM METEORS
110 GOSUB 1000:LET BL=20:GOSUB 2000
120 DATA 255,255,255,255,255,255,255,255,
    62,63,255,255,127,127,126,56
•130 BORDER 0:PAPER 0:CLS:LET L=0:LET J=11
•140 LET K=16:LET W=1:LET T=0:LET X=0.01
150 INPUT" LEVEL (1-20) ";D:GOSUB 3000
160 DIM X(8):DIM Y(8):DIM P(8):DIM Q(8)
•170 INK 7:DRAW 255,175:PLOT 128,0
•180 DRAW 0,175:PLOT 255,0:DRAW -255,175
190 LET R=J-1:LET C=0:GOSUB 4000
200 FOR N=1 TO 2*K:PRINT"_";NEXT N
210 DIM S(8):FOR N=0 TO 6:LET A=PI*N/30
220 LET R=J+INT(6.4*COS(A))
230 LET C=K+INT(7.2*SIN(A))
240 GOSUB 4000:PRINT"+":NEXT N
250 FOR N=0 TO 3:FOR C=K+N-4 TO K-N+3
260 LET R=J-N-1:GOSUB 4000:PRINT U$
270 LET R=J+N:GOSUB 4000:PRINT U$:NEXT C
280 NEXT N:FOR N=1 TO 8:READ R,C,V,H
290 GOSUB 4000:PRINT CHR$(N+48):GOSUB
    5000:PRINT CHR$(51):LET S(N)=3
300 LET X(N)=R:LET Y(N)=C:LET P(N)=V:LET
    Q(N)=H:NEXT N:FOR N=1 TO 50*W:NEXT N
•310 DATA 0,23,9,16,7,31,10,17,14,31,11
•320 DATA 17,21,23,12,16,21,8,12,15,14,0
•330 DATA 11,14,7,0,10,14,0,8,9,15
•400 INK 6:LET L=L+1:LET N=11:LET M=FNR(8)
410 LET F=X(M):LET G=Y(M):LET P=2*FNR(2)
420 IF F<2 OR F>2*J-3 THEN LET G=G+3*P-9
430 IF G=0 OR G=2*K-1 THEN LET F=F+2*P-6
440 LET A=(F-J)/J:LET B=(G-K)/J
450 LET R=INT(F):LET C=INT(G):LET N=N-1
460 GOSUB 4000:PRINT Z$:GOSUB 7000
470 FOR Z=1 TO (20-D)*W:NEXT Z:IF K$=
    CHR$(M+48) THEN LET T=T+5*N:GOTO 500
480 GOSUB 4000:PRINT" ":IF A/(F-J)>1.8/J
    AND B/(G-K)>1.8/J THEN GOTO 530
490 LET F=F-A:LET G=G-B:GOTO 450
•500 FOR Z=1 TO 5:BEEP X,1:BEEP X,0:NEXT Z
510 GOSUB 4000:PRINT U$:FOR Z=1 TO 50*W
520 NEXT Z:GOSUB 4000:PRINT" ":GOTO 550
530 LET V=P(M):LET H=Q(M):GOSUB 5000:LET
    S(M)=S(M)-1:PRINT CHR$(S(M)+48)
540 IF S(M)=0 THEN LET M$="SHIP
    DESTROYED":GOTO 570
550 IF L<30 THEN GOTO 400

```





# METEOR STORM

1

```
560 LET M$="THE SHIP IS SAFE"
570 CLS:PRINT AT 10,8;FLASH 1;M$:PRINT
580 PRINT TAB(K-13);"YOU SCORED ";T;
    " AT LEVEL ";D:PRINT:PRINT:STOP
```

## ★ BBC/ELECTRON

```
130 MODE 1:VDU 23,1,0;0;0;0;0;:PRINT:PRINT
140 PRINT:L=0:J=16:K=20:W=25:T=0:X=-15
170 DRAW 1279,1023:MOVE 1279,0
180 DRAW 0,1023:MOVE 640,0:DRAW 640,1023
310 DATA 1,28,14,20,10,39,15,21,21,39,16
320 DATA 21,30,28,17,20,30,11,17,19,21,0
330 DATA 16,18,10,0,15,18,1,11,14,19
340 ON ERROR GOTO 480
400 COLOUR 2:L=L+1:N=11:M=FNR(8)
405 *FX 15,1
500 SOUND 1,X,9,9: SOUND 0,X,6,7: SOUND 0,X,4,5
570 CLS:PRINT TAB(12,8);M$:PRINT
NB Electron users should reduce the value of W in line 140 from 25 to 10.
```

## ★ COMMODORE 64

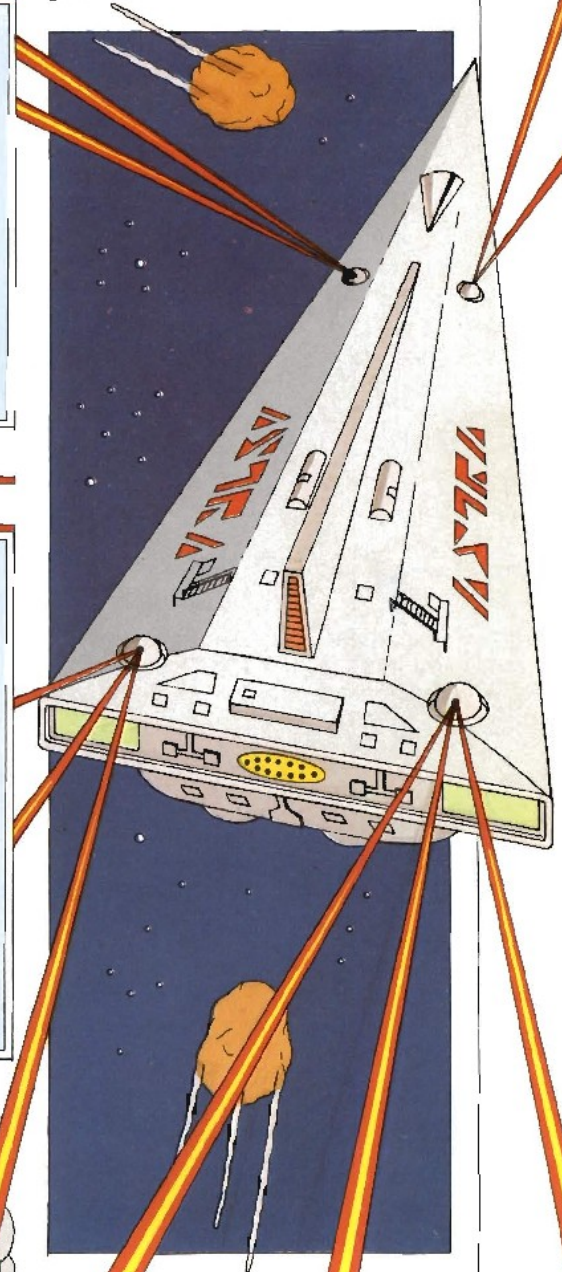
```
130 POKE 53280,0:POKE 53281,0:POKE 646,14
140 L=0:J=12:K=20:W=6:T=0:PI=3.1416:PRINT C$;
165 FOR N=1 TO 8:PRINT TAB(2*N);CHR$(109);
    TAB(20);CHR$(165);TAB(39-2*N);
170 PRINT CHR$(110):NEXT N:PRINT LEFT$(R$,5)
175 FOR N=16 TO 2 STEP -2:PRINT TAB(N);
    CHR$(110);TAB(20);CHR$(165);
180 PRINT TAB(39-N);CHR$(109):NEXT N
310 DATA 0,28,10,20,7,39,11,21,16,39,12
320 DATA 21,23,28,13,20,23,11,13,19,16,0
330 DATA 12,18,7,0,11,18,0,11,10,19
400 POKE 646,7:L=L+1:N=11:M=FNR(8)
500 DN=30:LD=0:HI=8:GOSUB 8000:HI=4:GOSUB 8000
570 PRINT C$:V=8:H=J:GOSUB 5000:PRINT M$:PRINT
```

at line 470 if the correct key is pressed or at line 480 when the meteor reaches the force field.

Lines 500-520: Display a 'hit'.

Lines 530-540: Reduce the strength of the force field and jump to line 570 if the new value is zero.

Lines 550-580: Send the program back to line 400 for a new meteor or print the final score.



5



### GAME NOTES

London is in peril! An Interplanetary Battle Fleet from Mars – whose inhabitants are infuriated by our claims that their planet is dead – has attacked. The Ministry of Defence is asleep, so you have to defend the city with just one missile launcher.

You can move the launcher to the left using capital Z, and to the right using X. Press F when you are ready to fire.

The aliens descend cautiously at first, but be on your guard. If you let a single ship through, London will be devastated.

During the game you score points for each invader you hit – more points for a ship high in the sky than for one close to the ground.

If you want to make the game more difficult, try replacing  $(N + 2)$  in line 200 by  $(N + 1)$  or reducing the 4 in line 290 to 3, 2 or 1.

### PROGRAM NOTES

**Lines 100-150:** Define 'invader' and 'building' graphics, then set up display colours and initialise variables.

**Lines 160-190:** Display various buildings of random height.

**Lines 200-280:** Move a randomly selected alien ship, checking to see whether or not it has landed on a building.

**Lines 290-320:** Check the keyboard and move the launcher if Z or X has been pressed.

**Lines 330-380:** Fire the missile. The computer jumps to line 390 if a ship is hit.

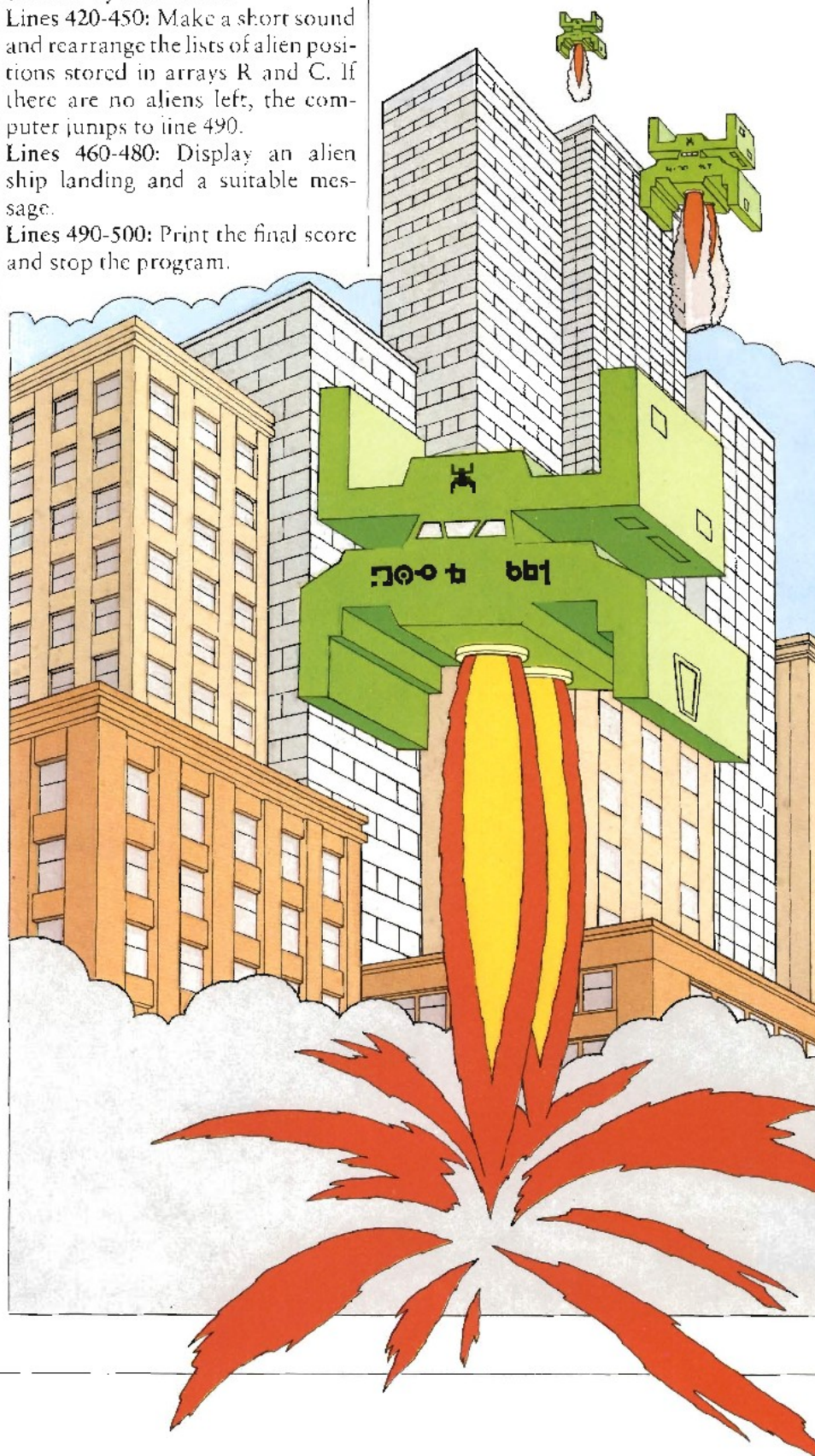
**Lines 390-410:** Calculate the new score and determine which ship has

been hit by the missile.

**Lines 420-450:** Make a short sound and rearrange the lists of alien positions stored in arrays R and C. If there are no aliens left, the computer jumps to line 490.

**Lines 460-480:** Display an alien ship landing and a suitable message.

**Lines 490-500:** Print the final score and stop the program.





## ★ BBC/ELECTRON

```

130 MODE 1:VDU 23,1,0;
    0;0;0;19,1,5;0;
135 *FX 11,20
140 COLOUR 1:V=30:W=38:
    H=19
200 COLOUR 2:A=RND(N+2)
290 FOR T=1 TO 4:K$=
    INKEY$(8)
340 COLOUR 3:M=POINT
    (32*H,1024-32*L)-1
420 SOUND 1,-15,53,4:
    PRINT TAB(H,L);" "
495 *FX 12
    
```

## ★ COMMODORE 64

```

130 POKE 53280,0:POKE
    53281,0:POKE 646,4
140 V=23:H=19:PRINT C$:
    POKE 650,128:W=38
200 POKE 646,7:HI=10:
    LO=0:A=FNR(N+2)
290 FOR T=1 TO 4:GET K$
292 FOR Z=1 TO 10:IF K$
    ="" THEN GET K$
294 NEXT Z
340 POKE 646,1:M=SGN(PE
    EK(984+40*L+H)-32)
420 DN=60:GOSUB 8000:
    GOSUB 6000:PRINT" "
    
