How to program the Spectrum in just 60 minutes

INSTANT Spectrum PROGRAMMING

Tim Hartnell

C-60 spoken word cassette, plus complete text of the tape with 19 demonstration programs. A further 30 games, utility and graphics programs are included.

INTERFACE

44-46 Earls Court Road, London W8 6EJ

INSTANT SPECTRUM PROGRAMMING

By
Tim Hartnell

How to program the Spectrum
in just sixty minutes...

First published Interface Publications, 1983
Copyright © Hartnell, 1983

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

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

ISBN 0 907563 22 8

First printing February, 1983

Printed in England by Commercial Colour Press, London E.7.

INSTANT SPECTRUM PROGRAMMING

This learning kit is made up of three elements–the tape, the program listings which follow this introduction, and the game and demonstration programs which follow the script of the spoken word cassette.

You are strongly advised to listen to the tape only with your computer turned on. You should make sure that, at the very least, you run the tape with your computer turned on the very first time you hear it. To do otherwise will greatly diminish the value of the tape.

After the first listening, you may find some value in just listening to it, so long as you refer to the program listings. Once you've heard it several times you may wish to read the whole 'script'. It follows the program listings.

After the script you'll find a number of programs which you should enjoy entering and running. Special thanks to Peter Shaw who helped prepare many of the programs for this book.

Tim Hartnell,
January, 1983

Program A

```
10 PRINT "tim"
20 BEEP 1,1
30 GO TO 10
```

Program B

```
10 PRINT "tim"
15 REM hello
20 BEEP 1,1
30 GO TO 10
```

Program C

```
10 INPUT a$
20 INPUT b
30 LET c=b+b
40 PRINT "hello ";a$
50 PRINT "your number was ";b
60 PRINT c;" is double ";b
```

```
hello tim
your number was 23
46 is double 23
```

Program D

```
10 INPUT a$
20 INPUT b
30 LET c=b+b
40 PRINT "hello ";a$
50 PRINT '"your number was ";b
60 PRINT c;" is double ";b
```

Program E

```
   10 INPUT a$
   20 INPUT b
   30 LET c=b+b
   40 PRINT TAB 6;"hello ";a$
   50 PRINT '"your number was ";b
   60 PRINT c;" is double ";b
```

```
        hello tim

your number was 342
684 is double 342
```

Program F

```
   10 INPUT "ink? ";a
   20 INPUT "paper? ";b
   30 INPUT "position across? ";c
   40 IF c<0 OR c>31 THEN GO TO 3
0
   50 INPUT "position down? ";d
   60 IF d<0 OR d>19 THEN GO TO 5
0
   70 PRINT AT d,c; INK a; PAPER
b;"x"
   80 GO TO 10
```

Program G

```
   10 BORDER 0
   20 INK 2
   30 PAPER 6
   40 CLS
   50 INPUT "give me a number ";a
   60 INPUT "and another ";b
   70 IF a>1 AND b>1 THEN PRINT a
;" and ";b;" are both positive"
   80 PAUSE 200: GO TO 40
```

Program H

```
   10 BORDER 0
   20 INK 2
   30 PAPER RND*7
   40 CLS
   45 PAUSE 25: GO TO 30
   50 INPUT "give me a number ";a
   60 INPUT "and another ";b
   70 IF a>1 AND b>1 THEN PRINT a
;" and ";b;" are both positive"
   80 PAUSE 200: GO TO 40
```

Program I

```
   10 LET a=RND
   20 PRINT a
   30 GO TO 10
```

Program J

```
   10 LET a=INT (RND*10)+1
   20 PRINT a
   30 GO TO 10
```

Program K

```
   10 PRINT '"stand by"
   20 PAUSE 50
   30 PRINT '"die one"
   40 PAUSE 50
   50 GO SUB 130
   60 PAUSE 50
   70 PRINT '"die two"
   80 PAUSE 50
   90 GO SUB 130
  100 PAUSE 100
  110 CLS
  120 GO TO 10
  130 REM subroutine
  140 BEEP .02,RND*50
  150 PRINT INT (RND*6)+1
  160 RETURN
```

Program L

```
  10 FOR a=1 TO 15
  20 PRINT a
  30 BEEP .1,a
  40 PAUSE 20
  50 NEXT a
```

Program M

```
  10 FOR a=-15 TO 0
  20 PRINT a
  30 BEEP .1,a
  40 PAUSE 20
  50 NEXT a
```

