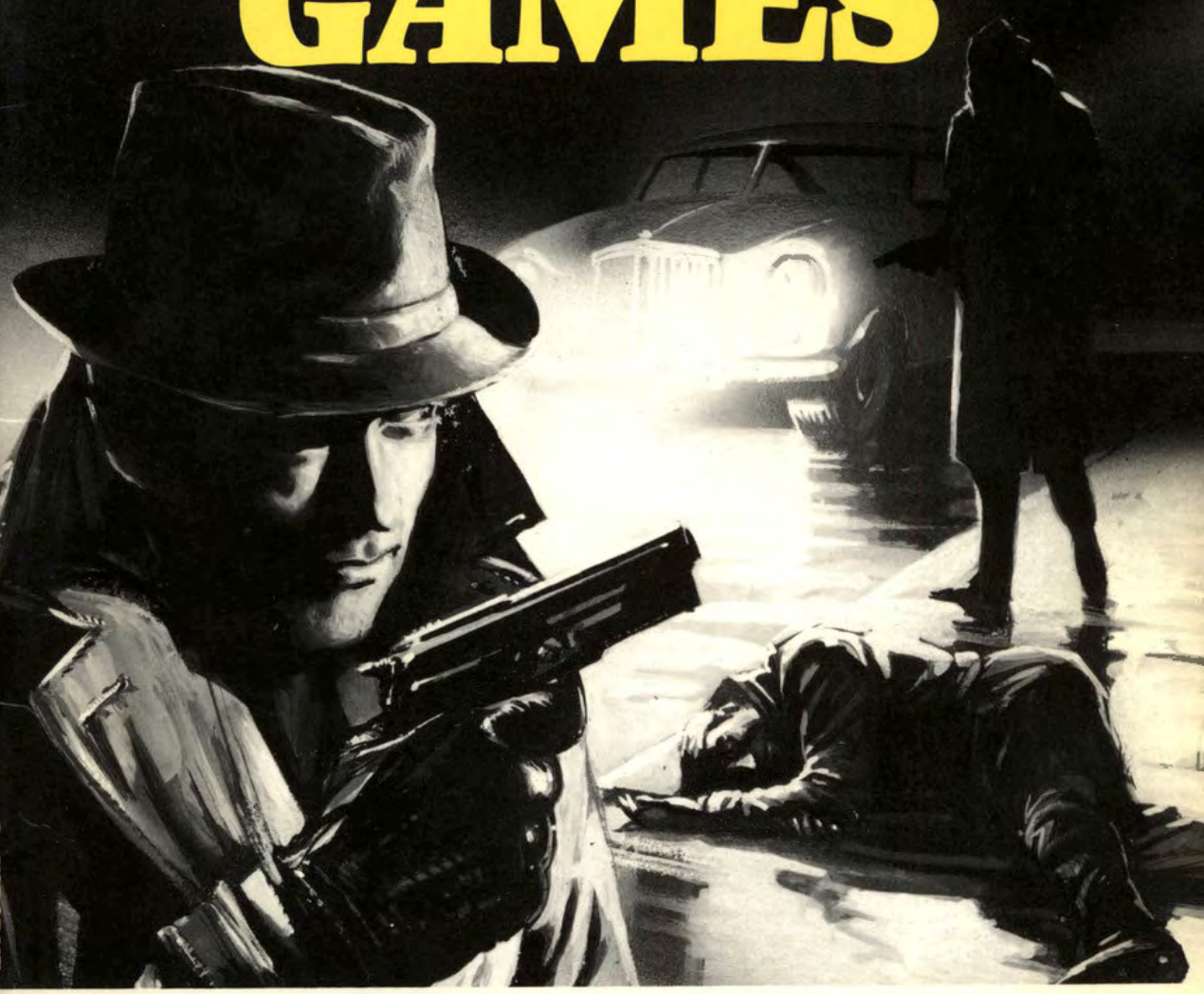
USBORNE

COMPUTER SPY GAMES

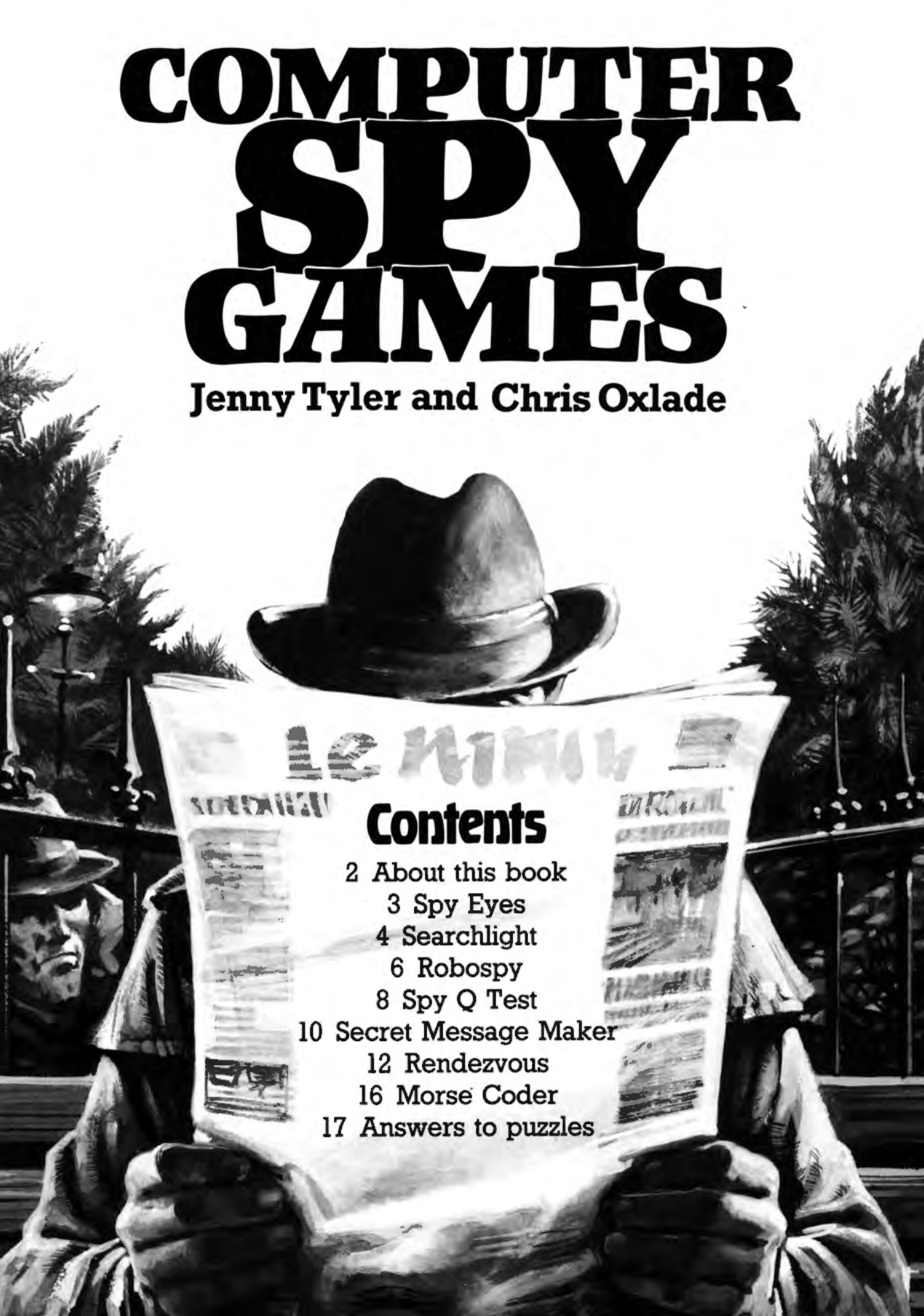


.....FOR...COMMODORE 64...VIC 20...APPLE....

...TRS 80 32K ...BBC...ELECTRON...SPECTRUM...

COMPUTER SPY GAMES

Jenny Tyler and Chris Oxlade



Contents

2 About this book
3 Spy Eyes
4 Searchlight
6 Robospy
8 Spy Q Test
10 Secret Message Maker
12 Rendezvous
16 Morse Coder
17 Answers to puzzles

About this book

The programs in this book are written in a standard version of BASIC and there are conversion lines to type in for most of the main types of home computers. Look down the left-hand side of the program for the symbol for your computer and then look at the list of changes for the correct version of that line. The symbols for the various computers are as follows:

▲ Commodore 64 and VIC 20
★ BBC and Electron
✡ Spectrum
● Apple
■ TRS-80 (extended BASIC version)

About the games

The games in this book are very simple. They are intended to help you get used to your computer and to the BASIC language by typing in listings, debugging them and seeing how they work. The programs do not contain graphics or sound as these vary so much from computer to computer, but you can try adding these.