```

## SPECTRUM LISTING

```

100 REM INVADERS
110 GOSUB 1000:LET BL=1:GOSUB 2000
120 DATA 255,153,153,255,255,153,153,
    255,0,129,126,24,126,255,231,189
•130 BORDER 0:PAPER 0:CLS:LET J=0.1
•140 INK 3:LET V=21:LET W=30:LET H=15
150 DIM R(W):DIM C(W):DIM H(W):LET S=0
160 LET N=W:LET L$=CHR$(94):FOR C=1 TO W
170 LET C(C)=C:LET H(C)=V-5+FNR(4)
180 FOR R=H(C) TO V-1:GOSUB 4000:PRINT U$
190 NEXT R:NEXT C:LET P=H
•200 INK 6:LET A=FNR(N+2)
210 IF A>N THEN GOTO 290
220 LET R=R(A):LET C=C(A):GOSUB 4000
230 PRINT" ";GOSUB 5000:PRINT L$
240 LET R=R+1:LET C=C+FNR(3)-2
250 LET R(A)=R:IF C<1 THEN LET C=1
260 IF C>W THEN LET C=W
270 LET C(A)=C:GOSUB 4000:PRINT Z$
280 IF R>H(C)-2 THEN GOTO 460
•290 FOR T=1 TO 4:BEEP J,-V:LET K$=INKEY$
300 IF K$="Z" AND H>1 THEN LET H=H-1:
    GOSUB 5000:PRINT L$;" ":LET P=H
310 IF K$="X" AND H<W THEN GOSUB 5000:
    PRINT" ";L$:LET H=H+1:LET P=H
320 NEXT T:IF K$<>"F" THEN GOTO 200
330 LET L=H(H)-1:GOSUB 6000:PRINT"!"
•340 INK 7:LET M=POINT(8*H,178-8*L)
350 GOSUB 6000:PRINT" ":LET L=L-1
360 IF L=0 THEN GOTO 200
370 IF M=1 THEN LET N=N-1:GOTO 390
380 GOSUB 6000:PRINT"!":GOTO 340
390 LET K=1:LET S=S+5*(30+W-N-L)
400 IF R(K)=L AND C(K)=H THEN GOTO 420
410 LET K=K+1:GOTO 400
•420 BEEP 0.2,0:PRINT AT L,H;" "
430 IF N=0 THEN LET M$="YOU HAVE SAVED
    LONDON":LET S=S+1000:GOTO 490
440 FOR T=K TO N:LET R(T)=R(T+1)
450 LET C(T)=C(T+1):NEXT T:GOTO 200
460 FOR R=H(C)-1 TO V:GOSUB 4000:PRINT" "
470 NEXT R:LET R=V-1:GOSUB 4000:PRINT Z$
480 LET M$="THE INVADERS HAVE LANDED"
490 LET R=2:LET C=2:GOSUB 4000:PRINT M$
500 PRINT:PRINT"      YOU SCORED ";S:STOP
    
```



## GAME NOTES

In this game the computer tries to guess your favourite food. It asks questions to which you can reply by entering capital Y for yes or capital N for no.

At first the computer does not do very well. However, if you play several games the computer will build up a question 'tree' by asking players to enter extra questions that have yes or no answers.

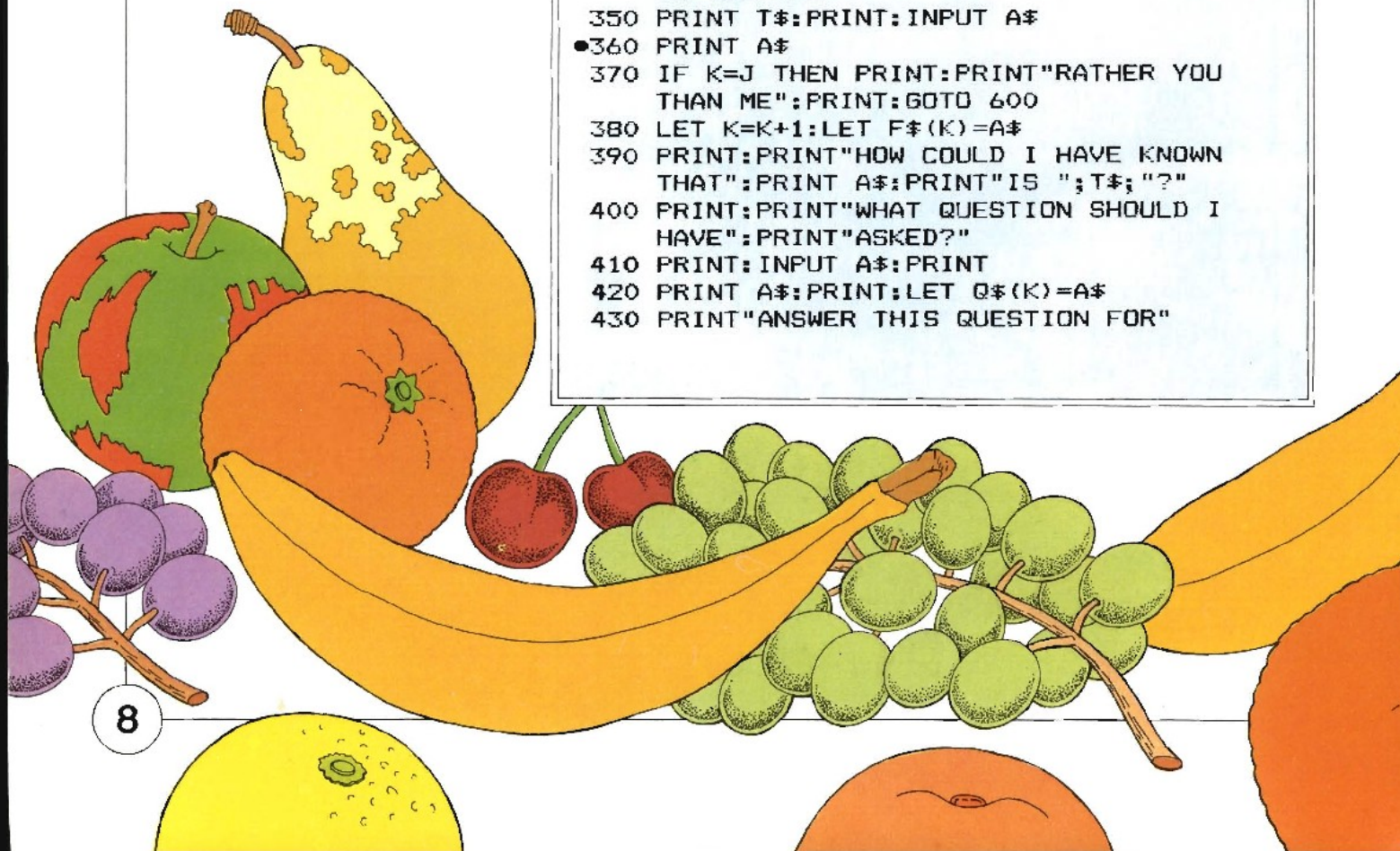
If the computer cannot answer your question the message "SUGGEST ANOTHER QUESTION" will appear. For instance, the question "WHAT COLOUR IS IT?" would be rejected because the answer is neither Y nor N. In the same way, if your favourite food is LEMONS, and the computer guesses BANANAS, the question "IS IT YELLOW?" would be no

## SPECTRUM LISTING

```

100 REM BANANAS
•110 LET J=50: DIM Q$(J,50): DIM F$(J,32)
120 FOR C=1 TO 4: READ Q$(C), F$(C): NEXT C
130 DIM Y(J): DIM N(J): READ Y(2), N(2), K, T$
140 LET Y(3)=J+1: LET Y(4)=J+3
150 LET N(3)=J+2: LET N(4)=J+4
•160 BORDER 1: PAPER 1: INK 7: CLS
200 PRINT: PRINT "I AM GOING TO GUESS ":
    PRINT T$: PRINT
210 PRINT "PLEASE DECIDE WHAT IT IS":
    PRINT "THEN PRESS A KEY": LET C=2
•220 PAUSE 0: CLS
230 IF C>K THEN GOTO 300
240 LET D=C: PRINT Q$(C): INPUT A$: PRINT
250 IF A$="Y" THEN LET C=Y(C): GOTO 230
260 IF A$="N" THEN LET C=N(C): GOTO 230
270 PRINT "ENTER Y OR N": PRINT: GOTO 240
300 PRINT: PRINT T$; " IS": PRINT F$(C-J)
310 PRINT "ISN'T IT?": INPUT A$: PRINT
320 IF A$="Y" THEN PRINT "THERE'S NO
    ACCOUNTING FOR TASTES": GOTO 600
330 IF A$<>"N" THEN GOTO 310
340 PRINT "I GIVE UP": PRINT "YOU'LL HAVE
    TO TELL ME"
350 PRINT T$: PRINT: INPUT A$
•360 PRINT A$
370 IF K=J THEN PRINT: PRINT "RATHER YOU
    THAN ME": PRINT: GOTO 600
380 LET K=K+1: LET F$(K)=A$
390 PRINT: PRINT "HOW COULD I HAVE KNOWN
    THAT": PRINT A$: PRINT "IS "; T$; "?"
400 PRINT: PRINT "WHAT QUESTION SHOULD I
    HAVE": PRINT "ASKED?"
410 PRINT: INPUT A$: PRINT
420 PRINT A$: PRINT: LET Q$(K)=A$
430 PRINT "ANSWER THIS QUESTION FOR"

```





```

440 PRINT F$(C-J):INPUT A$:PRINT
450 IF A$="Y" THEN LET Y(K)=C
460 IF A$="N" THEN LET N(K)=C
470 PRINT"ANSWER THIS QUESTION FOR"
480 PRINT F$(K):INPUT A$:PRINT
490 IF A$="Y" THEN LET Y(K)=K+J
500 IF A$="N" THEN LET N(K)=K+J
510 IF Y(K)*N(K)=0 THEN PRINT"SUGGEST
    ANOTHER QUESTION":PRINT:GOTO 410
520 PRINT"ARE YOU HAPPY WITH THE
    QUESTION?":INPUT A$:PRINT
530 IF A$<>"Y" THEN GOTO 410
540 IF Y(D)=C THEN LET Y(D)=K
550 IF N(D)=C THEN LET N(D)=K
600 INPUT"ANOTHER GAME";A$
*610 CLS:IF A$="Y" THEN GOTO 200
620 STOP
800 DATA "", "RADISHES", "IS IT SOLD AT
    THE GREENGROCERS?", "BANANAS"
810 DATA "IS IT A SALAD VEGETABLE?",
    "LIQUORICE", "IS IT SWEET?", "KIPPERS"
820 DATA 3,4,4, "YOUR FAVOURITE FOOD"
    
```

good because it has the answer Y for both fruits.

The program allows the computer to store up to 49 questions and 50 types of food. If your machine has room for more questions in its memory, you may increase the value of J in line 110.

Whenever you play this game remember that typing RUN clears the computer's memory. To restart the game without losing the current stock of questions, type GOTO 200.

Spectrum users can record the program together with all stored questions, using the command SAVE "BANANAS" LINE 200.

## PROGRAM NOTES

Lines 100-160: Reserve memory space for lists of questions and answers, store initial data and set display colours.

Lines 200-220: Give you time to decide on your favourite food.

Line 230: Sends the computer to the 'guess' routine if there are no more questions to ask.

Lines 240-270: Print a series of questions using the numbers stored in arrays Y and N. For example, if the answer to question Q\$(5) is Y then the number of the next question in the 'tree' is Y(5).

Lines 300-370: Print the computer's guess and jump to the end of the program if the guess is correct or if the reserved memory space is full.

Lines 380-550: Input an extra question and answers to it, then store appropriate numbers in arrays Y and N.

Lines 600-620: Send the computer back to the start of the question sequence or stop the program.

Lines 800-820: Store questions and answers.

## ★ BBC/ELECTRON

```

110 J=50:DIM Q$(J),F$(J)
160 MODE 6:VDU 19,0,4,0;
220 PAUSE=GET:CLS:PRINT
610 CLS:IF A$="Y" THEN GOTO 200
    
```

## ★ COMMODORE 64

```

110 J=50:DIM Q$(J),F$(J):PRINT CHR$(147)
160 POKE 53280,6:POKE 53281,6:POKE 646,1
220 GET G$:IF G$="" THEN 220
225 PRINT CHR$(147)
610 PRINT CHR$(147):IF A$="Y" THEN 200
    
```



## GAME NOTES

A spider has trapped a number of juicy flies in her web. She now wants to tour her larder, visiting each meal in turn.

The aim of the game is to move the spider from fly to fly and then back to her starting point in as few moves as possible. You start by choosing a level of difficulty from 1 (easy) to 6 (difficult).

On the screen the spider is shown as an asterisk and her victims are labelled A, B, C and so on. The spider can move in eight different directions. A diagram shows how to make these moves. She must inspect her prey in alphabetical order and may not land on any square twice.

If you want the computer to restart the game with the spider and flies in the same positions, enter Y when you see the message "SAME WEB AGAIN?" or type GOTO 280. You or a friend can then try for a lower score.

## PROGRAM NOTES

Lines 100-130: Define two graphics characters and reserve memory space.

Lines 140-170: Store the spider's eight moves in arrays V and H.

Lines 180-270: Set up display colours, input level of difficulty and select random positions for the spider and up to seven flies.

Lines 280-370: Display the web and the choice of moves.

Lines 380-480: Input your move, erase previous messages, then check that your chosen move is legal.

Lines 490-540: Move the spider and jump back to line 380 for a new

## SPECTRUM LISTING

```

100 REM SPIDER
110 GOSUB 1000:LET BL=30:GOSUB 2000
120 DATA 255,129,129,129,129,129,129,
    255,255,255,255,255,255,255,255
130 DIM R(9):DIM C(9):DIM V(8):DIM H(8)
140 LET P=0:FOR N=1 TO 4:READ D,E
150 LET V(N)=D:LET V(N+4)=-D
160 LET H(N)=E:LET H(N+4)=-E:NEXT N
170 DATA -2,1,-1,2,1,2,2,1
•180 BORDER 6:PAPER 6:CLS
•190 PAPER 7:INK 0
200 INPUT"LEVEL (1-6) ";W:GOSUB 3000
210 LET W=W+2:FOR N=1 TO W:LET K=1
220 LET R=FNR(16)+2
230 LET C=FNR(16)+4
240 IF N=1 THEN GOTO 270
250 IF R=R(K) AND C=C(K) THEN GOTO 220
260 LET K=K+1:IF K<N THEN GOTO 250
270 LET R(N)=R:LET C(N)=C:NEXT N
280 LET T=0:FOR V=3 TO 18:FOR H=5 TO 20
290 GOSUB 5000:PRINT U$:NEXT H:NEXT V
300 FOR V=6 TO 10:FOR H=25 TO 29
310 GOSUB 5000:PRINT U$:NEXT H:NEXT V
320 LET V=8:LET H=27:GOSUB 5000:GOSUB 600
330 FOR N=1 TO 8:LET K=N+48:LET H=27+H(N)
340 LET V=8+V(N):GOSUB 5000:PRINT CHR$(K)
350 NEXT N:GOSUB 4000:GOSUB 600:LET F=1
360 FOR N=1 TO W-1:LET V=R(N):LET H=C(N)
370 GOSUB 5000:PRINT CHR$(64+N):NEXT N
•380 LET B=0:INK 0
390 INPUT"WHICH MOVE (1-8) ";M:GOSUB 3000
400 IF M<1 OR M>8 THEN GOTO 390
410 LET L=20:GOSUB 6000:FOR N=1 TO 22
•420 INK 6:PRINT Z$;:NEXT N:INK 0
430 LET V=R+V(M):LET H=C+H(M)
440 IF V<3 OR V>18 OR H<5 OR H>20 THEN
    LET W$=" OFF THE WEB! ":LET B=1
•450 IF SCREEN$(V,H)="*" THEN LET B=1:
    LET W$="YOU'VE BEEN THERE!"
460 FOR N=F+1 TO W
470 IF V=R(N) AND H=C(N) THEN LET B=1:
    LET W$="<< WRONG ORDER! >>"
480 NEXT N:IF B=1 THEN LET P=4:GOSUB
    6000:PRINT W$:GOTO 380
490 LET T=T+1:GOSUB 5000:GOSUB 600

```



## SPIDER'S WEB

4