Program N

```
  10 FOR a=48 TO 0 STEP -4
  20 PRINT a
  30 BEEP .1,a
  40 PAUSE 20
  50 NEXT a
```

Program O

```
  10 PRINT AT 0,0; FLASH 1;"test
"
  20 INPUT a$: IF a$="s" THEN ST
OP
  30 PRINT AT 0,0; FLASH 0;"test
"
  40 INPUT a$: IF a$="s" THEN ST
OP
  50 RUN
```

Program P

```
  10 PRINT AT 0,0; INVERSE 1;"te
st"
  20 INPUT a$: IF a$="s" THEN ST
OP
  30 PRINT AT 0,0; INVERSE 0;"te
st"
  40 INPUT a$: IF a$="s" THEN ST
OP
  50 RUN
```

Program Q

```
  10 DIM A(5)
  20 FOR B=1 TO 5
  30 INPUT A(B)
  40 NEXT B
  50 PAUSE 100
  60 FOR B=1 TO 5
  70 PRINT "A(";B;") IS ";A(B)
  80 NEXT B
```

Program R

```
  10 DIM A$(5,10)
  20 FOR B=1 TO 5
  30 INPUT A$(B)
  40 NEXT B
  50 PAUSE 100
  60 FOR B=1 TO 5
  70 PRINT "A$(";B;") IS ";A$(B)
  80 NEXT B
```

Program S

```
  10 DIM A(3)
  20 DIM A$(3,5)
  30 FOR B=1 TO 3
  40 READ A(B)
  50 READ A$(B)
  60 NEXT B
  70 FOR C=1 TO 3
  80 PRINT A(C)
  90 PAUSE 30
 100 PRINT A$(C)
 110 PAUSE 30
 120 NEXT C
 130 DATA 99,"HELLO",4294
 140 DATA "HEART",-32676,"BONES"
```

Welcome to this cassette, INSTANT SPECTRUM PROGRAMMING. On this tape I will be leading you through most of the functions available from the SPECTRUM keyboard. The tape is designed to make the process of learning to program as easy as possible. While the tape is running I'll assume you will have your SPECTRUM connected up as shown in the manual and also that you have the booklet that accompanies this tape by you.

This tape is not designed to be listened to right through without taking a break. Rather, I think it be best if you listen to the description of a particular command or function on the SPECTRUM, trying out the programs when we come to them, then stop the tape and keep on experimenting with the particular command we have just discussed.

If you are not quite sure what I mean by some of the things I say you can listen to that section of the tape a number of times until it is clear. From time to time, of course, I will be referring to the booklet that comes with this tape. There are a number of program fragments there which we will be using in the process of learning.

The first thing we would like to look at is the keyboard. The SPECTRUM keyboard is somewhat bewildering when you first look at it–it seems there are so many words written on the keys that you'll never be able to work out what they all mean. It's worth being patient at this stage–following through the instructions that I give you, step by step. You'll be surprised at how easy it is to figure out the maze of words and symbols which confront you at the moment.

We'll start by looking at two very important keys which you will find on the bottom row. In the left hand corner there is the key marked CAPS SHIFT and second from the right in the bottom row is SYMBOL SHIFT. These keys are probably the most two important ones on the entire keyboard. These are the keys that decide what you are going to get when you actually hit a particular key.

Start by pressing the Z key which is right next to the CAPS SHIFT, the second one across from the bottom. The word COPY will appear on the screen. Holding down CAPS SHIFT press the zero key–the one in the extreme top right hand corner. If you do this, the word COPY will disappear. Holding down CAPS SHIFT and pressing the zero key acts as a 'rub out'. The word above the key is DELETE which you get by holding down the white shift key.

You get the white words above keys by holding down the SHIFT KEY and then pressing the relevant keys. Now the words in white

are, on the top row starting from the left, EDIT above the 1 key and then CAPS LOCK above 2, TRUE VIDEO above 3 and INV. VIDEO which stands for inverse video above 4. Above the 9 there is the word GRAPHICS and the word DELETE, which we have already seen, is above the zero key.

~~If you~~ let go of the CAPS SHIFT key and hold down the RED SYMBOL SHIFT, (that's the one on the right hand side on the bottom row) and then press a key such as the O key, you should get a semi-colon appearing on the screen. The semi-colon, as you can see, is written above that funny word POKE on the key. Still holding down the red key, press the G key (about the middle of the keyboard) and the word THEN will appear/So, you can see that you get the red word or symbol on a key by holding down the RED SYMBOL SHIFT KEY, then pressing the relevant key.