You can change and adapt the games as much as you like. There are suggestions for ways of doing this next to each program and you can experiment with your own ideas as well. This way you can use the games in this book as a basis for longer, more complicated games of your own.

Typing and running the programs

Remember, even short programs can be quite difficult and time-consuming to type in correctly. Check each line as you go. It is so easy to make mistakes, even if you are quite experienced. When you have typed in the whole listing, check it again, making sure you haven't missed any lines, spaces or punctuation.

To start the game, type RUN. Read the introduction to the game first so that you have some idea of what you are supposed to do before you start. If the program doesn't work properly, it is quite likely that there is a mistake in it somewhere, so LIST the program and check again.

When the game is over, the computer may ask if you want to play again or say something like BREAK in 200, in which case you must type RUN to play again.

Changing the speed

Some games depend on the speed of both your reactions and your computer. You may find you need to adjust the speed. You will find instructions for doing this next to the program listing.

***Robospy* was written by Adrian Hall.**

Illustrated by Sue Walliker, Rob McCaig and Martin Newton.

Designed by Roger Priddy

First published in 1984 by Usborne Publishing Ltd, 20 Garrick Street, London WC2E 9BJ.

Spy Eyes

If you think you're a good spy, try this.

The computer will print the numbers 1 to 9 on your screen. Watch them like a hawk while you press a key (any one will do). One of them moves, but which? When you think you know, press a key again and tell the computer.

Bet your powers of observation aren't as good as you thought.

```
 10 DIM X(9):DIM Y(9)
 20 LET P=0
 30 FOR I=1 TO 9
 40 GOSUB 340:LET X(I)=N+3
 50 GOSUB 340:LET Y(I)=N+3
 60 NEXT I
 70 GOSUB 360
 80 GOSUB 310
 90 GOSUB 340
100 LET M=N:GOSUB 340
110 LET X(M)=X(M)+SGN(N-5.1)
120 GOSUB 360
130 GOSUB 310
●▲140 CLS:PRINT
150 PRINT "WHICH NUMBER MOVED"
160 INPUT A
170 IF A<>M THEN GOTO 250
●▲180 CLS:PRINT
190 PRINT "WELL SPIED!"
200 LET P=P+1
210 PRINT "YOU NOW HAVE ";P;" POINTS"
220 PRINT:PRINT "PRESS A KEY"
230 GOSUB 310
240 GOTO 30
●▲250 CLS:PRINT:PRINT "WRONG - END OF GO"
260 PRINT "CORRECT ANSWER WAS ";M
270 PRINT "YOU SCORED ";P;" POINTS"
280 PRINT "ANOTHER GO? (Y/N)"
290 INPUT A$:IF A$="Y" THEN RUN
300 STOP
★●▲310 LET I$=INKEY$
320 IF I$="" THEN GOTO 310
330 RETURN
✡■340 LET N=INT(RND(1)*9)+1
350 RETURN
●▲360 CLS
370 FOR I=1 TO 9
✡■●▲380 PRINT TAB(X(I),Y(I));STR$(I)
390 NEXT I
400 RETURN
```

See if you can work out a high-score routine for this game.

How the program works

10: Sets up storage space for positions of numbers.

20: Sets variable for score to zero.

30-60: Choose positions for the numbers.

70: Goes to subroutine to print the numbers.

80: Goes to subroutine to wait for key press.

90-100: Go to subroutine to choose one of the numbers and call it M.

110: Chooses left or right move.

120: Goes to subroutine which prints numbers.

130: Goes to subroutine to wait for key press.

140-150: Clears screen. Prints message.

160: Gets number from you.

170: If you are wrong, program jumps to message at line 260.

180-190: Clears screen and prints message.

200-210: Increase score and print it out.

220-230: Print message and go to subroutine which waits for key press.

240: Goes back for next go.

250-300: Print messages if you were wrong and ask if you want another go.

310-330: Subroutine to wait for key press.

340-350: Subroutine to choose random number between 1 and 9.

360-400: Subroutine to print numbers at the chosen positions on the screen.

Conversion lines

```
● 140,180,250,360 Replace CLS with HOME
▲ 140,180,250,360 Replace CLS with PRINT CHR$(147)
★ 310 LET I$=INKEY$(0)
● 310 I$="":IF PEEK(-16384)>127 THEN GET I$
▲ 310 GET I$
✡ 340 Replace RND(1) with RND
■ 340 LET N=RND(9)
✡ 380 PRINT AT Y(I),X(I);STR$(I)
● 380 VTAB(Y(I)):HTAB(X(I)):PRINT STR$(I)
▲ 380 PRINT CHR$(19):FOR LL=1 TO Y(I):PRINT:NEXT:
     PRINT TAB(X(I));STR$(I)
■ 380 PRINT@ Y(I)*32+X(I),STR$(I);
```

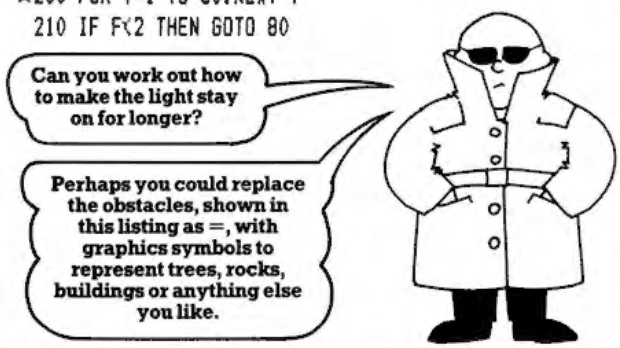
Searchlight

A mission most secret and desperately dangerous must be undertaken this very night – by you.

You must cross a closely watched section of enemy territory and return, avoiding their gigantic and very powerful searchlight. There are rocks, bushes and other obstacles to hide behind, but there are no second chances – once they've seen you, you've had it. When you have completed one mission successfully, there is another, even more difficult, to undertake. Keep going, we are all depending on your success.

Use key M to move right and N to move left. To complete one mission you must go right across from left to right and back again.

```
   10 GOSUB 450
   20 LET A=1:LET G=0:LET S=0
●▲30 CLS
   40 LET X=0:LET Y=12:LET B$=A$(A)
   50 GOSUB 380
   60 LET F=0:LET N=0:LET NN=0:GOSUB 340
   70 LET L=0:LET C=0:LET TC=10:LET C1=0
★●▲80 LET I$=INKEY$
   90 IF I$="N" THEN LET NN=N-1
  100 IF I$="M" THEN LET NN=N+1
  110 IF NN>19 THEN LET NN=19
  120 IF NN<0 THEN LET NN=0
  130 IF NN=19 AND F=0 THEN LET F=1
  140 IF NN=0 AND F=1 THEN LET F=2
  150 GOSUB 340
  160 IF N<>NN THEN LET S=S+1
  170 LET N=NN:LET G=G+1
  180 GOSUB 400
✡190 IF MID$(A$(A),N+1,1)=" " AND L=1 THEN GOTO 240
★200 FOR T=1 TO 50:NEXT T
  210 IF F<2 THEN GOTO 80
```



How the program works

10: Goes to subroutine which reads in data.

20: Sets up variables.

40-50: Print obstacles.

60-70: N is player's position. NN is new position.

80: Looks for key press.

90: If key press is N (left), new position is N minus 1. If it is M (right), new position is N plus 1.

100-120: Stop player moving off ends.

130-140: F is set to 1 when player reaches right-hand side and 2 when left-hand side is reached again.

150: Goes to subroutine to move player.

160: If player has moved, S is increased.

170: Lets player position equal new position and increases G by 1. G acts as a timer, increasing all the time whether or not the player moves.

180: Goes to subroutine which turns light on and off.

190: Checks if player is below a space when the light is on and if so goes to losing message.