```

500 IF T=1 THEN GOSUB 4000:PRINT Z$
510 IF T>1 THEN GOSUB 4000:PRINT "*"
520 LET R=V:LET C=H:LET V=13:LET H=24
•530 INK 0:IF R=R(F) AND C=C(F) THEN LET
    F=F+1:BEEP 0.2,0
540 IF F<=W THEN GOTO 380
550 GOSUB 5000:PRINT"SCORE ";T
560 INPUT"SAME WEB AGAIN Y/N ";A$
570 GOSUB 3000:IF A$="Y" THEN GOTO 280
580 STOP
•600 INK 2:PRINT INVERSE 1;"*":RETURN
    
```

### ★ BBC/ELECTRON

```

180 MODE 1:COLOUR 130:CLS:X=1004:Y=992
190 COLOUR 131:COLOUR 0:PRINT TAB(0,24);
380 B=0:COLOUR 0:PRINT TAB(0,24);
420 COLOUR 2:PRINT Z$;:NEXT N:COLOUR 0
450 IF POINT(16+H*32,X-V*32)=1 AND POINT(H*32,
    Y-V*32)=3 THEN B=1:W$="YOU'VE BEEN THERE!"
530 COLOUR 0:IF R=R(F) AND C=C(F) THEN F=F+1:
    SOUND 1,-15,53,4
555 PRINT TAB(0,24);
600 COLOUR 129:COLOUR 3:PRINT"*":COLOUR 131:
    COLOUR 1:RETURN
    
```

### ★ COMMODORE 64

```

180 POKE 53280,7:POKE 53281,1:POKE 646,7
190 PRINT C$;:FOR N=1 TO 880:PRINT Z$;:NEXT N:
    R=23:C=0:GOSUB 4000:POKE 646,11
380 B=0:POKE 646,11:V=23:H=0:GOSUB 5000
420 POKE 646,7:PRINT Z$;:NEXT N:POKE 646,11
450 IF PEEK(1024+H+40*V)=42 THEN B=1:
    W$="YOU'VE BEEN THERE!"
530 POKE 646,11:IF R=R(F) AND C=C(F) THEN
    F=F+1:HI=20:LO=0:DN=100:GOSUB 8000
555 V=23:H=0:GOSUB 5000
600 PRINT CHR$(28);CHR$(18);"*":RETURN
    
```

move unless the spider has completed her tour.

Lines 550-580: Print your score and give you a chance to restart the game with the same web.

Line 600: A subroutine to print the spider.





## GAME NOTES

The computer is going to draw a grid and set up ten hidden targets for you. Fire at the targets by entering a row number from 0 to 9, then a column letter from A to O.

After each shot a new symbol will appear on the grid in the space at which you fired – a circle for a miss, a square for a near miss or a star for a hit. The game ends when you have found all ten targets.

A score of under 80 shots is good. If you take less than 50 shots

to hit all ten targets, you must be an electronic mind reader.

Remember that two targets may be next to each other!

(If you accidentally stop the program in the middle of a game, type GOTO 280 to carry on.)

## PROGRAM NOTES

Lines 100-200: Initialise variables and draw the grid.

Lines 210-270: Select at random 10 different pairs of co-ordinates for the targets.

Lines 280-330: Input the row number and the column letter of your shot.

Lines 340-370: Check the co-ordin-

## SPECTRUM LISTING

```

100 REM TARGET
110 DIM R(10):DIM C(10):DIM H(10)
•120 BORDER 1:PAPER 1:INK 7:CLS
•130 LET K=0:LET H=0:LET L=21:LET P=22
•140 FOR N=12 TO 252 STEP 16
•150 PLOT N,0:DRAW 0,167:NEXT N
•160 FOR N=4 TO 164 STEP 16
•170 PLOT 8,N:DRAW 247,0:NEXT N
180 GOSUB 2000:LET V=K:GOSUB 5000:PRINT
   " A B C D E F G H I J K L M N O "
190 FOR N=1 TO 10:LET V=2*N+K:GOSUB 5000
200 PRINT CHR$(N+47):NEXT N
210 LET S=0:LET T=0:FOR N=1 TO 10
220 LET R=2*FNR(10)+K
230 LET C=2*FNR(15)+H
240 LET F=1:IF N=1 THEN GOTO 270
250 IF R=R(F) AND C=C(F) THEN GOTO 220
260 LET F=F+1:IF F<N THEN GOTO 250
270 LET R(N)=R:LET C(N)=C:NEXT N
•280 INK 6
290 INPUT"ROW ";R:GOSUB 3000:IF R<0
   OR R>9 THEN GOTO 290
300 LET R=2*R+K+2:LET M=0
310 INPUT"COLUMN ";G$:GOSUB 3000
•320 LET C=2*CODE(G$)-128+H
330 IF C<2+H OR C>30+H THEN GOTO 310
340 LET N=1:LET S=S+1:GOSUB 4000
350 IF R=R(N) AND C=C(N) THEN GOTO 400
360 IF ABS(R-R(N))<3 AND ABS(C-C(N))<3
   THEN LET M=1
370 IF N<10 THEN LET N=N+1:GOTO 350
380 IF M=0 THEN PRINT"O":GOTO 280
•390 PRINT PAPER 6;" ":PAPER 1:GOTO 280
400 IF H(N)=0 THEN LET T=T+1:LET H(N)=1
•410 PRINT FLASH 1;"*":BEEP 0.5,0
420 LET M$=" ":IF T>1 THEN LET M$="S "
430 GOSUB 6000:PRINT" ";T;" HIT";M$
440 IF T<10 THEN GOTO 280
•450 PRINT AT 21,12:PAPER 6:INK 0;
   " YOU TOOK ";S;" SHOTS ":STOP

```



# TARGET PRACTICE

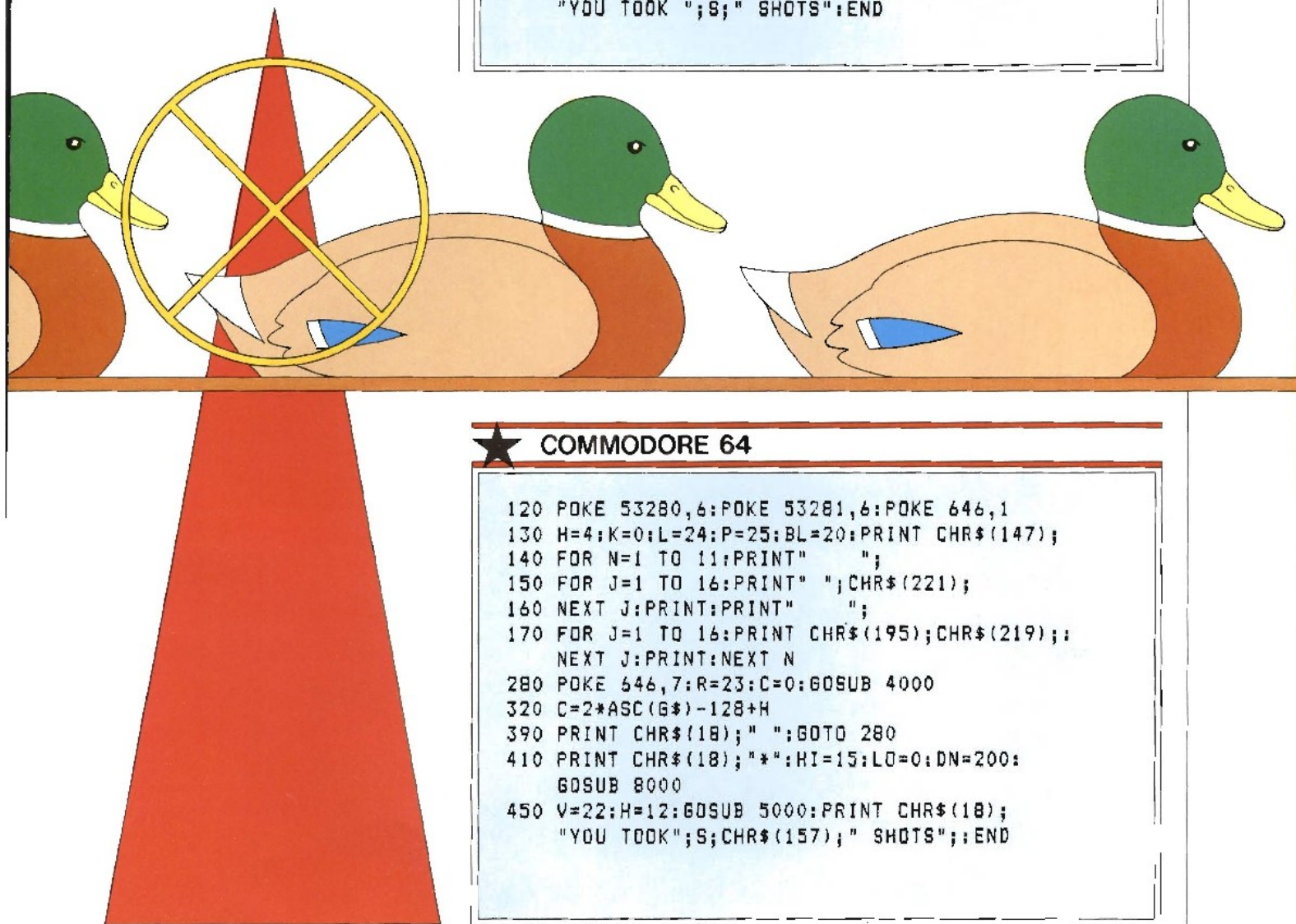
5

ates of the shot against the positions of the targets.

Lines 380-450: Print an appropriate symbol then jump to the input routine or if the game is over display the final score.

## ★ BBC/ELECTRON

```
120 MODE 1:H=4:K=4:L=27:P=17:BL=20
130 VDU 19,0,4;0;19,1,11;0;
140 FOR N=176 TO 1136 STEP 64
150 MOVE N,192:DRAW N,863:NEXT N
160 FOR N=208 TO 848 STEP 64
170 MOVE 160,N:DRAW 1152,N:NEXT N
280 COLOUR 128:COLOUR 2: PRINT TAB(0,29);
320 LET C=2*ASC(G$)-128+H
390 COLOUR 130:PRINT" ":GOTO 280
410 COLOUR 1:PRINT"*":COLOUR 3:SOUND 1,-15,53,6
450 COLOUR 130:COLOUR 0:PRINT TAB(12,29);
    "YOU TOOK ";S;" SHOTS":END
```



## ★ COMMODORE 64

```
120 POKE 53280,6:POKE 53281,6:POKE 646,1
130 H=4:K=0:L=24:P=25:BL=20:PRINT CHR$(147);
140 FOR N=1 TO 11:PRINT" ";
150 FOR J=1 TO 16:PRINT" ";CHR$(221);
160 NEXT J:PRINT:PRINT" ";
170 FOR J=1 TO 16:PRINT CHR$(195);CHR$(219);:
    NEXT J:PRINT:NEXT N
280 POKE 646,7:R=23:C=0:GOSUB 4000
320 C=2*ASC(G$)-128+H
390 PRINT CHR$(18);" ":GOTO 280
410 PRINT CHR$(18);"*":HI=15:LO=0:DN=200:
    GOSUB 8000
450 V=22:H=12:GOSUB 5000:PRINT CHR$(18);
    "YOU TOOK";S;CHR$(157);" SHOTS";:END
```



## GAME NOTES

Your computer is not a good fortune-teller. It has no mystic powers and the only predictions it can make use the information that you, the programmer, put into a program. However, if you do this well you may be able to make at least some of your friends think your computer is a gypsy!

To make this program work you type in your own set of DATA lines after the given listing. Here are some examples:

```
800 DATA "SCARLET",
      "BLUE", "VIOLET"
840 DATA "CHARMING",
      "PLUMP", "BLONDE"
880 DATA "WRESTLER",
      "DENTIST", "CLOWN"
920 DATA "PARIS", "THE
      BATH", "THE ZOO"
960 DATA "JELLY", "NUTS",
      "BLACK PUDDING"
```

The complete set of DATA lines should contain up to 12 colours, 12 descriptive words, 12 jobs, 12 places and 12 kinds of food. If you start at line 800 and put 3 words or phrases in each line, your word list should end around line 990.

Your word list can give the program a personal touch. Try to make anyone who runs the program laugh by putting unlikely words in your DATA lines.

When you have finished, the computer will be ready to ask its first victim to enter their name and birthday. 'Predictions' will then flash on-screen.

## PROGRAM NOTES

Lines 100-130: Set up screen colours and store your word list in an array WS.

## SPECTRUM LISTING

```
100 REM FORTUNES
●110 BORDER 2:PAPER 6:INK 0:CLS
●120 DIM W$(60,32):LET TD=200:GOSUB 2000
130 FOR N=1 TO 60:READ W$(N):NEXT N
140 PRINT:PRINT" WHAT IS YOUR NAME?"
150 PRINT:INPUT N$:PRINT:LET L=LEN(N$)
160 PRINT" WHEN IS YOUR BIRTHDAY?"
170 PRINT:INPUT" DAY (1-31) ";D
180 PRINT:INPUT" MONTH (1-12) ";M
●190 BORDER 3:PAPER 1:INK 7:CLS
●200 PRINT AT 3,13;"HELLO":PRINT:LET C=0
210 IF L<32 THEN LET C=INT((32-L)/2)
220 PRINT TAB(C);N$:PRINT:GOSUB 390
230 PRINT TAB(6);"YOUR LUCKY NUMBER IS"
240 PRINT:PRINT TAB(15);ABS(L-D)+5
250 PRINT:LET P$=W$(M):GOSUB 390
260 PRINT TAB(6);"YOUR LUCKY COLOUR IS"
270 PRINT:GOSUB 400:GOSUB 390
280 PRINT" PRESS ANY KEY TO CONTINUE"
●290 PAUSE 0:BORDER 4:PAPER 0:INK 6:CLS
300 PRINT:PRINT:DIM R(4):FOR N=1 TO 4:
  LET R(N)=FNR(12):NEXT N
310 PRINT" YOU ARE GOING TO MEET A"
320 LET P$=W$(R(1)+12):GOSUB 390
330 GOSUB 400:LET P$=W$(R(2)+24)
340 GOSUB 400:GOSUB 390:LET P$=
  W$(R(3)+36):PRINT TAB(15);"IN"
350 PRINT:GOSUB 400:GOSUB 390
360 PRINT" TOGETHER YOU WILL EAT"
370 LET P$=W$(R(4)+48):PRINT:GOSUB 400
●380 STOP
390 FOR N=1 TO TD:NEXT N:PRINT:RETURN
●400 LET L=32:LET Q$=""
●410 IF P$(L)=Q$ THEN LET L=L-1:GOTO 410
●420 LET P$=P$(TO L):LET C=INT((32-L)/2)
430 PRINT TAB(C);P$:PRINT:RETURN
```



## BBC/ELECTRON