To get the words in green above a key, you need to press both SHIFT KEYS at once, then let them go and touch the relevant key. Try it now by pressing both CAPS SHIFT and SYMBOL SHIFT at the same time. Hold them down, let them go, then press the T key. That's hold both down, let them go and then press the T key. The word RND should appear.

The only other thing we have to look at in the use of the SHIFT KEYS, are the red words underneath the keys. To get these you hold down both SHIFT KEYS at once, then release CAPS SHIFT (take your left hand finger off) but continue to hold down the SYMBOL SHIFT KEYS. Then you press the needed key.

By pressing the Z key should get you the word BEEP. Try it. Press both SHIFT KEYS down at once, then lift your left hand finger that's holding down CAPS SHIFT while leaving your right hand finger on SYMBOL SHIFT. Now, press the Z key and the word BEEP should appear. This is the way you get words like INK, PAPER, FLASH, BRIGHT, OVER, and INVERSE, the words in red underneath the letters.

Hold down CAPS SHIFT and put your finger on the zero key until you have rubbed out everything you have entered so far. ~~All that~~, perhaps, is not totally clear yet but you now know the rudiments of working the keyboard. I suggest you now wind the tape back to the beginning and go through it all again.

Before you do, there's one more thing I would like to discuss about the keyboard at this stage–the use of the GRAPHICS. If you hold down CAPS SHIFT and then press the 9 key you'll see that the

CURSOR, (that little black object which has been following you around since you began typing into the computer) has turned into a G (for GRAPHICS). If you press the 8 key, still holding down CAPS SHIFT, you'll see you get a little solid square as shown on the key. Try going across now–7, 6, 5, 4, 3, 2, 1,–and you'll see that in each case you get the little design shown on the key. These can be used to build up pictures.

If you look at the right hand side of your SPECTRUM, you'll see the second key from the bottom has the word ENTER on it. The ENTER key is one of the most important ones on the keyboard. You use it after you have typed in a program line to tell the computer you have finished it and enters it into the main body of the ~~computer~~. program

If you now look at the R key you'll see the word RUN. If you have a program in your computer, and you press RUN, the computer is almost ready then to go through and do what it has to do from the program. All you have to do is press ENTER (after pressing the R key for RUN). Then the computer will run the program. In general terms, you have to press ENTER to get the computer to execute anything which you have just entered.

It's time that we had a look at our first program. Make sure there's absolutely nothing in the computer. The best idea is to turn off the power and then turn it on again. This is not always advisable, but in this particular case it is the simplest way of 'emptying' the computer. We are now going to enter the first program, the one marked A in the booklet.

A

```
10 PRINT "tim"
20 BEEP 1,1
30 GO TO 10
```

Type in the number 10 and then press the P key. If you do this the word PRINT will appear. Now hold down the SYMBOL SHIFT KEY (the red one second from the right on the bottom) and press the P key again. If all is well a little double quote mark will appear. Now type in your name. Once you have done this, hold down the SYMBOL SHIFT KEY again and press the P key and again you will get the little quote mark. Now press the ENTER key and you'll see that the program line you have just written goes to the top of the screen.

This means it has been accepted by the computer. Now type in the number 20. Hold both SHIFT KEYS down (the CAPS SHIFT KEY on the left and the RED SYMBOL SHIFT) at the same time. Then, let go the CAPS SHIFT with your left hand still holding the SYMBOL SHIFT with your right and press the Z key. The word BEEP will appear. Now press the number 1. Follow this by holding down the RED SYMBOL SHIFT and press the N key. You'll get a little comma. Then press the number 1 again. Your line should read as in program A, 20 BEEP 1, 1.

When you are satisfied with this, press the ENTER key and the new line will join the 10 line at the top of the screen. Now type in 30 followed by the command GO TO, which you will get by pressing the G key. Notice that the first time you press a key after you have entered a program line number the word in white on the key appears. This is known as a 'key word.' Now if all is well, you've got 30 GO TO. Now type in the number ten (10) and press ENTER.

At the top of your screen you now have your very first program, a three-line program numbered in tens. Although the SPECTRUM will accept line numbers between 1 and 9999, we tend to work in tens because it makes it fairly easy to insert new lines in between the others if we discover we need to later on.