200: Pause

210: If player hasn't returned to left-hand side, then goes back to look for another key press.

```
220 LET A=A+1:IF A=8 THEN LET A=7
230 GOTO 30
240 LET X=4:LET Y=1:LET B$="YOU HAVE BEEN SEEN"
250 GOSUB 380:PRINT
260 PRINT "YOU SCORED ";INT((A-1+S/6)*100)
270 PRINT:PRINT "ANOTHER GO? (Y/N)"
280 INPUT C$:IF C$="Y" THEN RUN
290 PRINT "BYE.....":STOP
300 LET Y=3:LET X=10:LET B$="*"
310 GOSUB 380:RETURN
320 LET X=10:LET Y=3:LET B$=" "
330 GOSUB 380:RETURN
340 LET X=N:LET Y=13:LET B$=" "
350 GOSUB 380
360 LET X=NN:LET B$="S"
370 GOSUB 380:RETURN
■●▲380 PRINT TAB(X,Y);B$
390 RETURN
400 IF L=1 THEN LET C=C+1
✡■410 IF C=TC THEN LET L=0:LET C=0:LET TC=INT(RND(1)*8+(12-A)):GOSUB 320
420 IF L=0 THEN LET C1=C1+1
✡■430 IF C1=TC THEN LET L=1:LET C1=0:LET TC=INT(RND(1)*10+(8-A)):GOSUB 300
440 RETURN
✡450 DIM A$(7)
460 FOR I=1 TO 7:READ A$(I):NEXT I
470 RETURN
480 DATA "== = = ==  == = = =="
490 DATA "==  = ==  == =  == ="
500 DATA "=  ==  = = =  =  ="
510 DATA "=  =  =  =  ==  ="
520 DATA "=  =  = = =  =  ="
530 DATA "=  =  =  =  = ="
540 DATA "=  =  =  =  ="
```

Add an alarm sound for when player is seen.

220: Increases level of difficulty.

230: Goes back to print new obstacles.

240-290: Print losing message and score and ask if player wants another go.

300-310: Print searchlight.

320-330: Turn searchlight off.

340-370: Subroutine which alters player's position.

380-390: Subroutine which prints at position X,Y on screen.

400-440: Subroutine which determines length of time light is on and off.

450-470: Subroutine which reads in data.

480-540: Data lines.

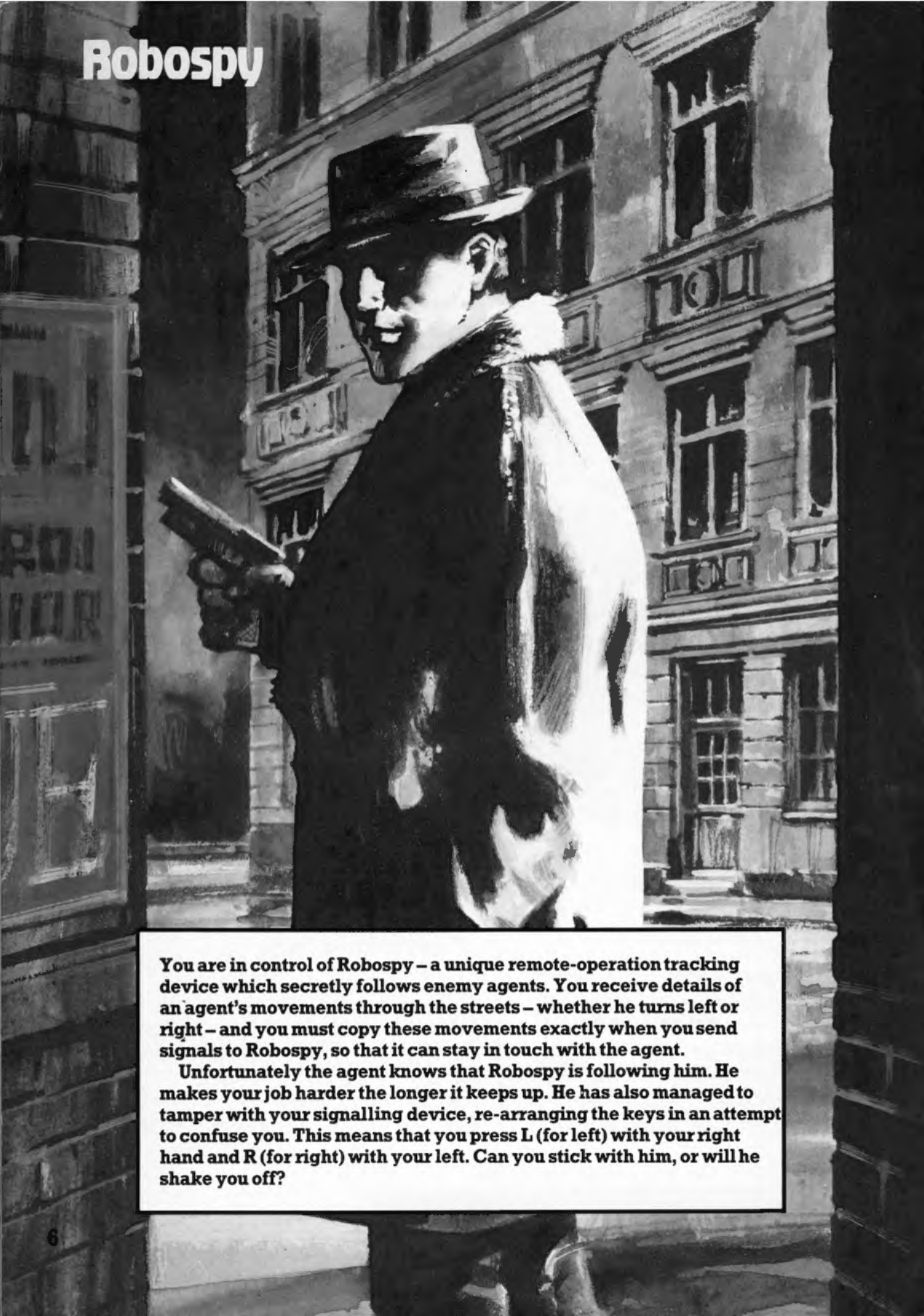
You could make the game easier by giving the player a second chance. Can you think how to do this?

You can change the positions of the obstacles by changing the data lines.

Conversion lines

●30 HOME
▲30 PRINT CHR$(147)
★80 LET I$=INKEY$(0)
▲80 GET I$
●80 I$="":IF PEEK(-16384)>127 THEN GET I$
✡190 IF A$(A,N+1)=" " AND L=1 THEN GOTO 240
✡380 PRINT AT Y,X;B$
●380 VTAB(Y):HTAB(X+1):PRINT B$
▲380 PRINT CHR$(19):FOR LL=1 TO Y:PRINT:NEXT:PRINT TAB(X);B$
■380 PRINT@ Y*32+X,B$;
✡410,430 Replace RND(1) with RND
■410,430 Replace RND(1) with RND(0)
✡450 DIM A$(7,20)
★Speed is controlled by line 200
For BBC and Electron change to FOR T=1 TO 150:NEXT T

Robospy



You are in control of Robospy – a unique remote-operation tracking device which secretly follows enemy agents. You receive details of an agent's movements through the streets – whether he turns left or right – and you must copy these movements exactly when you send signals to Robospy, so that it can stay in touch with the agent.

Unfortunately the agent knows that Robospy is following him. He makes your job harder the longer it keeps up. He has also managed to tamper with your signalling device, re-arranging the keys in an attempt to confuse you. This means that you press L (for left) with your right hand and R (for right) with your left. Can you stick with him, or will he shake you off?

```
    10 LET SP=0.3
 ●▲ 20 CLS
    30 LET H=0
    40 LET K=0:LET S=0:LET L=1
    50 PRINT ">>> ROBOSPY <<<":PRINT
    60 PRINT "PRESS G TO GO"
★●▲ 70 LET C$=INKEY$
    80 IF C$<>"G" THEN GOTO 70
 ●▲ 90 CLS:PRINT:PRINT
   100 IF K=5 THEN LET L=L+1:LET K=0
   110 LET M$=""
   120 LET C$=""
   130 FOR T=1 TO L
 ☼■ 140 LET N=RND(1)
   150 IF N>=0.5 THEN PRINT "LEFT":LET M$=M$+"L"
   160 IF N<0.5 THEN PRINT "RIGHT":LET M$=M$+"R"
   170 NEXT T
   180 LET K=K+1
   190 FOR D=1 TO 200*L*SP
★●▲ 200 LET C$=INKEY$
   210 NEXT D
 ●▲ 220 CLS:PRINT:PRINT
   230 LET E=0
   240 PRINT "WHAT WERE THE DIRECTIONS"
   250 FOR T=1 TO L
   260 PRINT "?";
★●▲ 270 LET C$=INKEY$
   280 IF C$<>"R" AND C$<>"L" THEN GOTO 270
   290 PRINT C$
  ☼ 300 IF C$<>MID$(M$,T,1) THEN LET E=1
   310 NEXT T
   320 IF E=0 THEN LET S=S+L
   330 IF S>=H THEN LET H=S
   340 IF E=1 THEN PRINT "YOU LOST HIM! SOME SPY!"
   350 PRINT "SCORE = ";S
   360 PRINT "HIGH SCORE = ";H
   370 IF E=1 THEN GOTO 40
   380 FOR D=1 TO 500*SP
   390 NEXT D
   400 GOTO 90
```

How it works

30-40: Set up variables. L is number of words printed on screen. K counts how many times L words have been printed.