```
110 MODE 6:VDU 19,1,5,0:CLS
120 DIM W$(60):TD=2000:GOSUB 2000
190 VDU 19,0,4,0:19,1,3,0:CLS
200 PRINT TAB(13,3);"HELLO":PRINT:LET C=0
290 PAUSE=GET:VDU 19,0,1,0:19,1,7,0:CLS
380 END
400 L=LEN(P$):C=INT((32-L)/2)
```





Lines 140-180: Input your name and birthday.

Lines 190-270: Set up new colours and print your 'lucky number' and 'lucky colour'. The computer calculates these using the length of your name and your birthday.

Lines 280-290: Wait for you to press a key then clear the screen and set up new colours.

Lines 300-390: Select and store in array R 4 random numbers, used to find the rest of your 'fortune'.

Lines 400-430: Subroutine to find the length and position of the words stored in P\$.

## ★ COMMODORE 64

```

110 POKE 53280,2:POKE 53281,7:
    POKE 646,0:BL=1
120 GOSUB 2000:PRINT C$:DIM W$(60):TD=300
190 POKE 53280,4:POKE 53281,6:POKE 646,1
200 PRINT C$:LEFT$(R$,3);TAB(13);"HELLO":
    PRINT:C=0
290 GET K$:IF K$="" THEN GOTO 290
295 PRINT C$;:POKE 53280,5:POKE 53281,0:
    POKE 646,7
380 END
400 LET L=LEN(P$):LET C=INT((32-L)/2)
    
```



## GAME NOTES

Imagine a snow-covered mountain-side in the Austrian Alps. Television viewers are waiting to see world-class skiers tackle a long, difficult slalom, speeding downhill through a series of flag-marked 'gates'.

You are going to control one of the skiers, using the Z key to move left and the X key to move right.

Your skier must stay on course and will be disqualified if he misses a 'gate'. The longer he survives, the narrower the gap between the flags becomes.

You can make the game more difficult at the start by reducing the value of J in line 150.

## PROGRAM NOTES

Lines 100-140: Define the 'skier' and 'flag' characters, then set up text and background colours.

Lines 150-180: Store initial values of the variables.

Lines 190-200: Select a random position for the left-hand flag.

Lines 210-220: Scroll the screen and print the skier.

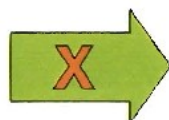
Line 230: Ensures that the right-hand flag is on screen.

Line 240: Checks that the skier is between the flags.

Lines 250-260: Print a new pair of flags.

Lines 270-330: Move the skier if the Z or X keys are pressed.

Lines 400-410: Print the score and stop the program.



## SPECTRUM LISTING

```

100 REM SLALOM
110 GOSUB 1000:LET BL=1:GOSUB 2000
120 DATA 48,48,24,252,27,48,31,248
130 DATA 128,241,255,255,143,129,129,1
•140 BORDER 7:PAPER 7:INK 0:CLS
•150 LET W=30:LET D=21:LET G=10:LET J=7
•160 LET Z=3582:POKE 23561,10
170 LET R=D-J:LET C=W/2:LET NC=C
180 LET L=C-5:LET NL=L:LET T=0
190 LET M=FNR(11)-6:LET L=NL
200 LET NL=L+M:IF NL<1 THEN LET NL=1
•210 PRINT AT R,C;" ":LET A=USR Z
220 LET C=NC:GOSUB 4000:PRINT U$
230 IF NL+G>W THEN LET NL=W-G
240 IF C<L OR C>L+G THEN GOTO 400
250 LET G=10-INT(T/20):LET T=T+1
•260 PRINT INK 2;AT D,NL;Z$;TAB(NL+G);Z$
270 FOR N=2 TO J
•280 PRINT AT R,C;" ":LET A=USR Z
290 LET C=NC:GOSUB 4000:PRINT U$
300 GOSUB 7000
310 IF K$="Z" AND C>0 THEN LET NC=C-1
320 IF K$="X" AND C<=W THEN LET NC=C+1
330 NEXT N:GOTO 190
400 LET V=10:LET H=1:GOSUB 5000:PRINT
  "YOUR SCORE IS ";T*10
•410 POKE 23561,35:STOP

```



## BBC/ELECTRON

```

140 MODE 1:COLOUR 131:COLOUR 0:CLS
150 VDU 23,1,0;0;0;0;W=38:D=30:G=10:J=10
160 *FX 11,20
210 PRINT TAB(C,R);" ";TAB(0,D):PRINT
260 COLOUR 1:PRINT TAB(NL,D);Z$;TAB(NL+G,D);
  Z$:COLOUR 0
280 PRINT TAB(C,R);" ";TAB(0,D):PRINT
325 FOR K=1 TO 80:NEXT K
410 *FX 12,0
420 END

```

NB Electron users should change line 325 by substituting 10 (or less) for 80.



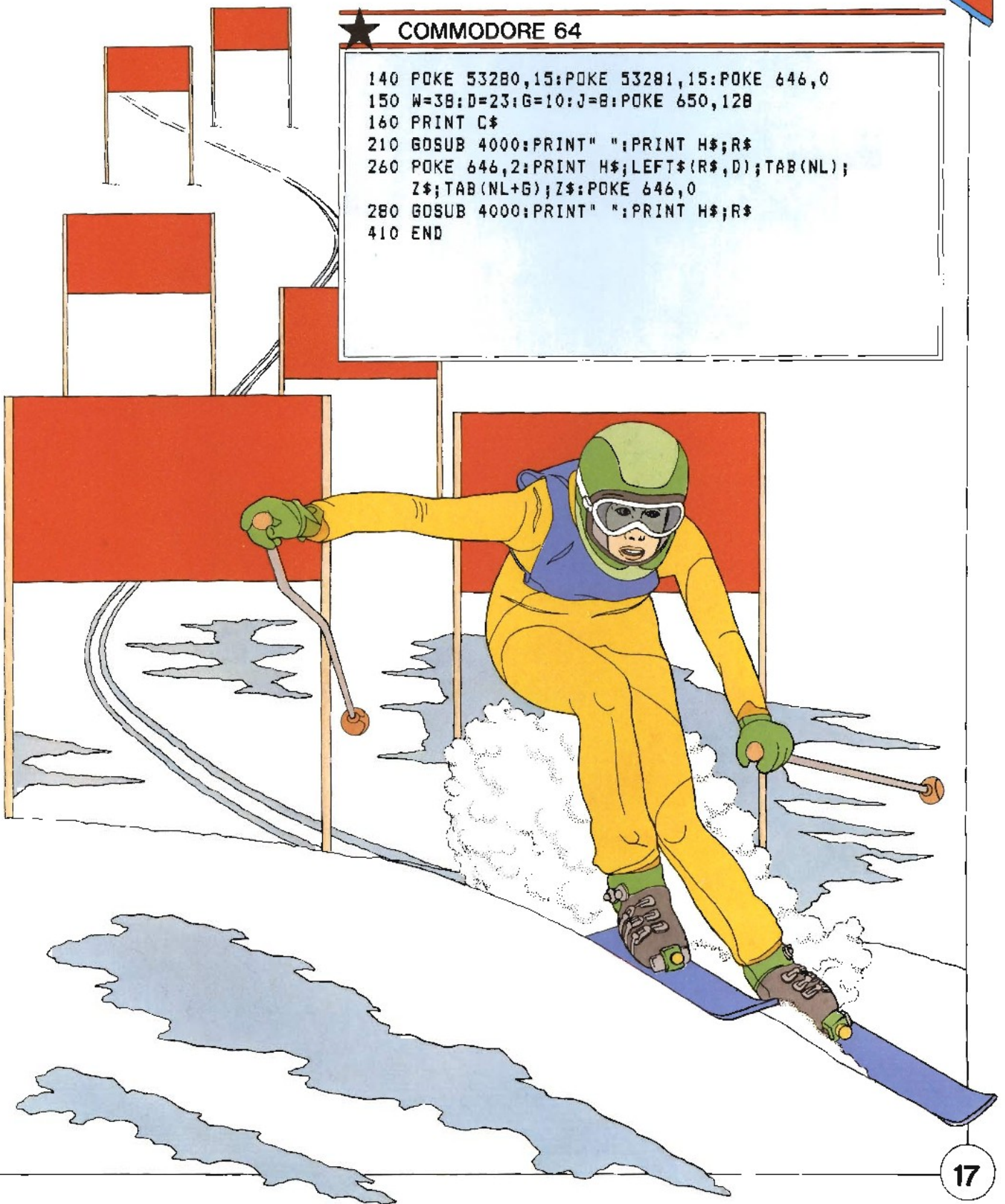
# GRAND SLALOM

7



## COMMODORE 64

```
140 POKE 53280,15:POKE 53281,15:POKE 646,0
150 W=38:D=23:G=10:J=8:POKE 650,128
160 PRINT C$
210 GOSUB 4000:PRINT " ":PRINT H$:R$
260 POKE 646,2:PRINT H$:LEFT$(R$,D);TAB(NL);
    Z$;TAB(NL+G);Z$:POKE 646,0
280 GOSUB 4000:PRINT " ":PRINT H$:R$
410 END
```





## GAME NOTES

You are a hi-tech locksmith working for Counter Intelligence. Your job is to make a copy of the 'key' to an ultra-sophisticated laser lock.

The key consists of nine mirror units that are fixed in various positions. You must find the correct combination of their positions by testing the lock with pulses of laser light.

The secret combination uses four mirror unit positions:

H - horizontal pair of mirrors

V - vertical pair of mirrors

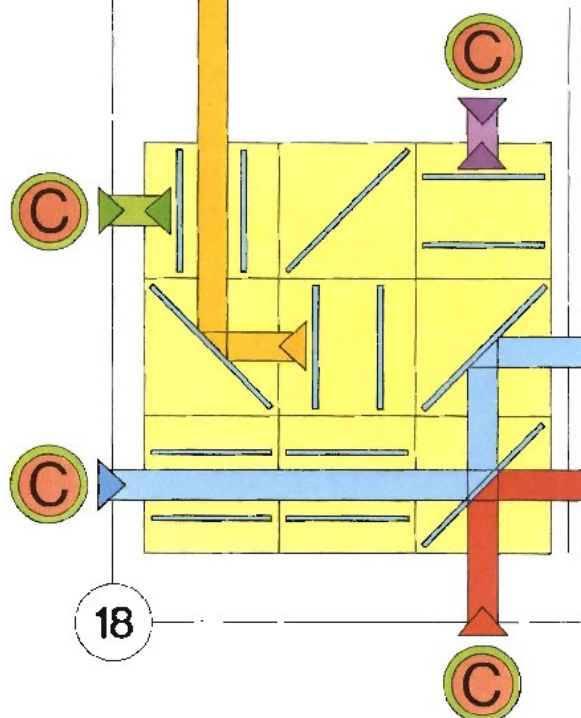
F - forward sloping mirror

B - backward sloping mirror

Positions of mirror units

V	F	H
B	V	F
H	H	F

C - move cursor  
L - fire laser  
M - guess mirror position



## SPECTRUM LISTING

```

100 REM LASER
110 GOSUB 1000:LET BL=30:GOSUB 2000
120 DATA 255,255,255,255,255,255,255,
    255,0,20,20,20,20,20,20,0,3,6,3,9,3
•130 BORDER 0:PAPER 0:INK 7:CLS
•140 LET D=4:LET L=21:LET P=0:DIM C$(4,1)
150 DIM M(9):DIM X(12):DIM Y(12)
160 FOR J=1 TO 12:READ X(J),Y(J):NEXT J
170 DATA 12,8,17,11,17,14,17,19,12,19,
    9,19,6,14,1,11,1,8,1,"H","V","F","B"
180 FOR J=1 TO 4:READ C$(J):NEXT J
190 FOR J=1 TO 9:LET M(J)=FNR(4):NEXT J
200 FOR R=7 TO 15:FOR C=5 TO 13
210 GOSUB 4000:PRINT U$:LET V=R
220 LET H=C+17:GOSUB 5000:PRINT U$
230 NEXT C:NEXT R:GOSUB 600:LET N=49
240 GOSUB 900:GOSUB 600:LET V=3:LET H=6
250 LET M=0:LET T=0:LET G=12:GOSUB 700
•260 IF INKEY$<>" " THEN GOTO 260
•270 IF INKEY$="" THEN GOTO 270
•280 LET K$=INKEY$
290 LET F=0:IF K$="C" THEN GOSUB 700
300 IF K$="M" THEN LET M=M+1:GOTO 430
310 LET N=3: IF K$<>"L" THEN GOTO 260
320 LET Y=INT((Y(G)-6)/9):LET B=Y(G)
330 LET X=INT((X(G)-8)/9):LET A=X(G)
340 GOSUB 800:LET A=A-5*X:LET B=B-5*Y
350 LET S=Y:LET U=M(A+B/3-9)
360 IF U=1 THEN LET X=-1*X
370 IF U=2 THEN LET Y=-1*Y
380 IF U=3 THEN LET Y=-X:LET X=-S
390 IF U=4 THEN LET Y=X:LET X=S
400 LET A=A-3*X:LET B=B-3*Y:IF A>7 AND
    A<15 AND B>5 AND B<13 THEN GOTO 350
410 LET A=A+2*X:LET B=B+2*Y:LET N=5
420 GOSUB 800:GOSUB 720:GOTO 260
430 LET N=49:GOSUB 900:FOR J=1 TO 9
•440 INPUT("MIRROR UNIT ";J;" ?");A$
450 LET Z=INT((J-1)/3):LET D$=C$(M(J))
460 IF D$<>A$ THEN LET F=F+5:GOTO 480
470 LET R=3*Z+8:LET C=3*J-9*Z+20:
    GOSUB 4000:PRINT D$
480 NEXT J:LET T=T+M*F:LET M$="CORRECT -"
490 LET Q=1:IF F>0 AND M<4 THEN GOTO 260
500 IF F>0 THEN LET M$="YOU BLEW IT -"
510 FOR J=3 TO 5:FOR C=23 TO 29 STEP 3
520 LET R=3*J-1:GOSUB 4000:PRINT C$(M(Q))

```



The diagrams below left show a combination of mirror units being tested by laser beams fired from different positions.

To play the game use capital C to move the cursor – the laser gun – to the position you want. Press L to fire a laser pulse, or M when you are ready to guess the mirror combination.

Enter your guesses with the letters H, V, F and B. The computer will display correct guesses but refuse wrong ones. You also get penalty points for wrong guesses. As there are only four chances to get the right combination, you will find it very helpful to keep notes as you go along.

## PROGRAM NOTES

Lines 100-130: Define two graphics characters and set up colours.

Lines 140-250: Produce the screen display, select 9 mirror units at random and store the 12 positions of the cursor in arrays X and Y.

Lines 260-310: Wait for you to press a key then jump to the appropriate line.

Lines 320-420: Send a laser pulse through the 'key'.

Lines 430-490: Input your solution.

Lines 500-540: Display the correct combination of mirror units and the outcome of the problem.

Lines 600-920: Subroutines to display coloured lights and laser pulses, move the cursor and print numbers 1 to 9 in position.