I suggest that once you have the program running you turn off the tape and play with the program for a while. Press the R key so the word RUN appears and then press ENTER. You'll find that the screen will slowly fill up with your name making a fairly monotonous noise as it does so. It will eventually stop with the word 'scroll' and a question mark at the bottom. If you press any key on the keyboard (except for BREAK SPACE) you'll find out that the program will continue. When you are tired of running this program press BREAK SPACE, and it will stop.

Now that you have run that program, it is time to look at our next important word–REM. You'll find the REM on the E key. REM stands for REMARK and a REMARK in a program is there only for *your* benefit; the computer ignores it completely.

Let me show you how it works. Type in the number 15 then press the E key and the word REM will appear. Now put anything you like after that; maybe you could put HELLO or something. Just type in a word. Press the ENTER key again and you'll find that line 15 automatically moves up to its correct position between lines 10 and 20 as you can see in program B.

B

```
10 PRINT "tim"
15 REM hello
20 BEEP 1,1
30 GO TO 10
```

If your program doesn't look like program B you have done something wrong so I suggest you re-wind the tape slightly and see what you have left out. Now run the program again and you should find that the REM statement makes no difference at all. Stop the tape, run the program and then come back to the tape. The computer ignores REM statements as you can see.

We can look now at another word in white above the keys that is very important, the word EDIT which is above the 1 key. If you look at your program on the screen, you'll see that there is a little sideways V pointing at line 15, the latest line you entered. If you hold down CAPS SHIFT and then press 1 you'll see line 15 comes to the bottom of the screen. Now when it's at the bottom of the screen you can easily modify it. Still holding down CAPS SHIFT press the 8 key and you'll see that the cursor moves across the REM statement. First it jumps across the word REM in one jump, and then goes through the letters HELLO or whatever word you have got in line 15, one by one. Get the cursor to the very end, and still holding down CAPS SHIFT, press the zero key (DELETE). You'll see that, one by one, the letters will be rubbed out. Once you have rubbed out the word HELLO, type anything else you like.

Then let go of CAPS SHIFT and press the ENTER key. You'll see the new version of line 15 takes the place of the old line 15 at the top. The EDIT key is very useful for changing the contents of program lines without having to type the entire line all over again. It doesn't matter much when there's a short line, but with a long line you'll appreciate how useful it is to change program lines in this way.

As you have just seen, the 5, 6, 7 and 8 keys are useful for moving the cursor around. You saw that it jumped across REM and then HELLO in your original line 15. If you had then used the 5 key you could have moved the cursor back. You'll see that the little sideways V which is a "greater than" sign is still pointing at line 15. If you hold down the CAPS SHIFT and press the 7 you'll see that this "greater than" sign moves up to line 10. Pressing the 6 will move, eventually, right to the end of the program. Now the line that the cursor is pointing to, is the line that you are able to EDIT. Move the cursor back up to the top (up to line 10) by holding down CAPS SHIFT and press the 7. Then press the 1 key (the EDIT key) and you will see that line 10 comes to the bottom of the screen. Press ENTER and the line 10 disappears from the bottom to join the rest of the program at the top.

I think we have had enough of that particular program. It is very easy to get rid of a program in the computer without turning the power off. Press the A key and you will see the word NEW appear. Press ENTER. You'll see the screen goes black and then clears just leaving the words COPYRIGHT SINCLAIR RESEARCH at the bottom. To check that there is nothing in the computer, press LIST (on the K key) and then press ENTER. If there is program in the computer, it will appear on the screen. As it is empty, all you get is the message 0 OK, 0:1 at the bottom of the screen.

We are now going to enter the next program, program C from the booklet. Instead of reading out the program letter by letter, this time I'd like you to copy it from the booklet.

C

```
10 INPUT a$
20 INPUT b
30 LET c=b+b
40 PRINT "hello ";a$
50 PRINT "your number was ";b
60 PRINT c;" is double ";b
```

```
hello tim
your number was 23
46 is double 23
```

There are a few things to watch when you do copy it. If you are looking at program C now you'll see the first line reads 10 INPUT a $. You'll get the dollar sign from the 4 key by holding down the SYMBOL SHIFT (the red shift key) and then pressing the 4, so that the dollar appears. The equals sign in line 30, and the plus sign you get using the SYMBOL SHIFT–the equals is on the L or LET key, and the plus sign is next to it on the K or LIST key. In line 40 you'll see that after the word hello there is a semi colon. This is very important. You will find that the computer will not accept the line without it. The semi colon is on the O key.