60-80: Print message and wait for key press to start.

90: Clears screen and leaves 2 empty lines.

100: If K is 5, value of L is increased by 1 and K set to zero.

110-120: Set up two empty string variables.

130-170: Loop round L times choosing sequence of lefts and rights randomly, print them and store them in M$.

180: Increases K by 1.

190-210: Pause for player to see words. Gets longer for more words.

230: Error flag – will be set to 1 if player makes a mistake.

250: Starts loop to get and check answers.

260-270: Print prompt, get a key press and put it in C$.

280: If key press not L or R, goes back for another.

300: Checks if letter in C$ matches appropriate letter in M$ and sets error flag if not.

320: Adds to score if no error.

330: Sets new high score if necessary.

340: Prints message if error made.

350-360: Print scores.

370: Goes back for new game.

380-390: Pause to see score.

400: Goes back for next go.

Conversion lines

```
● 20,90,220 Replace CLS with HOME
▲ 20,90,220 Replace CLS with PRINT CHR$(147)
★ 70,200,270 LET C$=INKEY$(0)
● 70,200,270 C$="":IF PEEK(-16384)>127 THEN GET C$
▲ 70,200,270 GET C$
☼ 140 LET N=RND
■ 140 LET N=RND(0)
☼ 300 IF C$<>M$(T) THEN LET E=1
```

Line 10 controls the speed of the game. Change this number as follows: Spectrum 0.2, VIC 20 0.4, Electron 0.6, BBC 1, Apple 0.1. Whichever computer you are using, the lower the number, the faster the game. Try speeding it up and see how good you are.

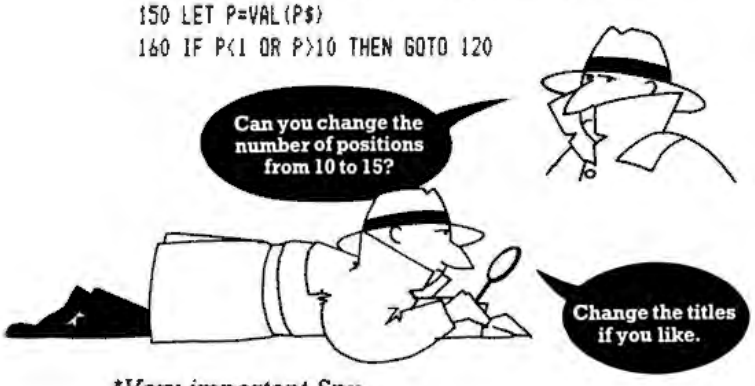
Spy Q Test

As a new recruit at Spy School, you've a lot of Spy Q tests to pass if you are to move up through the Grades. You start as lowest of the low – a Grade 5 Trainee Spy. Your goal is to reach the top and become a Grade 1 VIS* and even achieve the ultimate accolade: the Super Spy Award.

In each Spy Q Test, you are given ten positions on your computer screen. You are then given numbers between 1 and 100. Your aim is to put these numbers in order into the ten positions, with the lowest in position 1 and the highest in 10.

You are allowed to discard some numbers if they won't fit by pressing D. The number of numbers you are allowed to discard is the same as the number of your grade.

```
10 DIM N(10):DIM N$(5)
20 CLS
30 GOSUB 510
40 LET W$=""
50 LET D=5
60 LET G=0
70 FOR I=1 TO 10:LET N(I)=0:NEXT I
80 LET I=1
90 GOSUB 430
100 LET M=INT(RND(1)*99+1)
110 PRINT:PRINT "WHERE WILL YOU PUT ";M
120 PRINT:INPUT P$
130 IF P$="D" AND G<D THEN LET G=G+1:GOTO 90
140 IF P$="D" THEN PRINT "YOU CAN'T!":GOTO 120
150 LET P=VAL(P$)
160 IF P<1 OR P>10 THEN GOTO 120
```



Very important Spy

How the program works

10: Sets aside storage space for arrays.

20: Clears screen.

30: Goes to subroutine which reads data to put in array N$().

40: Sets W$ to empty string.

50: Sets grade to 5 for start.

60: G is number of discards used.

70: Puts zeros in all 10 positions for start.

80: Counts numbers player has positioned.

90: Goes to subroutine which prints your status, the numbers 1 to 10 and any numbers already in those positions.

100: Chooses a number between 1 and 100.

110-120: Print number and find out where you want to put it.

130-140: Check if you pressed D and then whether or not you are allowed to discard.

150: Works out numerical value of your input.

160: Rejects if not between 1 and 10.

```
170 IF N(P)>0 THEN PRINT "ALREADY FULL":GOTO 120
180 LET N(P)=M
190 LET F=0
200 FOR L=P TO 10
210 IF N(L)<M AND N(L)<>0 THEN LET F=1
220 NEXT L
230 FOR L=1 TO P
240 IF N(L)>M AND N(L)<>0 THEN LET F=1
250 NEXT L
260 IF F=1 THEN GOTO 360
270 LET I=I+1:IF I<11 THEN GOTO 90
280 LET D=D-1:IF D=0 THEN GOTO 330
290 PRINT "WELL DONE, GO TO GRADE ";D
300 PRINT:PRINT "YOU ARE NOW A ";N$(D)
310 LET W$=""
320 GOTO 400
330 PRINT "TERRIFIC - YOU HAVE REACHED"
340 PRINT "THE GRADE OF SUPER SPY"
350 STOP
360 PRINT "WRONG! NOT GOOD ENOUGH"
370 PRINT:PRINT "YOU ARE STILL A ";
380 PRINT N$(D)
390 LET W$="STILL"
400 PRINT:PRINT "DO YOU WANT TO TRY AGAIN? (Y/N)"
410 INPUT A$:IF A$="Y" THEN GOTO 60
420 STOP
●▲430 CLS
440 PRINT:PRINT "YOU ARE ";W$;" A ";N$(D)
450 PRINT
460 FOR J=1 TO 10
470 PRINT J;
480 IF N(J)>0 THEN PRINT N(J);
490 PRINT:NEXT J
500 RETURN
510 FOR I=1 TO 5:READ N$(I)
520 NEXT I
530 RETURN
540 DATA "VIS","SPY","JUNIOR SPY"
550 DATA "SPYING ASSISTANT","TRAINEE SPY"
```

170: Rejects if position already used.

180: Puts number in position you wanted.

190-220: Check all the positions above the one you said to see if there is a lower number already in one of them. Set F to 1 if there is.

230-250: Check positions below and set F to 1 if there is a higher number already in one of them.

260: If position is wrong, goes down program to tell you so.

270: Increases counter and goes back for another number.

280-300: Change D to next grade and print message. (Goes to Super Spy if D is 0.)

310: Sets W$ to empty string.

320: Goes to "try again" message.

330-350: Super Spy message.

360-380: Print losing message.

390: Sets W$ to "still".

400-420: Find out if you want to try again, if not stop game.

430-500: Subroutine which prints message and numbers.

510-530: Subroutine which reads names of grades into array N$.

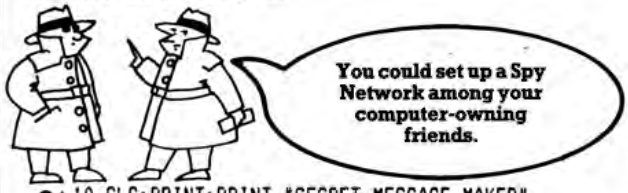
540-550: Data.