```
530 LET Q=Q+1:NEXT C:NEXT J:GOSUB 6000
540 PRINT M$;" ";T;" PENALTY POINTS":STOP
600 FOR J=3 TO 5:FOR C=6 TO 12 STEP 3
  610 LET K=FNR(7)-1:INK K:LET R=3*J-1
  620 GOSUB 4000:PRINT U$:NEXT C:NEXT J
  630 INK 7:RETURN
700 GOSUB 5000:PRINT " "
710 LET G=G+1:IF G=13 THEN LET G=1
720 LET V=X(G):LET H=Y(G):GOSUB 5000:
  PRINT"*":RETURN
800 LET L$="":IF Y=0 THEN LET L$=Z$
810 FOR J=1 TO N:LET R=A-J*X:LET C=B-J*Y:
  GOSUB 4000:PRINT L$:NEXT J
820 FOR W=1 TO D:GOSUB 600:NEXT W
830 FOR J=1 TO N:LET R=A-J*X:LET C=B-J*Y:
  GOSUB 4000:PRINT " ":NEXT J:RETURN
900 FOR J=3 TO 5:FOR C=23 TO 29 STEP 3
  910 LET R=3*J-1:GOSUB 4000:PRINT CHR$(N)
  920 LET N=N+1:NEXT C:NEXT J:RETURN
```

## ★ BBC/ELECTRON

```
130 MODE 1:VDU 23,1,0;0;0;0;CLS
140 D=20:L=21:P=3:DIM C$(4)
260 LET K$=GET$
440 PRINT TAB(0,23);"MIRROR UNIT ";J;:
  INPUT " ? "A$:GOSUB 3000
610 LET K=RND(3)-1:COLOUR K:LET R=3*J-1
630 COLOUR 3:RETURN
```

## ★ COMMODORE 64

```
130 POKE 53280,0:POKE 53281,0:POKE 646,12
140 PRINT C$:D=5:L=21:P=0:DIM C$(4)
260 GET K$:IF K$<>" " THEN 260
270 GET K$:IF K$="" THEN 270
440 PRINT H$:LEFT$(R$,23);"MIRROR UNIT ";J;
  " ";:INPUT A$:GOSUB 3000
610 LET K=FNR(10)*1:POKE 646,K:R=3*J-1
630 POKE 646,7:RETURN
```



## GAME NOTES

Your computer can produce all kinds of exciting graphics simply using keyboard characters. With this program you can create many colourful displays.

To plan your design, think of the screen as a piece of squared paper with numbered rows and columns. (The Spectrum uses rows 0 to 21 and columns 0 to 31. The BBC/Electron: rows 0 to 29, columns 0 to 19. The Commodore: rows 0 to 21, columns 0 to 39.)

To define the screen area in which you will be working, select top and bottom row numbers, then left and right column numbers. The computer will fill the chosen area by printing a repeating pattern up to 10 characters long.

If you decide on a pattern length of 4 then the computer will ask you 4 times to enter a character. In each case you also choose the colours in which the character will be displayed. For example, an Electron user might begin by entering \*(RETURN) followed by 3(RETURN) for blue and then 4(RETURN) for yellow to get a blue background and a yellow star.

You can use all the colours and characters on your computer to build an endless variety of patterns.

If you stop the program by accident type GOTO 200 to continue.

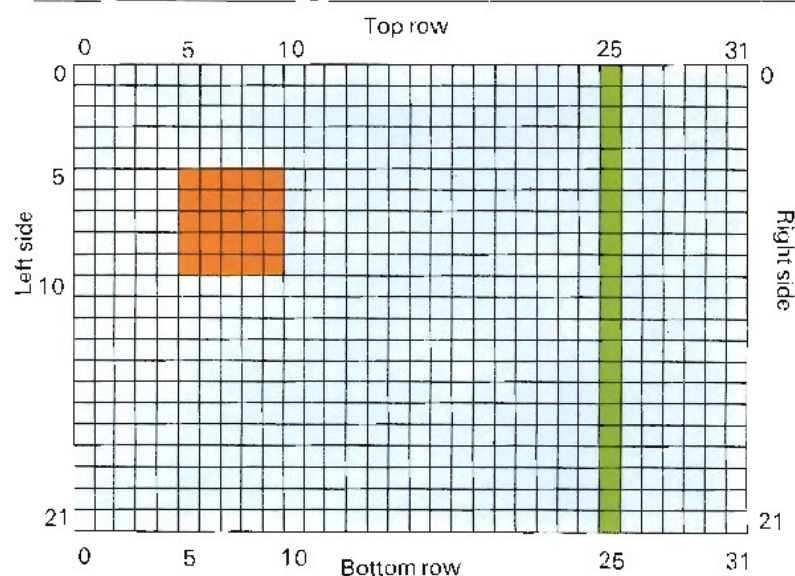
The diagram on the right shows two designs on a Spectrum screen. A square of solid colour fills the area defined by columns 5 to 9 and rows 5 to 9. A band of colour fills column 25 from top to bottom (from row 0 to row 21). On page 21 is a block pattern you could make. Remember you can print flashing colours, graphics symbols and letters on screen.

## SPECTRUM LISTING

```

100 REM BLOCKS
•110 BORDER 0:PAPER 0:INK 7:CLS
•120 DIM P$(10,1):DIM B(10):DIM F(10)
•130 DIM T(10):LET W=31:LET D=21
200 INPUT"TOP ROW ";TR:GOSUB 3000
210 INPUT"BOTTOM ROW ";BR:GOSUB 3000:
    IF BR>D THEN LET BR=D
220 INPUT"LEFT COLUMN ";LC:GOSUB 3000
230 INPUT"RIGHT COLUMN ";RC:GOSUB 3000:
    IF RC>W THEN LET RC=W
240 INPUT"PATTERN LENGTH ";N:GOSUB 3000
250 FOR J=1 TO N
•260 INPUT("CHARACTER ";J;" ");P$(J)
•270 INPUT"PAPER COLOUR (0 TO 8) ";B(J)
•280 INPUT"INK COLOUR (0 TO 8) ";T(J)
•290 INPUT"FLASH (0 OR 1) ";F(J)
300 GOSUB 3000:NEXT J:LET J=1
310 FOR R=TR TO BR
320 FOR C=LC TO RC
•330 PRINT AT R,C;PAPER B(J);INK T(J);
    FLASH F(J);P$(J)
340 LET J=J+1:IF J>N THEN LET J=1
350 NEXT C:NEXT R
360 INPUT"ANOTHER GO (Y/N)";A$
370 GOSUB 3000:IF A$="Y" THEN GOTO 200
•380 STOP

```





## PROGRAM NOTES

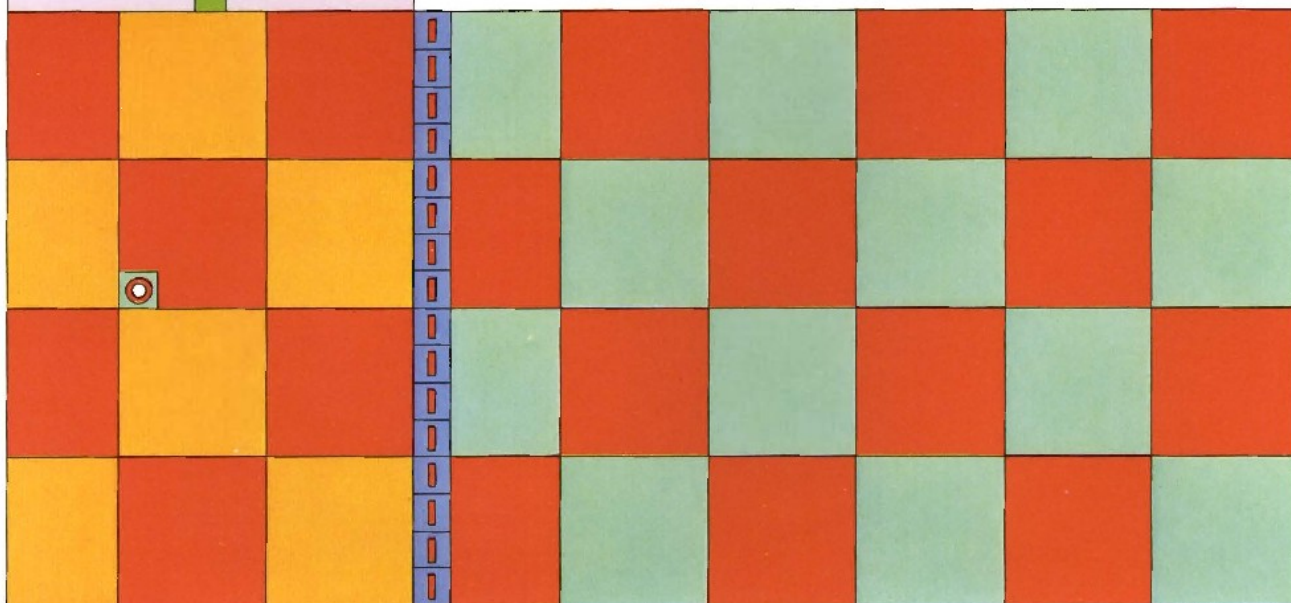
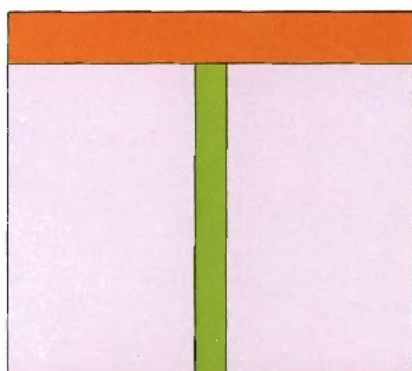
Lines 100-130: Clear the screen and reserve memory space for your pattern.

Lines 200-240: Input the position of the pattern block and the number of characters to be used.

Lines 250-300: Input a sequence of colour codes and characters.

Lines 310-350: Print the chosen character sequence until the pattern block is complete.

Lines 360-370: Send the computer back to the input routine at line 200 if you wish to print another pattern block.



## ★ BBC/ELECTRON

```

110 MODE 2:PRINT TAB(0,30);:W=19:D=29
120 BL=19:DIM P$(10),B(10),T(10)
260 PRINT"CHARACTER ";J;" ";:INPUT LINE P$(J)
270 GOSUB 3000:INPUT"BACKGROUND(0-15)";B(J)
280 GOSUB 3000:INPUT"TEXT(0-15)";T(J)
330 COLOUR 128+B(J):COLOUR T(J):PRINT
    TAB(C,R);LEFT$(P$(J),1)
355 COLOUR 128:COLOUR 7:PRINT TAB(0,30);
380 END

```

## ★ COMMODORE 64

```

110 BL=30:GOSUB 2000:POKE 53280,0:POKE 53281,0
120 POKE 646,15:PRINT C$;LEFT$(R$,23);
130 DIM P$(10),R(10),T(10):W=39:D=21
260 PRINT"CHARACTER";J;:INPUT P$(J)
270 GOSUB 3000:INPUT"TEXT COLOUR (0-15) ";T(J)
280 GOSUB 3000:INPUT"REVERSE (0 OR 1) ";A
290 IF A=0 OR A=1 THEN R(J)=A
330 POKE 646,T(J):PRINT H$;TAB(C);LEFT$(R$,R);
335 PRINT CHR$(18+128*R(J));LEFT$(P$(J),1);
355 POKE 646,15:PRINT H$;LEFT$(R$,23);
    CHR$(146);
380 PRINT CHR$(145);:END

```



## GAME NOTES

Mu Torere is a traditional Maori game from New Zealand. The board consists of an 8 pointed star and a central area called the 'putahi'. The original boards were either drawn on the ground or marked out on the bark of a tree.

Now you can play Mu Torere against the computer. You have the black pieces (numbered 1 to 4) and the computer has the red pieces. This is a blocking game. The aim is to blockade the computer's pieces so that they cannot move.

There are three types of move:

- (1) from the centre to a point of the star;
- (2) from one point to an adjacent point;
- (3) from a point of the star to the centre (but only if one or both neighbouring points are occupied by the enemy).

The best way to learn the game is by playing it as the computer will not allow you to break the rules!

(When typing the program Spectrum users should note that the keywords DEF FN and FN are used in several places.)

## PROGRAM NOTES

Lines 100-240: Display the board, store the initial positions of the pieces using arrays R, C, P and Q, then decide which player starts.

Lines 250-380: Determine the computer's move. Lines 260-310 enable the computer to win in the current turn if possible. Otherwise lines 320-370 select a valid move at random. Line 380 is reached only if the computer cannot move.

Lines 400-430: Decide whether the computer has won.

## SPECTRUM LISTING

```

100 REM MU-TORERE
•110 BORDER 1:PAPER 7:INK 0:GOSUB 2000
•120 CLS:LET V=0:LET H=0:INVERSE 0
130 FOR J=0 TO 120:LET A=PI*J/60
140 LET K=9-5*ABS(SIN(4*A))
150 LET R=INT(V+10.5-K*SIN(A))
160 LET C=INT(H+16.5-K*COS(A))
170 GOSUB 4000:PRINT"*":NEXT J:DIM M(5)
180 DIM R(9):DIM C(9):DIM P(9):DIM Q(9)
190 FOR J=1 TO 9:LET P(J)=J:LET Q(J)=J
200 READ R(J),C(J):LET R(J)=V+R(J)
210 LET C(J)=H+C(J):NEXT J:READ S,X
220 DATA 10,7,3,9,1,16,3,23,10,25,17,23,
    19,16,17,9,10,16,9,4
•230 LET V=20:LET H=1:INVERSE 1:GOSUB 700
240 IF FNR(2)=1 THEN LET W=1:GOTO 400
250 LET M$=" MY TURN ":GOSUB 900
260 IF S=9 OR Q(9)<5 THEN GOTO 320
270 LET F=FNC(S):LET G=FNA(S)
280 LET Y=Q(F)>4 AND Q(FNC(F))>4
290 LET Z=Q(G)>4 AND Q(FNA(G))>4
300 IF Y THEN LET N=Q(F):GOTO 350
310 IF Z THEN LET N=Q(G):GOTO 350
320 FOR J=1 TO 5:LET M(J)=J+4:NEXT J
330 LET K=FNR(X):LET N=M(K)
340 LET P=1:LET T=1:GOSUB 800:LET W=1
350 IF T=1 THEN GOSUB 600:GOTO 400
360 FOR J=K TO 4:LET M(J)=M(J+1):NEXT J
370 IF X>1 THEN LET X=X-1:GOTO 330
380 LET M$="YOU HAVE WON":GOTO 950
400 LET M$=" YOUR TURN ":GOSUB 900
410 IF S=9 OR Q(9)<5 OR P(W)=FNC(S) OR
    P(W)=FNA(S) THEN GOTO 440
420 IF W<4 THEN LET W=W+1:GOTO 410
430 LET M$=" I HAVE WON ":GOTO 950
440 INPUT" WHICH PIECE ";N:GOSUB 3000
450 LET T=0:IF N<1 OR N>4 THEN GOTO 470
460 LET P=5:LET T=1:GOSUB 800
470 IF T=0 THEN LET M$=" TRY AGAIN ":
    GOSUB 900:GOTO 410
480 GOSUB 600:LET X=4:GOTO 250
500 DEF FNC(P)=P+1-8*INT(P/8)
510 DEF FNA(P)=P-1+8*INT((9-P)/8)
600 LET P(9)=P(N):LET Q(P(N))=9

```



Lines 440-480: Input your move, checking that it is legal.

Lines 500-510: Define functions which give the positions next to position P, clockwise and anti-clockwise.

Lines 600-740: Store and display the new positions of the pieces after each move.

Lines 800-840: Check that a move does not break the rules.

Lines 900-950: Print messages and end the game.