Conversion lines

```
✡10 DIM N(10):DIM N$(5,16)
●20,430 Replace CLS with HOME
▲20,430 Replace CLS with PRINT CHR$(147)
✡100 LET M=INT(RND*99+1)
■100 LET M=RND(99)
```

Secret Message Maker

Use this program to send coded messages to your friends. They'll need a computer to decode them, but not necessarily the same type as yours. (They'll need a copy of the program too don't forget.) They can decode your messages and then send coded messages back to you.




```
●▲10 CLS:PRINT:PRINT "SECRET MESSAGE MAKER"
   20 PRINT "==============="
   30 PRINT:PRINT "DO YOU WANT TO:"
   40 PRINT
   50 PRINT "   1. CODE A MESSAGE"
   60 PRINT "OR 2. DECODE A MESSAGE"
   70 PRINT:PRINT
   80 PRINT "ENTER NUMBER":INPUT A
   90 IF A=1 THEN GOSUB 120
  100 IF A=2 THEN GOSUB 210
  110 GOTO 30
  120 LET C$="CODED":GOSUB 400
■☼130 LET X=INT(RND(1)*25+1)
  140 LET M$="F"+M$
  150 GOSUB 310:GOSUB 420
  160 LET M$=CHR$(X+64)+M$
  170 IF LEN(M$)/2=INT(LEN(M$)/2) THEN GOSUB 450
  180 PRINT "THE CODED MESSAGE IS:"
  190 PRINT M$
  200 RETURN
  210 LET C$="DECODED":GOSUB 400
  220 IF LEN(M$)/2=INT(LEN(M$)/2) THEN GOSUB 450
 ☼230 LET K$=LEFT$(M$,1)
 ☼240 LET M$=RIGHT$(M$,LEN(M$)-1)
 ☼250 LET X=ASC(K$)-64
  260 LET X=-X:GOSUB 420
```

How the program works

10-80: Print "menu" and ask whether coding or decoding required.

90-100: Go to relevant subroutine.

110: Goes back to menu again.

120-200: CODING SUBROUTINE
120: Goes to subroutine to ask for message.

130: Chooses amount to shift letters and stores it in X.

140: Adds a letter in case first word is only one character.

150: Goes to subroutines to shift messages X letters along alphabet and then reverse them.

160: Adds letter to message to tell decoder subroutine what the value of X is.

170: Checks if message has even number of characters. If so, goes to subroutine to swap letters in each pair.

210-300: DECODING SUBROUTINE
210: Goes to subroutine to ask for message.

220: If message contains an even number of letters then swap them round.

230-240: Take off first letter. (This is the one added at line 160 to show amount of shift.)

250: Works out the value of X.

260: Changes direction of shift and goes to subroutine which reverses letters.

```
✡270 GOSUB 310:M$=RIGHT$(M$,LEN(M$)-1)
  280 PRINT "THE DECODED MESSAGE IS:"
  290 PRINT M$
  300 RETURN
  310 LET N$=""
  320 FOR I=1 TO LEN(M$)
✡330 LET Q$=MID$(M$,I,1):LET N=ASC(Q$)
  340 IF N=32 THEN GOTO 380
  350 LET N=N+X
  360 IF N>90 THEN LET N=N-26
  370 IF N<65 THEN LET N=N+26
  380 LET N$=N$+CHR$(N):NEXT I
  390 LET M$=N$:RETURN
  400 PRINT "WHAT IS THE MESSAGE TO BE ";C$
  410 INPUT M$:RETURN
  420 LET N$="":FOR I=LEN(M$) TO 1 STEP -1
✡430 LET N$=N$+MID$(M$,I,1):NEXT I
  440 LET M$=N$:RETURN
  450 LET N$="":LET L=LEN(M$)
  460 FOR I=1 TO LEN(M$)-1 STEP 2
✡470 LET N$=N$+MID$(M$,I+1,1)
✡480 LET N$=N$+MID$(M$,I,1)
  490 NEXT I:LET M$=N$:RETURN
```

270: Goes to subroutine which shifts letters through alphabet. Takes first letter off.

310-390: SHIFTING SUBROUTINE
310: Puts message in N$.

320: Start of loop, which takes letters from N$ one at a time.

330: Changes letter into computer's code number.

340: If a space, keep the same.

350: Adds value of X to letter code.

360-370: Adjust code number if it goes beyond the code for Z or before A.

380: Adds new letter to N$ and goes back to beginning of loop.

390: When all letters have been shifted, the new message is put in M$.

400-410: Subroutine which asks for a message.

420-440: Subroutine which reverses letters.

450-490: Subroutine which takes pairs of letters from message in turn and switches the two letters in each pair round.

Conversion lines

```
●10 Replace CLS with HOME
▲10 Replace CLS with PRINT CHR$(147)
✡130 LET X=INT(RND*25)+1
■130 LET X=RND(25)
✡230 LET K$=M$(1)
✡240 LET M$=M$(2 TO )
✡250 LET X=CODE(K$)-64
✡270 GOSUB 310:LET M$=M$(2 TO )
✡330 LET Q$=M$(I):LET N=CODE(Q$)
✡430 LET N$=N$+M$(I):NEXT I
✡470 LET N$=N$+M$(I+1)
✡480 LET N$=N$+M$(I)
```

Rendezvous

Your mission is a complicated one, so read these instructions carefully.

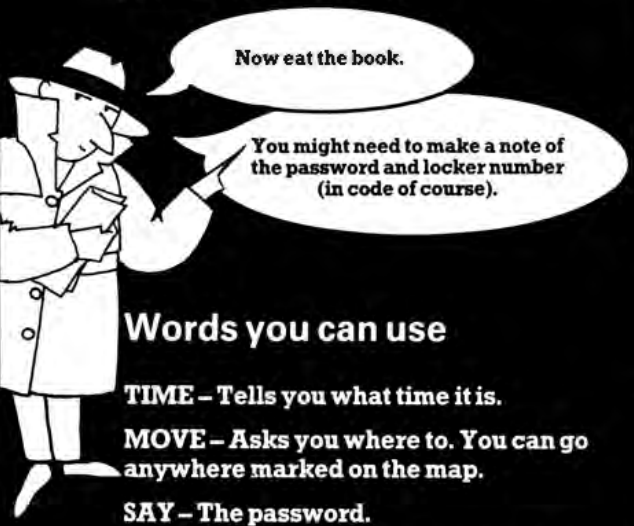
You must collect a case from a locker at the station, hand it over to your contact and get back to the airport before the last plane takes off (your computer will tell you what time this is).

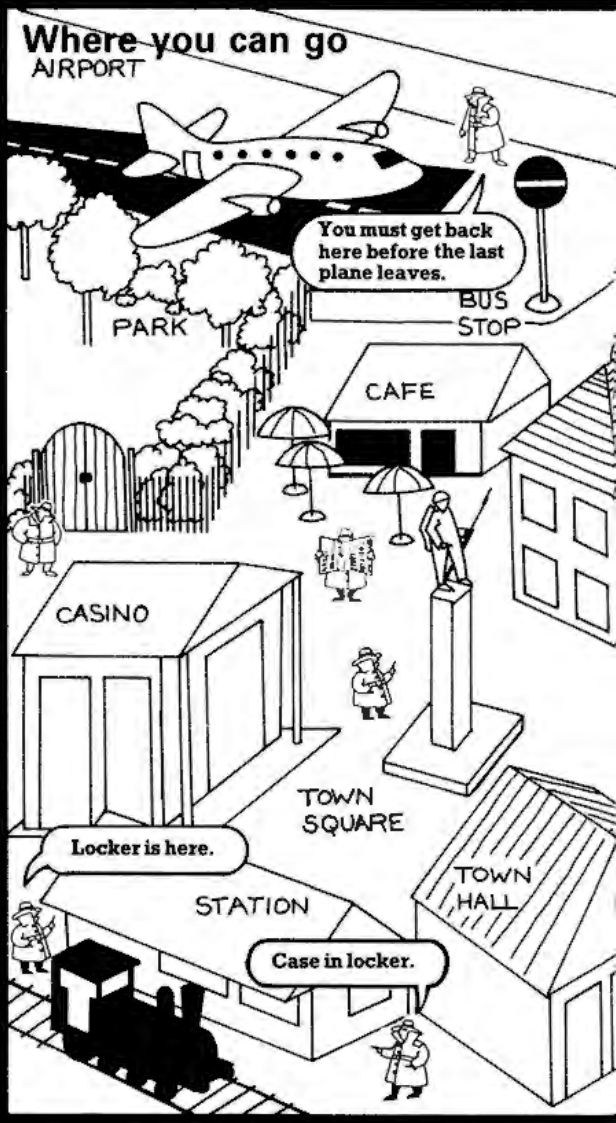
Your computer will tell you where your contact will be at what time. You must leave a message at that place, before he gets there, telling him where and when you will meet him to hand over the case.

You must find out the password before you meet him, and make sure you are not more than 15 minutes late.

Before you can get the case, you must find the key to the locker and also its number. Unfortunately the key is in the hands of enemy agents, whose HQ is at the Hotel. You must find an enemy spy and follow him, hoping he will be careless enough to drop the key (and of course that he won't see you).

The map shows you the places you can go to and the list below shows the words you can use in the game.





Words you can use

TIME – Tells you what time it is.

MOVE – Asks you where to. You can go anywhere marked on the map.

SAY – The password.

EXAMINE – Anything. (Examine the key to get the number.)

READ – A message.

OPEN – The locker.

FOLLOW – An enemy spy.

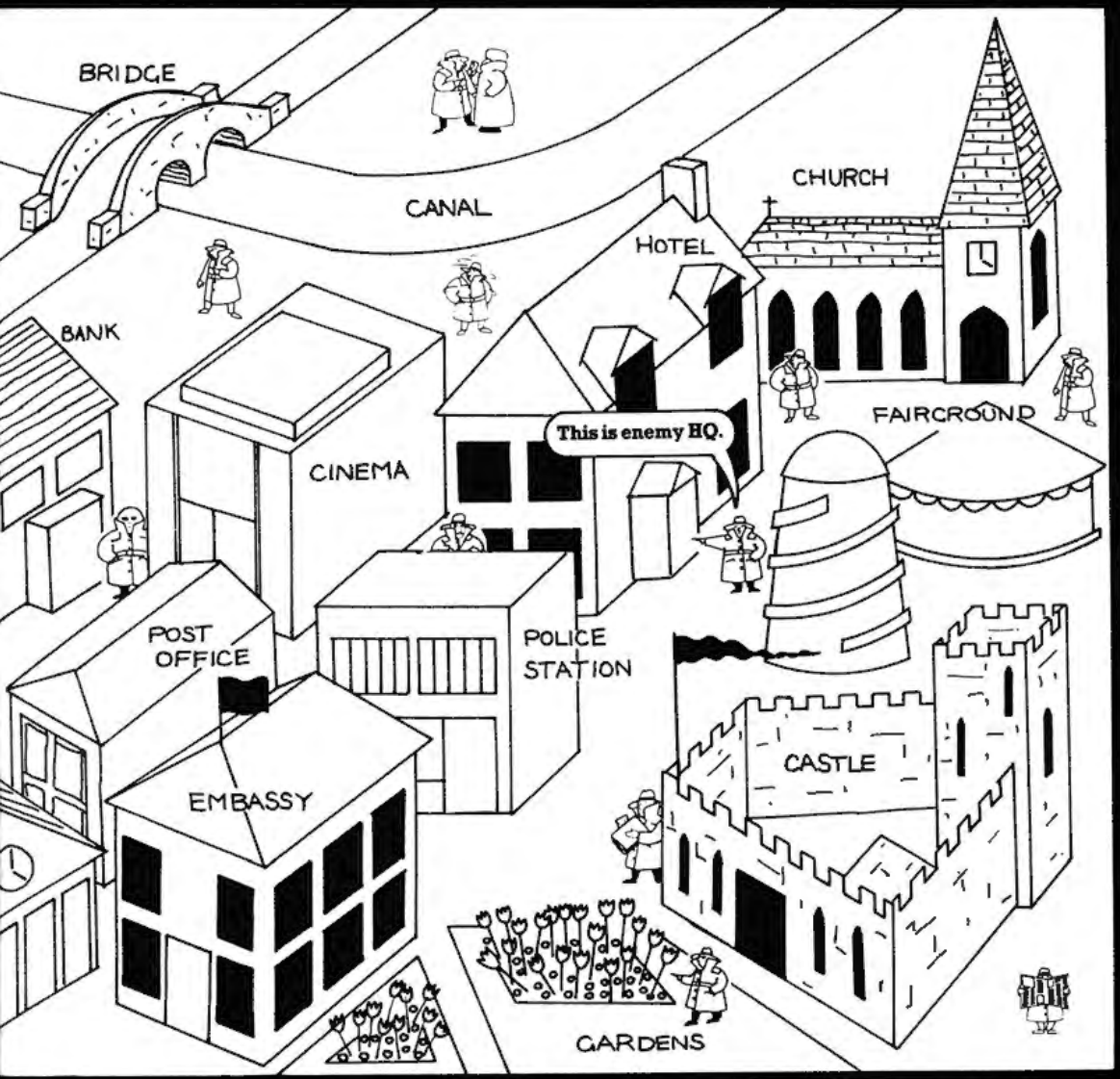
WAIT – For any length of time.

LEAVE – A message.

SEARCH – Anywhere (to find the key).

HELP – Reminds you of time and place of meeting.

You can also use any of the names in capital letters on the map.



If you are using a VIC 20, you will need extra memory for this game.

```
10 DEF FNA(X)=INT(RND(1)*X)+1
20 GOSUB 1010:GOSUB 1130
30 LET P=1:LET Y=0:LET X=1:LET EP=10
40 LET U=0:LET NM=0
50 LET S$=""
60 LET B$=""
70 CLS:PRINT:PRINT "RENDEZVOUS"
80 PRINT "=========="
90 LET NM=NM+1
100 PRINT:PRINT
110 PRINT B$:PRINT
120 PRINT:PRINT "YOU ARE AT THE ";
130 PRINT R$(P)
140 IF EP=P THEN PRINT "ENEMY AGENT IS HERE":LET NE=NE+1
150 IF EP<>P THEN LET NE=0
160 IF P=MP AND F(1)=0 THEN PRINT "MESSAGE FOR YOU HERE"
170 LET F(3)=0
180 LET T1=H+M/100
```

How the program works

10: Sets up a function to choose random number between 1 and X.

20: Go to end of program to read in data and set game (initialize).

30: P, Y and X record player's position. EP is enemy position.

60: B$ is for computer's messages to player.

90: Increases NM (number of moves).

140: NE counts how long player is in same place as enemy agent.

180: Works out the time (H = hours, M = minutes)

```
190 IF F(4)=1 AND R$(P)=S$ AND U(=T1 AND U+.15)T1 THEN PRINT T$:LET F(3)=1
200 IF P=1 AND H(FH AND F(7)=1 THEN GOTO 890
210 PRINT:PRINT:PRINT "WHAT NEXT"
220 LET B$=""
230 INPUT I$
240 LET V=0:FOR I=! TO 11
250 IF I$=V$(I) THEN LET V=I
260 NEXT I
270 IF V=0 THEN LET V=12
280 IF NE=3 AND FNA(10))3 AND V(>1 THEN LET B$="ENEMY AGENT SEES YOU!":GOTO 70
290 IF NE=4 THEN PRINT "YOU ARE CAPTURED!":STOP
300 ON V GOSUB 360,420,490,540,570,640,710,730,780,810,820,870
310 LET M=M+DT:IF M)59 THEN LET M=M-60:LET H=H+1
320 IF F(2)=1 AND H)=CH THEN LET F(4)=1
330 IF H=FH THEN GOTO 880
340 IF FNA(10))9 THEN LET EP=10
350 GOTO 70
360 PRINT:PRINT "WHERE TO":INPUT N$
370 LET NP=0:FOR I=1 TO 20
380 IF N$=R$(I) THEN LET NP=I
390 NEXT I:IF NP=0 THEN GOTO 360
400 GOSUB 950
410 LET P=NP:RETURN
420 LET DT=5
430 PRINT:PRINT "SAY WHAT":INPUT Q$
440 IF EP=P THEN LET B$="YOU ATTRACTED THE ENEMY AGENT!":RETURN
450 IF F(3)=0 THEN LET B$="NOBODY HEARS YOU":RETURN
460 IF Q$(>P$ THEN LET B$="CONTACT IGNORES YOU!":RETURN
470 IF F(6)=1 THEN LET B$="YOU MADE CONTACT - HE TAKES THE CASE!"
480 LET F(7)=1:RETURN
490 LET DT=5
500 PRINT:PRINT "WHAT DO YOU WANT TO EXAMINE":INPUT Q$
510 IF Q$="CASE" THEN LET B$="TOP SECRET!":RETURN
520 IF Q$="KEY" THEN LET B$="A NUMBER - "+STR$(NL):RETURN
530 LET B$="NOTHING SPECIAL!":RETURN
540 IF P(>MP OR F(1)=1 THEN LET B$="NOTHING TO READ!":RETURN
550 LET B$="A WORD - '"+P$+"'"
560 LET F(1)=1:RETURN
570 LET DT=5
580 IF P(>16 THEN LET B$="NOTHING TO OPEN":RETURN
590 IF F(5)=0 THEN LET B$="YOU HAVE NO KEY":RETURN
600 PRINT:PRINT "WHAT NUMBER LOCKER":INPUT YN
610 IF NL(>YN THEN LET B$="THE KEY DOES NOT FIT":RETURN
620 LET B$="LOCKER IS OPEN - YOU HAVE THE CASE!":LET F(6)=1
630 RETURN
640 LET DT=5
650 IF EP(>P THEN LET B$="FOLLOW WHO?":RETURN
660 LET NP=FNA(20):GOSUB 950:LET P=NP
670 IF FNA(10))8 THEN LET P=KP
680 IF FNA(10))7 THEN LET B$="YOU LOST HIM AFTER A WHILE!":RETURN
690 LET EP=P
700 LET B$="YOU KEPT HIM IN SIGHT":RETURN
710 PRINT:PRINT "HOW MANY MINUTES":INPUT DT
720 RETURN
730 PRINT:PRINT "WHERE DO YOU WANT TO MEET":INPUT S$
740 PRINT:PRINT "WHAT TIME (HH.MM)"
750 INPUT U
760 IF P=CP AND T1(U AND H(CH THEN LET F(2)=1
770 LET DT=5:RETURN
780 LET B$="NOTHING HERE":LET DT=10
790 IF P=KP THEN LET B$="YOU FOUND A KEY":LET F(5)=1
800 RETURN
810 LET DT=0:LET B$="TIME IS NOW "+STR$(H)+"."+STR$(M):RETURN
820 LET DT=5
830 IF U=0 THEN GOTO 860
```