```

610 LET Q(S)=N:LET P(N)=S:LET S=P(9)
•620 FOR Z=1 TO 100:NEXT Z:BEEP 0.2,0
700 FOR J=1 TO 9:LET P$=CHR$(Q(J)+48)
•710 IF Q(J)>4 THEN P$=" ":INK 2
•720 IF Q(J)=9 THEN INK 6
730 LET R=R(J):LET C=C(J):GOSUB 4000:
PRINT P$
•740 INK 0:NEXT J:RETURN
800 LET L=P(N):IF L=9 THEN RETURN
810 LET J=P:IF FNC(L)=S OR FNA(L)=S
THEN RETURN
820 IF S=9 AND (P(J)=FNC(L) OR P(J)=
FNA(L)) THEN RETURN
830 IF J<P+3 THEN LET J=J+1:GOTO 820
840 LET T=0:RETURN
900 GOSUB 5000:PRINT M$:PRINT:RETURN
•950 GOSUB 900:INVERSE 0:STOP
    
```

## ★ BBC/ELECTRON

```

110 MODE 1:COLOUR 131:COLOUR 0
120 CLS:V=3:H=4:BL=18:GOSUB 2000
230 COLOUR 128:COLOUR 3:GOSUB 700:V=25:H=1
255 FOR J=1 TO 2000:NEXT J
335 IF X=1 THEN N=M(1)
620 FOR Z=1 TO 800:NEXT Z:SOUND 1,-15,53,4
710 IF Q(J)>4 THEN P$=" ":COLOUR 129
720 IF Q(J)=9 THEN COLOUR 130
740 COLOUR 128:NEXT J:RETURN
950 GOSUB 900:END
    
```

## ★ COMMODORE 64

```

110 POKE 53280,6:POKE 53281,15:POKE 646,0:LD=0
120 BL=20:GOSUB 2000:V=1:H=4:PRINT C$:PI=3.142
230 V=21:H=1:HI=20:DN=100:GOSUB 500:GOSUB 700
520 RETURN
620 FOR Z=1 TO 200:NEXT Z:GOSUB 8000
710 IF Q(J)>4 THEN P$=" ":POKE 646,2
720 PRINT CHR$(18);:IF Q(J)=9 THEN POKE 646,8
740 POKE 646,0:NEXT J:RETURN
950 GOSUB 900:PRINT CHR$(145);:END
    
```



## GAME NOTES

You are a detective hunting down a bank robber on the run and trying to recover the money he stole. At the start of the game the computer displays a forest surrounded by a hedge. The robber has just entered the forest from the West (left-hand side). As he blunders about among the trees he leaves a trail of footprints and scattered banknotes. After a while the robber may drop the bag of money in the hope of throwing you off his trail.

You will have to wait a few seconds while the computer gives the robber a ten-move start. Then begin tracking him by keying in the direction (N, E, S, W) you want to go, followed by a number for the distance you wish to move. After each move, any clues you find about the robber's trail will be shown on the screen:

N, S etc = footprints going

North, South etc

? = confused footprints going in several directions

£ = scattered banknotes

| = shows that the robber no longer has the money

\* = the stolen money

R = the robber

If the robber reaches the East (right) side of the hedge he may escape from the woods and get away.

The game continues until both the robber (R) and the money (\*) have been found. The object is to find both of them in as few moves as possible.

## SPECTRUM LISTING

```

100 REM ROBBER
•110 BORDER 0:PAPER 0:INK 4:DIM W$(315,1)
•120 DIM D$(4,1):LET U=0:LET P=3:CLS
130 DIM D(5):DIM R(300):DIM C(300)
140 GOSUB 2000:LET S=FNR(15)+U:LET E=S-U
150 READ D$(1),D$(2),D$(3),D$(4),G,T,X,M
160 DATA "N","S","E","W",0,0,B,3
170 LET B$="%%%%%%%%%"
180 LET MB=0:LET A=1:LET L=U:LET T$="£"
190 GOSUB 6000:PRINT " ";B$:FOR J=1 TO 15
200 PRINT TAB(P+2);"%";TAB(P+23);"%
210 NEXT J:PRINT TAB(P+2);B$:LET W=3.5
•220 INK 6:LET TD=20:LET V=8:LET H=6
230 LET D=P+3:LET R(1)=S:LET C(1)=H
240 FOR J=1 TO 315:LET W$(J)=" ":NEXT J
250 FOR J=1 TO 10:GOSUB 610:NEXT J
•260 GOSUB 5000:PRINT INVERSE 1;W$(8)
•270 INPUT"DIRECTION ";A$;" DISTANCE ";N
280 LET T=T+1:GOSUB 5000:PRINT W$(X)
290 IF A$="N" AND V-N>U THEN LET V=V-N
300 IF A$="W" AND H-N>P+2 THEN LET H=H-N
310 IF A$="S" AND V+N<U+16 THEN LET V=V+N
320 IF A$="E" AND H+N<P+23 THEN LET H=H+N
330 LET X=15*(H-P-3)+V-U:LET L=U+19
340 IF X<>Q OR G=2 THEN GOTO 370
350 LET W$(Q)="R":LET G=G+M:GOSUB 6000
360 PRINT"YOU HAVE CAUGHT THE ROBBER"
370 LET L=U+20:GOSUB 6000
380 IF X<>MB OR G=1 THEN GOTO 410
390 PRINT" YOU HAVE FOUND THE MONEY"
400 LET G=G+1:LET MB=0:LET W$(X)="*"
•410 GOSUB 5000:PRINT INVERSE 1;W$(X)
420 IF G<2 THEN GOSUB 600
430 IF G<3 THEN GOTO 270
440 IF M=3 THEN LET L=U+20:GOSUB 6000:
PRINT" WITH THE MONEY"
450 LET L=U+21:GOSUB 6000
460 PRINT" IN ";T;" TURNS"
470 FOR J=1 TO A:LET R=R(J)
480 LET C=C(J):LET N=15*(C-P-3)+R-U
•490 GOSUB 4000:PRINT INVERSE 1;W$(N)
500 FOR K=1 TO TD:NEXT K:NEXT J:STOP
600 IF ABS(S-V)+ABS(D-H)<3 THEN RETURN
610 FOR K=1 TO 4:LET D(K)=K:NEXT K
•620 LET F=0:LET N=FNR(W):LET B=D(N)
630 IF B=4 THEN LET W=W+0.5
640 IF B<>3 OR D-P<>22 THEN GOTO 670

```



```

650 LET G=G+M:LET L=U+19:GOSUB 6000
660 PRINT" THE ROBBER HAS ESCAPED"
670 IF (B=1 AND S=U+1) OR (B=2 AND S=U+
    15) OR (B=4 AND D=P+3) THEN GOTO 720
680 IF B<3 THEN LET R=S+2*B-3:LET C=D
690 IF B>2 THEN LET R=S:LET C=D-2*B+7
700 LET Q=15*(C-P-3)+R-U
710 IF W$(Q)=" " OR F=1 THEN GOTO 750
720 FOR K=N TO 4:LET D(K)=D(K+1):NEXT K
730 LET W=W-1:IF W>0 THEN GOTO 620
740 LET B=3:LET F=1:GOTO 690
750 LET P$=D$(B):LET S=R:LET D=C
760 LET W=3.5:IF FNR(3)<2 THEN LET P$=T$
770 IF W$(E)<>" " THEN LET P$="?"
780 IF M=3 AND FNR(30)=1 THEN LET MB=E:
    LET T$="+":LET M=2
790 LET A=A+1:LET W$(E)=P$:LET E=Q:IF E=
    MB THEN LET T$="£":LET M=3:LET MB=0
800 LET R(A)=S:LET C(A)=D:RETURN
    
```

## PROGRAM NOTES

Lines 100-260: Assign initial values to the variables and display your starting position in the wood.

Lines 270-330: Input your move and calculate your new position.

Lines 340-410: Decide whether you have found the robber or the money, then display your new position.

Line 420: Moves the robber, if necessary, using the subroutine at line 600.

Line 430: Jumps to line 270 for a new input if the game is not over.

Lines 440-500: Display the outcome of the game and the robber's route.

Lines 600-740: Move the robber in a randomly selected direction.

Lines 750-800: Determine whether or not the robber has the money.

## ★ BBC/ELECTRON

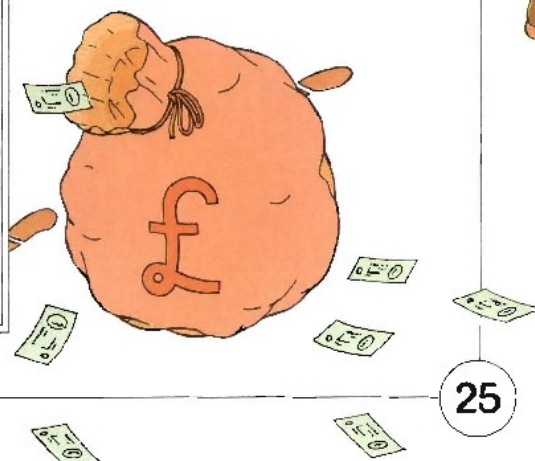
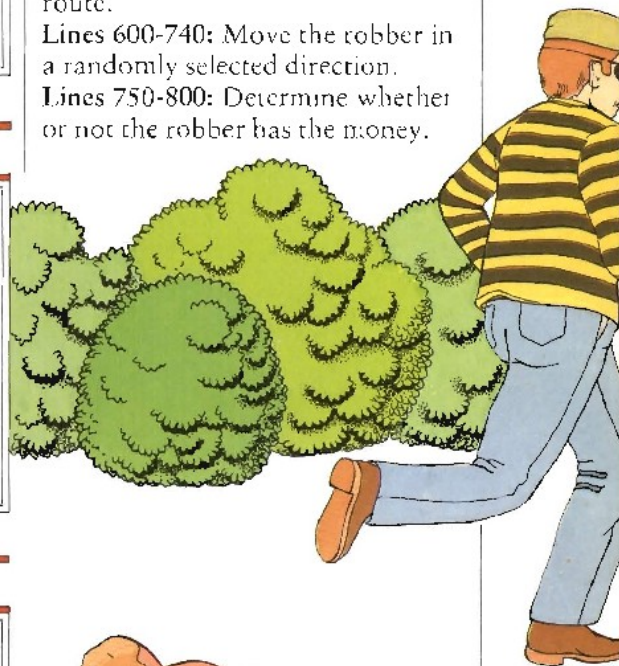
```

110 BL=39:IF PAGE=&E00 THEN MODE 1 ELSE MODE 4
120 VDU 19,1,2,0;:C.1:CLS:U=4:P=7:DIM D$(4)
220 V=12:H=10:TD=500:C.3:DIM W$(315)
260 C.131:C.0:PRINT TAB(H,V);W$(8):C.128:C.3
270 INPUT TAB(0,28)"DIRECTION "A$;TAB(18,28)
    "DISTANCE "N:GOSUB 3000
410 C.131:C.0:PRINT TAB(H,V);W$(X):C.128:C.3
490 C.131:C.0:PRINT TAB(C,R);W$(N);TAB(0,28)
620 F=0:N=INT(RND(1)*W)+1:B=D(N)
    
```

## ★ COMMODORE 64

```

110 POKE 53280,0:POKE 53281,0:POKE 646,5:BL=39
120 PRINT CHR$(147):TD=50:U=0:P=7:DIM D$(4)
220 POKE 646,7:DIM W$(315):V$=CHR$(18):V=8:H=10
260 GOSUB 5000:PRINT V$;W$(8)
270 PRINT H$;LEFT$(R$,23);"DIRECTION ";
275 INPUT A$:PRINT TAB(18);CHR$(145);
    "DISTANCE ";:INPUT N:GOSUB 3000
410 GOSUB 5000:PRINT V$;W$(X)
490 GOSUB 4000:PRINT V$;W$(N);H$;LEFT$(R$,20)
620 LET F=0:LET N=FNR(W):LET B=D(N)
    
```





## GAME NOTES

You are a gallant knight trying to rescue a helpless princess (or you may be a warrior maiden on a similar quest). To succeed, and win the game, you must reach the top of the steps leading to the sorcerer's tower.

Choose a level of difficulty from 1 (easy) to 20 (impossible), then as you set out to climb, the sorcerer will bombard you with magic symbols. To destroy each of these strange missiles you must press the matching key as quickly as possible.

Slowly you fight your way up the steps. Every time you are hit by a falling symbol, the strength of your enchanted sword is reduced and the game becomes more difficult. The symbols fall more quickly and drive you down the steps.

## SPECTRUM LISTING

```

100 REM SORCERER
110 GOSUB 1000:LET BL=0:GOSUB 2000
120 DATA 57,121,121,49,121,125,123,121
130 DATA 112,120,120,120,72,68,68,102,28
•140 BORDER 2:PAPER 6:INK 0:CLS
•150 DEF FNT(X)=PEEK 23672+256*PEEK 23673
160 LET L=5:LET P=1:GOSUB 6000:PRINT
    "SORCERER'S STEPS"
170 READ W:PRINT:INPUT "LEVEL (1-20) ";D
•180 CLS:FOR N=1 TO 704:PRINT PAPER 2;
    "#";:NEXT N
190 LET D=21-D:IF D<1 THEN LET D=1
200 FOR R=0 TO 21:IF R>8 THEN LET W=W-2
210 FOR C=0 TO W:GOSUB 4000:PRINT " "
220 NEXT C:NEXT R:LET S=0:LET H=2
•230 POKE 23672,0:POKE 23673,0:LET F=21
240 LET R=F-1:LET C=H:GOSUB 4000:PRINT U$
250 LET R=F:GOSUB 4000:PRINT Z$
260 GOSUB 6000:IF H=1 THEN GOTO 500

```

## ★ BBC/ELECTRON

```

140 MODE 1:VDU 23,1,0;
    0;0;0;
150 DEF FNT(X)=TIME/2
180 CLS:COLOUR 0:COLOUR
    129:FOR N=1 TO 1200
185 PRINT"#";:NEXT N:
    COLOUR 130
230 TIME=0:F=21
300 SOUND 1,-10,100-2*
    (R+F),D*0.4
305 FOR Z=1 TO D*40:
    NEXT Z
335 *FX 15,1
350 SOUND 1,-15,53,3:
    LET H=H-1

```

## ★ COMMODORE 64