190-200: Work out if mission has been completed.

210-270: Get word from player and check for match with words in memory. V is number of matching word.

280-290: Check value of NE to see if seen by enemy.

300: Branches to subroutine depending on word entered.

310: Increases time.

330: Checks if player has run out of time.

340: Moves enemy back to hotel.

360-410: MOVE SUBROUTINE

420-490: SAY SUBROUTINE

500-540: EXAMINE SUBROUTINE

550-570: READ SUBROUTINE

580-640: OPEN SUBROUTINE

650-700: FOLLOW SUBROUTINE

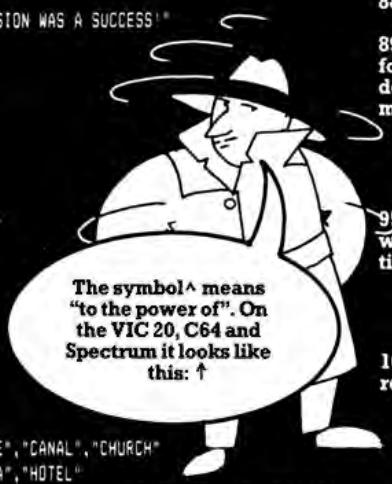
710-720: WAIT SUBROUTINE

730-770: LEAVE SUBROUTINE

780-800: SEARCH SUBROUTINE

810: TIME SUBROUTINE

```
 840 PRINT:PRINT "MEETING PLACE IS"
 850 PRINT S$;" AT ";U
 860 GOSUB 1300:RETURN
 870 LET DT=0:LET B$="PARDON?":RETURN
 880 PRINT "TOO LATE ":STOP
 890 PRINT:PRINT "WELL DONE. YOUR MISSION WAS A SUCCESS!"
 900 LET TL=(FH-H)*60-M
 910 LET S=INT((20/NM+TL/120)*50)
 920 PRINT:PRINT "YOUR SPY RATING"
 930 PRINT "IS ":S
 940 STOP
 950 LET NY=INT((NP-1)/5)
 960 LET NX=NP-5*NY
 970 LET DX=ABS(X-NX):LET DY=ABS(Y-NY)
 980 LET X=NX:LET Y=NY
 990 LET D=SQR(DX^2+DY^2)
1000 LET DT=INT(5*D):RETURN
1010 DIM R$(20),V$(11),F(7)
1020 FOR I=1 TO 20:READ R$(I)
1030 NEXT I
1040 FOR I=1 TO 11:READ V$(I):NEXT I
1050 RETURN
1060 DATA "AIRPORT","BUS STOP","BRIDGE","CANAL","CHURCH"
1070 DATA "PARK","CAFE","BANK","CINEMA","HOTEL"
1080 DATA "CASINO","TOWN SQUARE","POST OFFICE","POLICE STATION","FAIRGROUND"
1090 DATA "STATION","TOWN HALL","EMBASSY","GARDENS","CASTLE"
1100 DATA "MOVE","SAY","EXAMINE","READ","OPEN","FOLLOW","WAIT","LEAVE","SEARCH"
1110 DATA "TIME","HELP"
1120 DATA "CUSTARD","KIPPER","KOALA","CRUMPET","CROSSWORD","KANGAROO"
1130 LET H=FNA(2)+8:LET M=0
1140 LET FH=FNA(2)+14
1150 LET CH=FNA(2)+H
1160 CLS:PRINT:PRINT
1170 LET NE=0:LET T$="CONTACT IS HERE"
1180 LET MP=FNA(18):LET KP=FNA(18)
1190 LET EP=FNA(18):LET CP=FNA(18)
1200 FOR I=1 TO FNA(6)
1210 READ P$:NEXT I
1220 LET NL=FNA(900)+99
1230 PRINT "RENDEZVOUS"
1240 PRINT "==========":PRINT
1250 PRINT "COLLECT MESSAGE FROM"
1260 PRINT "THE ";R$(MP)
1270 PRINT "CONTACT WILL COLLECT"
1280 PRINT "FROM THE ";R$(CP)
1290 PRINT "AT ";CH;".00"
1300 PRINT "LAST FLIGHT LEAVES"
1310 PRINT "AT ";FH;".00"
1320 PRINT:PRINT"PRESS RETURN TO CONTINUE"
1330 INPUT Q$:RETURN
```



820-860: HELP SUBROUTINE

870: "Word not recognized" SUBROUTINE.

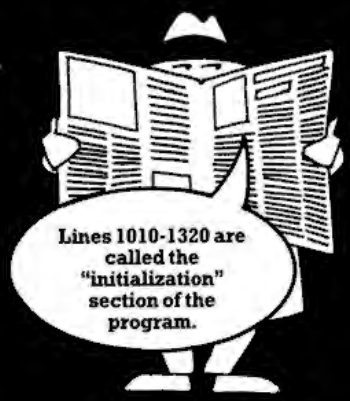
880: "Run out of time" message.

890-940: Print message and score for successful mission. Rating depends on number of moves made and time left.

950-1000: Subroutine which works out distance moved and time taken by player's move.

1010-1050: Subroutine which reads in data.

1060-1120: Data lines.



1130: Chooses starting time.

1140: Chooses time of last flight.

1150: Chooses time for contact to collect message.

1180-1190: Choose positions for message, key, enemy and where contact will collect.

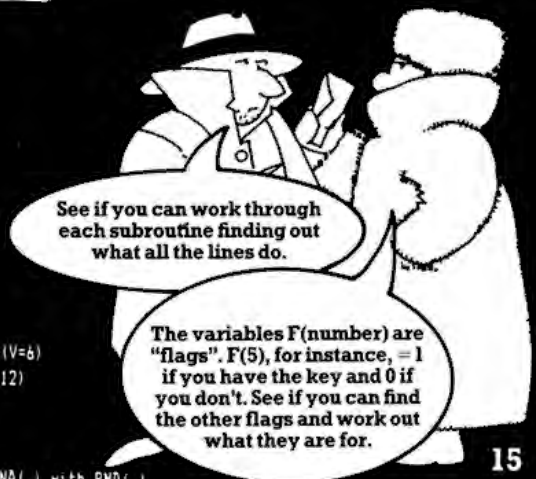
1200-1210: Choose password.

1220: Choose locker number.

1230-1320: Print introduction to game.