```

140 POKE 53280,2:POKE
    53281,7
145 POKE 646,0:PRINT C$
150 DEF FNT(X)=
    INT(TI*5/6)
180 FOR N=0 TO 999:POKE
    55296+N,2
185 POKE 1024+N,163:
    NEXT N
230 TI$="000000":F=21
300 FOR Z=1 TO (D-1)*10
305 NEXT Z
350 HI=30-F:LO=0:DN=60:
    GOSUB 8000:H=H-1

```

## PROGRAM NOTES

Lines 100-150: Define the 'knight' graphics characters and set up screen colours.

Lines 160-170: Allow you to choose a level of difficulty.

Lines 180-230: Display the flight of steps and store initial values of the variables.

Lines 240-270: Print the knight and jump to the win/lose message routines if appropriate.

Lines 280-340: Move a randomly selected character down the screen and decide whether the correct key is being pressed.

Lines 350-360: Move the knight one step backwards and reduce the time delay in the character-moving routine.

Lines 400-430: Move the knight one step forward if the correct key



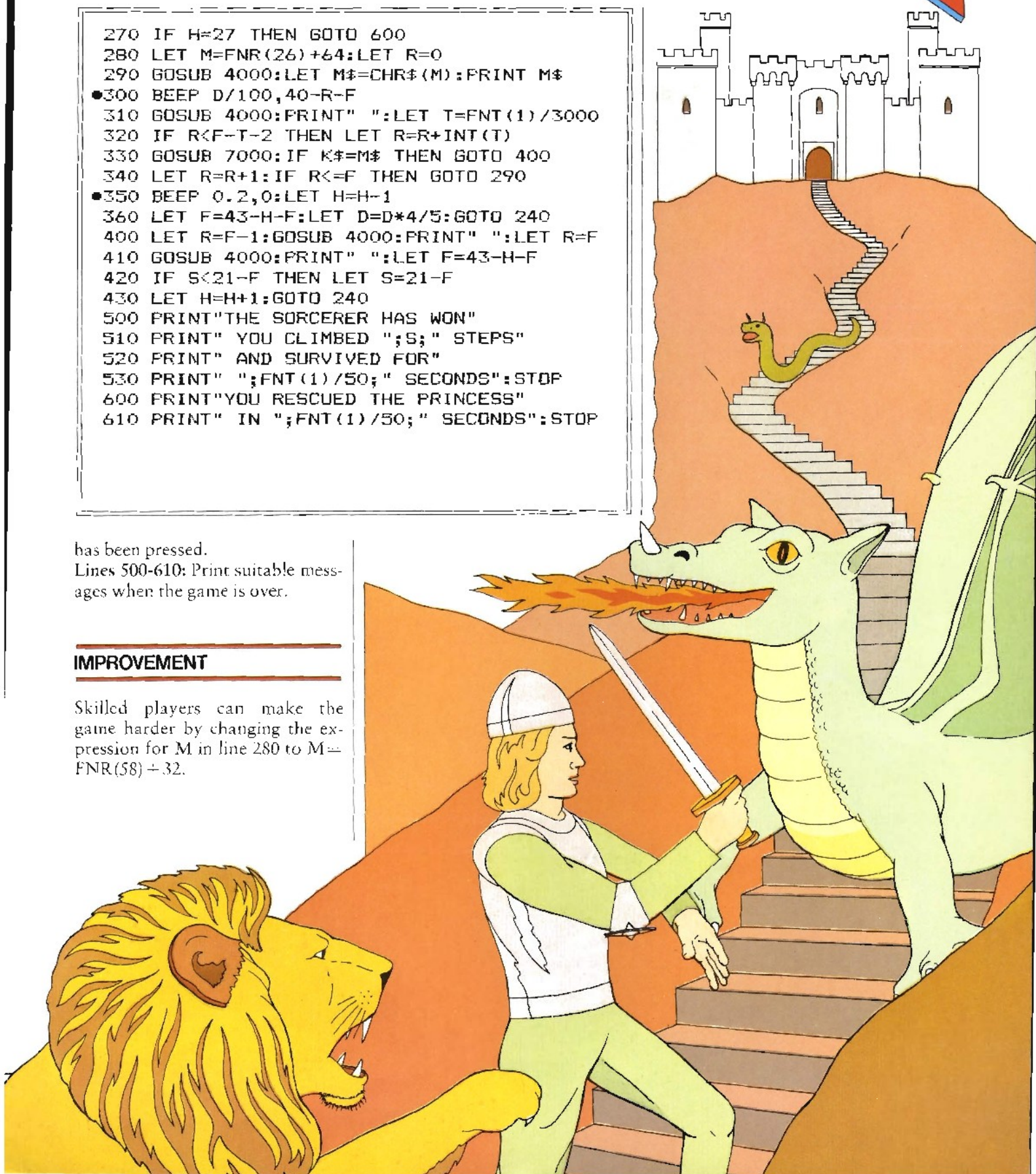
```

270 IF H=27 THEN GOTO 600
280 LET M=FNR(26)+64:LET R=0
290 GOSUB 4000:LET M$=CHR$(M):PRINT M$
•300 BEEP D/100,40-R-F
310 GOSUB 4000:PRINT " ":LET T=FNT(1)/3000
320 IF R<F-T-2 THEN LET R=R+INT(T)
330 GOSUB 7000:IF K$=M$ THEN GOTO 400
340 LET R=R+1:IF R<=F THEN GOTO 290
•350 BEEP 0.2,0:LET H=H-1
360 LET F=43-H-F:LET D=D*4/5:GOTO 240
400 LET R=F-1:GOSUB 4000:PRINT " ":LET R=F
410 GOSUB 4000:PRINT " ":LET F=43-H-F
420 IF S<21-F THEN LET S=21-F
430 LET H=H+1:GOTO 240
500 PRINT"THE SORCERER HAS WON"
510 PRINT" YOU CLIMBED ";S;" STEPS"
520 PRINT" AND SURVIVED FOR"
530 PRINT" ";FNT(1)/50;" SECONDS":STOP
600 PRINT"YOU RESCUED THE PRINCESS"
610 PRINT" IN ";FNT(1)/50;" SECONDS":STOP
    
```

has been pressed.  
Lines 500-610: Print suitable messages when the game is over.

## IMPROVEMENT

Skilled players can make the game harder by changing the expression for M in line 280 to  $M = FNR(58) + 32$ .





## GAME NOTES

Ten farms in a remote part of the country have been cut off by muddy floodwaters. You are the pilot of a small aircraft; your assignment is to drop a food parcel as close to each farm as possible.

Your aircraft keeps circling each farm until you press capital D to drop a parcel. The centre of the target area is marked by a cross.

You can score up to 400 points for a parcel dropped on target during the first flight pass over any farm. Later drops on target score up to 200 points. Your score will vary depending on how close you are to the target. No points are given for parcels which land outside the target area.

The "TOO LATE" message indicates that you pressed D too late and the parcel has landed off screen in floodwater.

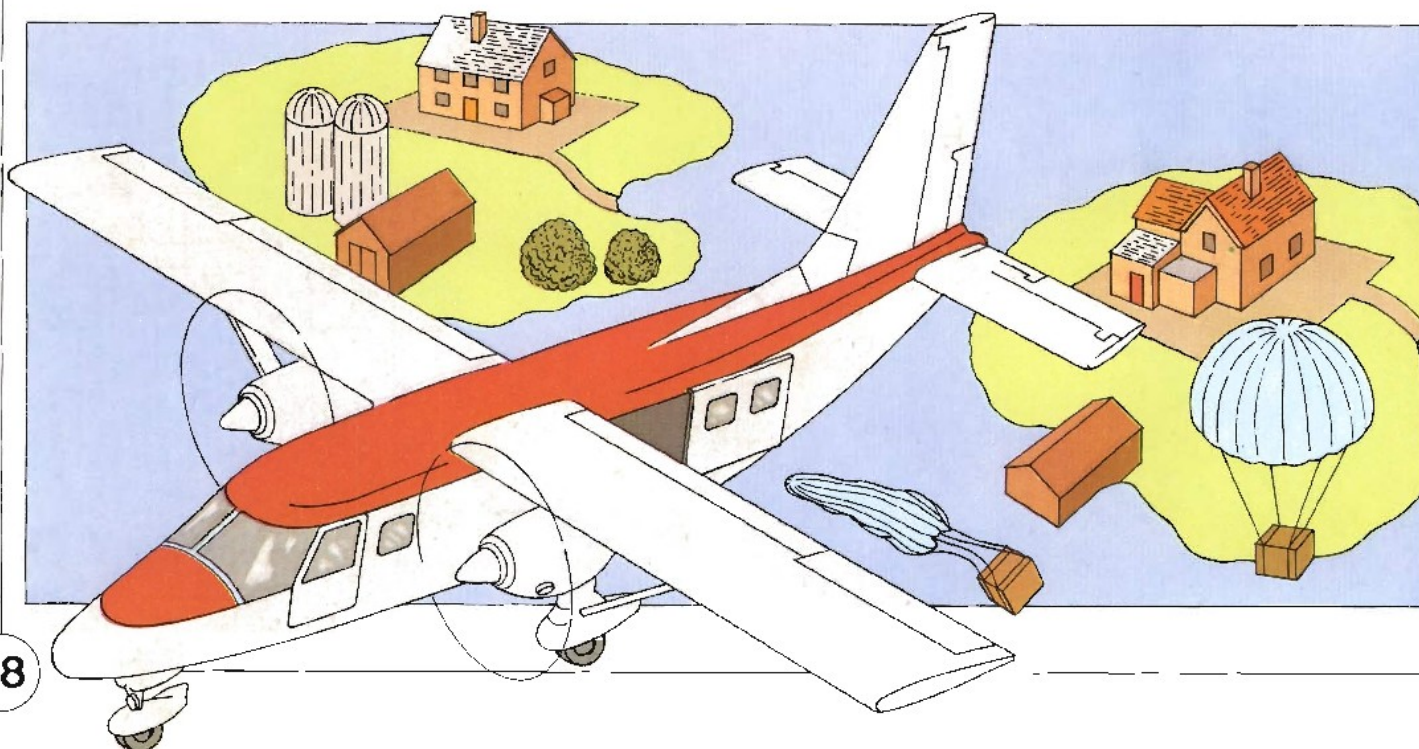
The object of the game is to gain the highest score after visiting ten farms – scores over 1500 are very good indeed.

## SPECTRUM LISTING

```

100 REM PARCELS
110 GOSUB 1000:LET BL=1:GOSUB 2000
120 DATA 0,0,0,63,127,63,1,0
130 DATA 0,2,6,254,255,224,192,0
•140 LET W=30:LET D=20:LET L=0:LET B=200
•150 BORDER 5:PAPER 5:INK 0:CLS
160 LET S=0:FOR T=1 TO 10:LET P=1:LET
    V=D+1:LET H=0:LET F=FNR(W-16)+3
170 GOSUB 6000:PRINT"FARM ";T:GOSUB 5000
•180 PAPER 6:FOR N=0 TO 31:PRINT" ";
190 NEXT N:LET H=F-3:GOSUB 5000
•200 PRINT"--<+>--":PAPER 5
210 LET K=2:LET R=FNR(7)+L:LET C=W+1
220 GOSUB 500:GOSUB 7000
230 IF K<>"D" THEN GOTO 220
240 LET V=R+1:LET H=C+1:LET Z=0
250 GOSUB 5000:PRINT"#":GOSUB 500
260 LET NV=V+(2*Z+1)/5:LET Z=Z+1
270 LET NH=H-1:IF NV>D THEN LET NH=H-
    INT((D-V)/(NV-V)+0.5):LET NV=D
280 GOSUB 5000:PRINT" ":LET V=INT(NV)
290 LET H=NH:GOSUB 5000:PRINT"#":GOSUB
    500
300 IF V<D AND H>0 THEN GOTO 260

```





## PROGRAM NOTES

Lines 100-150: Define the 'aircraft' graphics characters, initialise variables and set up screen colours.

Lines 160-200: Print the ground and a randomly selected target area.

Lines 210-230: Move the aircraft across the screen until the key D is pressed.

Lines 240-310: Move the aircraft and the parcel until the parcel lands or reaches the edge of the screen.

Lines 320-360: Calculate and print the score.

Lines 370-400: Produce a short pause then either send the computer back to line 160 for the next farm or display the final score.

Lines 500-520: A subroutine to move the plane forward by one character square.

```

310 IF V<D THEN LET Q=0:GOSUB 5000:PRINT
    "TOO LATE":GOTO 350
320 IF ABS(H-F)>3 THEN LET Q=0:GOTO 350
330 LET M=ABS(H-F):LET Q=110-5*M*(M+1)
•340 IF M=0 THEN BEEP 0.2,0:LET Q=Q+90
350 IF C>0 THEN GOSUB 500:GOTO 350
360 LET S=S+Q*K:LET P=15:GOSUB 6000:PRINT
    "SCORE SO FAR ";S:LET P=1
370 GOSUB 4000:PRINT"  "
380 FOR J=1 TO B:NEXT J
390 GOSUB 5000:PRINT"      ":NEXT T
400 GOSUB 6000:PRINT"  YOUR TOTAL SCORE
    IS ";S;"  ":PRINT:STOP
500 GOSUB 4000:PRINT"  ":LET C=C-1
510 IF C<0 THEN LET C=W:LET K=1
•520 GOSUB 4000:PRINT INK 2;U$;Z$:RETURN
    
```

## ★ BBC/ELECTRON

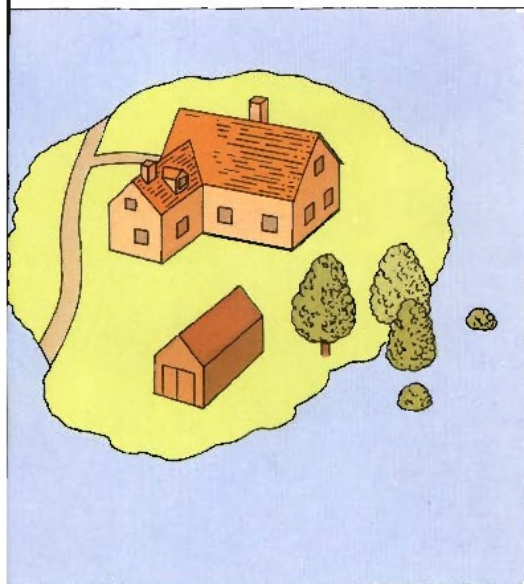
```

140 W=38:D=26:L=4:B=1500
150 MODE 1:VDU 19,3,6;0;24,0;0;1279;159;
155 COLOUR 131:COLOUR 0:CLS:GCOL 0,130:CLG:
    VDU 23,1,0;0;0;0;
180 COLOUR 130:FOR N=0 TO 39:PRINT" ";
200 PRINT"--<+>--":COLOUR 131
340 IF M=0 THEN SOUND 1,-15,53,4:LET Q=Q+90
520 COLOUR 1:PRINT TAB(C,R);U$;Z$:FOR
    J=1 TO 40:NEXT J:COLOUR 0:RETURN
    
```

## ★ COMMODORE 64

```

140 W=38:D=22:L=1:B=200:DN=80:PRINT C$
150 POKE 53280,14:POKE 53281,14:POKE 646,0
180 HI=20:LQ=0:PRINT CHR$(149);CHR$(18);:
    FOR N=0 TO 38:PRINT" ";
200 PRINT"--<+>--"
340 IF M=0 THEN GOSUB 8000:Q=Q+90
520 POKE 646,2:GOSUB 4000:PRINT U$;Z$:
    POKE 646,0:RETURN
    
```





## GAME NOTES

A clever forger is at work. He is producing counterfeit copies of top-grade copper ingots. The forgeries contain high proportions of cheaper metals such as lead, iron and zinc.

You have just been shipped a consignment of copper ingots which includes one forgery. You can detect it because it will be either heavier or lighter than the other ingots. The aim of the game is to find the forgery with as few weighings as possible on your pair of scales.

Choose a level of difficulty from 1 (easy) to 8 (difficult). The computer will then display up to 50 copper bars.

You now enter the number of bars you wish to weigh. If there are 8 bars in all, for instance, when you see the message "IN LEFT PAN?", you might place the first 4 bars in the left hand pan of the scales by entering 1, and then entering 4. The next 4 bars you would place in the right-hand pan by entering 5, and then 8. To place just 1 bar into a pan, enter the number of the bar (number 8 for example) both times.

The computer weighs the ingots and shows you the result. Keep weighing the ingots until you think you know which ingot is a forgery; then enter N when the computer asks "WOULD YOU LIKE ANOTHER WEIGHING?".

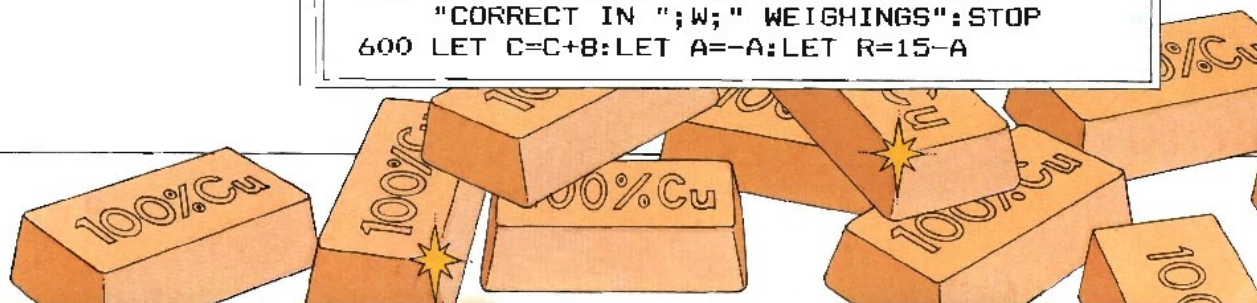
You should then key in the number of the forgery and H if you think it is heavier or L if you think it is lighter. The computer will reveal how accurate your detective work has been.

## SPECTRUM LISTING

```

100 REM FORGERY
110 GOSUB 1000:LET BL=39:GOSUB 2000
120 DATA 255,255,255,255,255,255,255,
    255,24,24,24,60,60,60,126,126
•130 BORDER 0:PAPER 7:INK 0:CLS:LET Z=1
•140 DIM B$(50,2):LET TD=200:LET C=15
150 INPUT"LEVEL (1 TO 8) ";L:GOSUB 3000
160 LET N=6*L-4+FNR(6):DIM W(N):LET W=0
170 LET L=16:LET F=FNR(N):LET G=FNR(2)
180 LET W(F)=2*G-3:LET W$=CHR$(80-4*G)
190 LET A=0:LET P=L:GOSUB 6000:PRINT"*"
200 PRINT TAB(P);Z$:PRINT TAB(P);U$
210 PRINT TAB(P);U$:PRINT TAB(C);U$:U$:U$
220 LET B=0:LET C=1:GOSUB 600:GOSUB 600
230 FOR J=1 TO N:LET V=3*INT((J-1)/10)
240 LET H=3*J-10*V-Z:LET B$(J)=U$+U$
250 GOSUB 5000:PRINT STR$(J):NEXT J
260 GOSUB 700:LET S=0:LET D=0:LET W=W+1
•270 INPUT"IN LEFT PAN ";X;" TO ";Y
280 IF X<1 OR X>N THEN GOTO 270
290 LET B=1:LET C=1:FOR J=X TO Y
300 IF B$(J)<>U$+U$ THEN GOTO 320
310 LET S=S+W(J):LET B$(J)="LP"
320 NEXT J:GOSUB 700:GOSUB 600
330 INPUT"MORE IN LEFT PAN (Y/N)";A$
340 GOSUB 3000:IF A$="Y" THEN GOTO 270
•350 INPUT"IN RIGHT PAN ";X;" TO ";Y
360 IF X<1 OR X>N THEN GOTO 350
370 LET C=9:FOR J=X TO Y
380 IF B$(J)<>U$+U$ THEN GOTO 400
390 LET D=D+W(J):LET B$(J)="RP"
400 NEXT J:GOSUB 700:GOSUB 600:LET C=1
410 INPUT"MORE IN RIGHT PAN (Y/N)";A$
420 GOSUB 3000:IF A$="Y" THEN GOTO 350
430 LET A=SGN(S-D):GOSUB 600:GOSUB 600
440 INPUT"WOULD YOU LIKE ANOTHER
    WEIGHING ";A$:GOSUB 3000
450 LET A=0:IF A$="Y" THEN GOTO 220
460 INPUT"NUMBER OF FORGERY ";G
470 GOSUB 3000:INPUT" IS IT HEAVIER OR
    LIGHTER (H/L) ";A$:GOSUB 3000
480 IF G=F AND A$=W$ THEN GOTO 510
•490 PRINT#1;"          SORRY-NOT RIGHT"
500 FOR J=1 TO TD:NEXT J:GOTO 220
510 LET R=21:LET C=5:GOSUB 4000:PRINT
    "CORRECT IN ";W;" WEIGHINGS":STOP
600 LET C=C+8:LET A=-A:LET R=15-A

```





```

610 GOSUB 4000:PRINT"      ":LET R=R+1
620 GOSUB 4000:PRINT"*";U$;"      ";U$;"*"
•630 LET R=R+1:IF B=1 THEN PRINT INK 2;AT
    R-1,C+2;U$;U$;U$
640 GOSUB 4000:PRINT"  ";U$;U$;U$;U$;U$;"  "
650 LET R=R+1:GOSUB 4000:PRINT"      "
•660 RETURN
700 FOR J=1 TO N:LET V=3*INT((J-1)/10)+1
710 LET H=3*J-10*V+9:GOSUB 5000
•720 PRINT INK 2;B$(J):NEXT J:RETURN
    
```

## PROGRAM NOTES

Lines 100-150: Define two graphics characters, set display colours and input level of difficulty.

Lines 160-180: Select random numbers which determine the number of ingots, the location of the forged ingot and its weight.

Lines 190-260: Print the scales with empty pans and then the set of copper ingots.

Lines 270-340: Input the numbers of the bars to be placed in the left-hand pan.

Lines 350-420: Input the numbers of the bars to be placed in the right-hand pan.

Lines 430-450: Display the result of the weighing and jump back to line 220 if a further weighing is required.

Lines 460-510: Input your answers to the problem and print suitable messages.

Lines 600-660: Subroutine to print the copper bars and their position on the scales.

### ★ BBC/ELECTRON

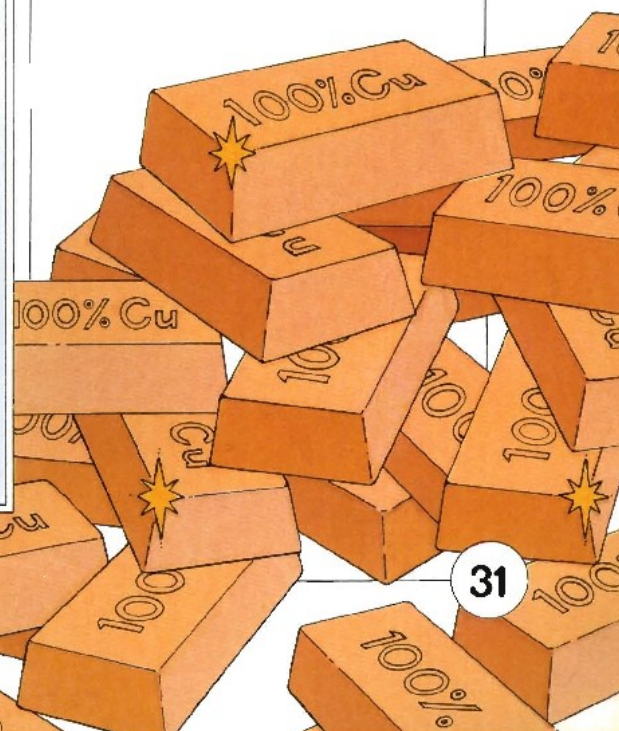
```

130 MODE 1:COLOUR 131:
    COLOUR 0:CLS
135 Z=1:C=15:TD=2000
140 DIM B$(50):PRINT
    TAB(0,25);
270 PRINT TAB(0,25)
272 GOSUB 3000:INPUT
    "IN LEFT PAN "X
274 INPUT TAB(15,25)
    "TO "Y:GOSUB 3000
350 INPUT TAB(0,25)
    "IN RIGHT PAN "X
355 INPUT TAB(16,25)
    "TO "Y:GOSUB 3000
490 PRINT TAB(9,25);
    "SORRY-NOT RIGHT"
630 LET R=R+1:IF B=1
    THEN GOSUB 800
660 PRINT TAB(0,25);:
    RETURN
720 COLOUR 1:PRINT
    B$(J):COLOUR 0
730 NEXT J:RETURN
800 COLOUR 1
810 PRINT TAB(C+2,
    R-1);U$;U$;U$
820 COLOUR 0:RETURN
    
```

### ★ COMMODORE 64

```

130 POKE 53280,0:POKE
    53281,12:POKE 646,0
135 C=15:TD=400:Z=2:
    DIM B$(50)
140 PRINT C;
    LEFT$(R$,23);
270 L=23:P=0:GOSUB 6000
272 PRINT:GOSUB 3000
274 INPUT"IN LEFT
    PAN ";X
276 PRINT CHR$(145);
    TAB(17);"TO ";
278 INPUT Y:GOSUB 3000
350 GOSUB 6000:INPUT"IN
    RIGHT PAN ";X
352 PRINT CHR$(145);
    TAB(18);"TO ";
354 INPUT Y:GOSUB 3000
490 PRINT TAB(9);"SORRY
    -NOT RIGHT"
630 LET R=R+1:IF B=1 TH
    EN PRINT CHR$(28);
    TAB(C+2);CHR$(145);
    U$;U$;U$;CHR$(144)
660 GOSUB 6000:RETURN
720 PRINT CHR$(28);
    B$(J);CHR$(144)
730 NEXT J:RETURN
    
```





## SUBROUTINE NOTES

The subroutines numbered 1000, 2000, 3000 and on up to 8000 contain instructions that are used again and again throughout the book. They should be typed in at the end of the main program.

Recording the subroutines on another cassette makes a lot of sense as it saves you a great deal of time typing. *Before you start to enter a new program, load the subroutines first and then type in the main listing; otherwise it is not possible to merge the two.*

## LINE NOTES

Lines 1000-1100: Set up two specially designed graphics characters and store them as variables U\$ and Z\$.

Lines 2000-2030: Define a random number and on the Commodore 64 set up the variables used to position the cursor.

Lines 3000-3010: Clear the input line on the screen so it is ready for a new input.

Lines 4000, 5000, 6000: Move the cursor to a given position.

Line 7000: Stores any key you press as variable K\$.

Lines 8000-8040: Produce a short sound for the Commodore 64 only. The variables H\$ and L\$ determine the pitch of the note and variable DN the length of the note.

## SUBROUTINE LISTINGS

### SPECTRUM:

```
1000 FOR N=0 TO 15:READ D:POKE USR"A"+N,D
1010 NEXT N:LET U$=CHR$144:LET Z$=CHR$145
1020 RETURN
2000 DEF FN R(N)=INT(RND*N)+1
2010 RANDOMIZE:RETURN
3000 RETURN
4000 PRINT AT R,C:;RETURN
5000 PRINT AT V,H:;RETURN
6000 PRINT AT L,P:;RETURN
7000 LET K$=INKEY$:RETURN
```

### BBC/ELECTRON:

```
1000 FOR J=0 TO 1:VDU 23,224+J
1010 FOR N=1 TO 8:READ D:VDU D:NEXT N
1020 NEXT J:U$=CHR$224:Z$=CHR$225:RETURN
2000 DEF FNR(N)=RND(N)
2010 RETURN
3000 PRINT CHR$11;SPC(BL);CHR$13:;RETURN
4000 PRINT TAB(C,R):;RETURN
5000 PRINT TAB(H,V):;RETURN
6000 PRINT TAB(P,L):;RETURN
7000 K$=INKEY$(0):RETURN
```

### COMMODORE 64:

```
1000 PRINT:IF PEEK(53272)=29 THEN 1090
1010 PRINT"PLEASE WAIT FOR A MOMENT"
1020 POKE 53272,28:POKE 52,48:POKE 56,48
1030 CLR:POKE 56334,PEEK(56334) AND 254
1040 POKE 1,PEEK(1) AND 251
1050 FOR N=0 TO 1535
1060 POKE 12288+N,PEEK(53248+N):NEXT N
1070 POKE 1,PEEK(1) OR 4
1080 POKE 56334,PEEK(56334) OR 1:GOTO 110
1090 FOR N=0 TO 15:READ D:POKE 12584+N,D
1100 NEXT N:U$="Z":Z$="&":RETURN
2000 DEF FNR(N)=INT(RND(0)*N)+1
2010 FOR N=1 TO 24:R$=R$+CHR$(17):NEXT N
2020 FOR N=1 TO BL:BL$=BL$+" ":NEXT N
2030 C$=CHR$(147):H$=CHR$(19):RETURN
3000 PRINT CHR$(145);BL$:CHR$(145);
3010 PRINT CHR$(13):;RETURN
4000 PRINT H$;TAB(C);LEFT$(R$,R):;RETURN
5000 PRINT H$;TAB(H);LEFT$(R$,V):;RETURN
6000 PRINT H$;TAB(P);LEFT$(R$,L):;RETURN
7000 GET K$:RETURN
8000 POKE 54272,L0:POKE 54273,H1
8010 POKE 54277,0:POKE 54278,240
8020 POKE 54296,15:POKE 54276,33
8030 FOR TM=1 TO DN:NEXT TM
8040 POKE 54276,32:RETURN
```







# Micromasters Computer Games

gives line-by-line listings for keying in 14 games for the Spectrum, BBC, Electron and Commodore 64. The games vary from simple blocking games to brain-teasers, games of strategy and keyboard trainers; levels of difficulty range from simple to near impossible! The listings are quick to key in, and subroutines for colour and sound are included.

*Author Patricia Grady teaches computer studies and has taught mathematics from junior to diploma level. She has published books on mathematics, games and puzzles.*