Conversion lines

```
10 DEF FNA(X)=INT(RND*X)+1
10 leave out this line
70,1160 Replace CLS with HOME
70,1160 Replace CLS with PRINT CHR$(147)
235 IF LEN(I$)<7 THEN LET I$=I$+" ":GOTO 235
300 GOSUB 360*(V=1)+420*(V=2)+490*(V=3)+540*(V=4)+570*(V=5)+640*(V=6)
    +710*(V=7)+730*(V=8)+780*(V=9)+810*(V=10)+820*(V=11)+870*(V=12)
365 IF LEN(N$)<14 THEN LET N$=N$+" ":GOTO 365
735 IF LEN(S$)<14 THEN LET S$=S$+" ":GOTO 735
1010 DIM R$(20,14):DIM V$(11,7):DIM F(7)
280,340,660,670,680,1130,1140,1150,1180,1190,1200,1220 Replace FNA( ) with RND( )
    e.g. FNA(20) becomes RND(20)
```



Morse Coder

If you want to be a really successful spy, you need to know how to send, receive and, of course, intercept messages in Morse Code. This program will help you learn. If you have never used Morse Code before, you will need to make yourself a chart of letters and their Morse equivalents. Use lines 400-450 of the program to do this. They show the Morse code for each letter of the alphabet in order.

What you have to do

In Morse Code, each letter is represented by a series of long and short sounds or flashes. This program uses a flashing star. It will give you the code for a letter and then ask you which it was. You will have to watch carefully to pick out the long and short flashes and remember them. You will see the cursor flashing too at the left of the screen. Ignore this it has nothing to do with the code.

```
10 GOSUB 370
20 CLS
30 LET S=30
40 PRINT:PRINT "MORSE TESTER"
50 PRINT:PRINT"WHAT LEVEL ?"
60 PRINT:PRINT "(1=FAST)"
70 PRINT "(5=SLOW)"
80 INPUT P:LET P=P*S
90 CLS
100 PRINT:PRINT "GET READY"
110 FOR T=1 TO 20*S:NEXT T
120 GOSUB 310
130 LET F$=M$(ASC(Q$)-64)
140 GOSUB 220
150 CLS:PRINT
160 PRINT "TYPE IN YOUR ANSWER"
170 INPUT X$
180 IF X$=Q$ THEN PRINT "CORRECT"
190 IF X$<>Q$ THEN PRINT "NO. THE ANSWER IS :";Q$
200 FOR T=1 TO 30*S:NEXT T
210 GOTO 90
220 FOR J=1 TO LEN(F$)
230 LET W$=MID$(F$,J,1)

240 IF W$="." THEN LET K=1
250 IF W$="-" THEN LET K=3
260 GOSUB 330:LET K=1
270 GOSUB 340
280 NEXT J
290 RETURN
300 PRINT
310 LET Q$=CHR$(INT(RND(1)*26+65))
320 RETURN
330 PRINT TAB(10,10);"*"
340 FOR T=1 TO P*K:NEXT T
350 PRINT TAB(10,10);" "
360 RETURN
370 DIM M$(26)
380 FOR I=1 TO 26:READ M$(I):NEXT I
390 RETURN
400 DATA ".-","-...","-.-.","-..","."
410 DATA "..-.","--.","....","..",".---"
420 DATA "-.-",".-..","--","-.","---"
430 DATA ".--.","--.-",".-.","...","-"
440 DATA "..-","...-",".--","-..-"
450 DATA "-.--","--.."
```

As you learn the letters, try covering up the chart.

Speed is controlled by line 30. For BBC and Electron change the value of S to 100. (The lower the number, the faster the program.)

You could try adding a scoring routine. (The Robospy program should give you an idea of how to do this.)

You could try adding sounds too, either as well as or instead of the flashing stars. Remember a dot is a short sound and a dash a longer one (3 times as long).

Conversion lines

```
20,90,150 Replace CLS with HOME
20,90,150 Replace CLS with PRINT CHR$(147)
130 LET F$=M$(CODE(Q$)-64)
230 LET W$=F$(J)
235 IF W$=" " THEN GOTO 260
310 LET Q$=CHR$(INT(RND*26+65))
310 LET Q$=CHR$(RND(26)+64)
330 PRINT AT 10,10;"*"
330 VTAB(10):PRINT TAB(10);"*"
330 PRINT CHR$(19):FOR I=1 TO 10:PRINT:
    NEXT:PRINT TAB(10);"*"
330 PRINT@ 240,"*"
350 PRINT AT 10,10;" "
350 VTAB(10):PRINT TAB(10);" "
350 PRINT CHR$(19):FOR I=1 TO 10:PRINT:
    NEXT:PRINT TAB(10);" "
350 PRINT@ 240," "
370 DIM M$(26,4)
```

How it works

10: Goes to subroutine to read in data.

30: Sets speed.

50-80: Ask for level you want and work out speed of flashes depending on what you say.

110: Pause

120: Goes to subroutine which chooses random letter and stores it in Q$.

130: Finds the dots and dashes code for the letter and puts them in F$.

140: Goes to subroutines which print flashing star.

160-170: Get answer from you.

180-190: Check if answer is correct or not and print message.

200-210: Pause then go back for next letter.

220-250: Check through length of F$ setting K to 1 for dots and 3 for dashes. (K sets how long the star stays on the screen each time.)

260-270: Go to subroutine to print star. Come back, set K to 1, then go back to subroutine at line 340 to "print" a space. This gives the gap between the dots and dashes.

280: Goes back for next dot or dash.

310-320: Subroutine to choose a random letter.

330: Turns star on.

340: Waits for length of time depending on K.

350: Prints space instead of star.

370-390: Subroutine which reads in data.

400-450: Data lines.

Answers to puzzles

Here are answers to some of the puzzles set in this book. Your answers may be different, but if they work this doesn't matter. Check they are as neat and simple as the ones given here though.

Spy Eyes

Here is how to add a high score routine.

```
 15 LET H=0
215 PRINT "HIGH SCORE = ";H
275 IF P>H THEN LET H=P:PRINT "THAT IS THE HIGH SCORE"
290 INPUT A$:IF A$="Y" THEN GOTO 20
```

Searchlight

This is how to make the light stay on longer...

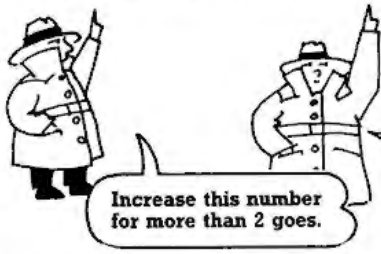
```
430 IF C1=TC THEN LET L=1:LET C1=0:LET TC=INT(RND(1)*10+(8-A))+10:GOSUB 300
```

Use your computer's version of RND.

...and this is how to give the player a second chance.

```
 15 LET NG=0
255 LET NG=NG+1
257 IF NG<2 THEN FOR T=1 TO 200:NEXT T:GOTO 30
```

NG is a new variable which stands for "number of goes".



This loop makes a pause so you can read the message. Increase the number in it for a longer pause.

Increase this number for more than 2 goes.

Robospy

These lines will make the computer ask for the player's name when a new high score is reached.

```
 25 LET H$="NOBODY"
330 Take out this line
365 IF E=1 AND S>H THEN LET H=S:GOSUB 410
410 PRINT "THAT IS THE HIGH SCORE!"
420 PRINT "PLEASE TYPE YOUR NAME"
430 INPUT H$:RETURN
```

Spy Q Test

Here is how to change the number of positions from 10 to 15.

```
10 DIM N(15):DIM N$(5)
70,160,200,460 Change the 10 in these lines to 15
```

Usborne Computer Books

"Highly recommended to anyone of any age." Computing Today
"Without question the best general introduction to computing I have ever seen." Personal Computer World
". . . perhaps the best introduction around . . . outstanding . . ." Educational Computing
"These books are outstanding . . . they make all other young people's computer books look meretricious." Times Educational Supplement

Guide to Computers
Understanding the Micro
Computer and Video Games
Computer Jargon
Computer Graphics
Inside the Chip
Computer Programming
Practise Your BASIC
Better BASIC
Machine Code for Beginners
Practical Things to do with a Microcomputer
Computer Spacegames
Computer Battlegames
Write Your Own Adventure Programs
Creepy Computer Games

New Titles

Programming Tricks & Skills Professional tips and tricks for better programming.

Experiments With Your Computer An exciting introduction to scientific investigation with a microcomputer.

Expanding Your Micro A detailed guide to add-ons and interfaces.

Write Your Own Fantasy Games A step-by-step guide to writing fantasy games, with program listing.

Weird Computer Games, Computer Spy Games Short listings to run on most main home computers.

Mystery of Silver Mountain, Island of Secrets Unique adventure game books containing full program listings.

ISBN 0-86020-828-1

9 780860 208280

Printed in Belgium

99p