Apart from that you should have no real problems with the program so I suggest you turn off the tape, enter the program then come back to the tape for an explanation of what to do with it.

I assume now you have program C in your computer. Press RUN the R key, followed by ENTER.

What you see at the bottom of the screen now occurs when the computer is waiting for an input, that is, it's waiting for you to enter something. Type in your name and when you have done this, press ENTER. You'll see that the computer then waits for another input. This time it's a NUMERICAL input it wants. Enter a number, press ENTER again. If all is well the computer should now print up 'hello' and your name, 'your number was' and whatever your number was, and then another number 'is double' and your number again. Press LIST then ENTER so that the program comes back up on the screen. I will now lead you through it line by line we can learn a lot from this simple program.

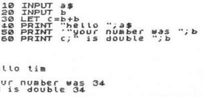
The INPUT command (and you got the input from the I key) tells the computer to wait for you to enter something from the keyboard. The a$, b and the c are variables. If there is a dollar sign after a variable, as in a$, it means this variable will refer to a word, or what we call in computer terms a 'string'. If there is just a letter or a combination of letters this is a *numeric* variable, which is to be set equal to a number. When the computer came to line 10, it waited for a 'string' input and this is where you entered your name. The B in line 20 meant it waited for a number. When you entered a number the computer assigned it to the variable B.

Line 30 did some assigning of its own. Using the LET statement, it made a equal to b plus b. When you use a LET command it makes what is on the *left* hand side of the equal sign – in this case the c – to be set equal to what is on the *right* hand side, in this case b plus b. So line 30 sets c to b plus b. When the computer comes to line 40, instead of printing out (as you see in the listing) 'hello a$' it prints up what a$ has been set equal to. In line 10, you'll recall, you assigned a$ to your name and so line 40 prints up 'hello' your name. Similarly in line 50. Instead of printing a b, it prints up the number that you gave it in line 20. And line 60 prints up the value of c as assigned by the LET statement in line 30, and the b as you gave it in line 20.

If this is all a bit confusing I suggest you rewind the tape just a bit and listen to the explanation again. When you run the program you will see on the screen the words hello, your name and then 'your number was' are printed directly underneath each other. There is a way of getting the computer to space out lines. If you look on the 7 key, you will see a single apostrophe printed in red. Using the EDIT key, bring line 50 down to the bottom of the screen and move across the word PRINT (using CAPS SHIFT and the 8 key) and put

the apostrophe before the open quote marks in that PRINT statement. Use SYMBOL SHIFT to get the apostrophe and put it so that line 50 now looks as it does in program D. Run the program again and you'll notice this time the computer puts a blank line between the 'hello', your name and the 'your number was' lines.

D



```
  10 INPUT a$
  20 INPUT b
  30 LET c=b+b
  40 PRINT "hello ";a$
  50 PRINT '"your number was ";b
  60 PRINT c;" is double ";b


hello tim

your number was 34
68 is double 34
```

The apostrophe available from the 7 key (using the red SYMBOL SHIFT key) moves the PRINT statement down one line. If you add three apostrophes before the words 'your number was', you'll find it prints three blank lines. We can use the apostrophe in this way to move PRINT statements down the screen but they are a little limited.

In a moment we'll talk about a more convenient way of moving PRINT statements about on the screen as you want them. But first I would like to introduce the word TAB which you'll find written in green above the P key. You will remember from the introduction of this tape that you get the words in green above a key by pressing both the shift keys at once then letting them go and then touching the relevant key. Try it now. Press both the CAPS SHIFT and the SYMBOL SHIFT at the same time, let them go, then press the P key and if all is well you will get the word TAB. Now use the 7 key and the CAPS SHIFT to move the 'greater than' sign up next to line 40. Using the EDIT key (the 1 and the CAPS SHIFT) bring the line 40 down to the bottom of the screen. Move across the word PRINT with the 8 key so the word TAB appears. Next press the 8 and then using the RED SYMBOL SHIFT touch the O key to get the semi-colon. Line 40 should now look like the line 40 in the program E in your booklet.

E

```
  10 INPUT a$
  20 INPUT b
  30 LET c=b+b
  40 PRINT TAB 8;"hello ";a$
  50 PRINT '"your number was ";b
  60 PRINT c;" is double ";b
```

```
         hello tim

your number was 342
684 is double 342
```

Once you have this, press the ENTER key so that the new line 40 moves up to its place within the program. Now run the program again. This time you will see that the word hello starts eight spaces across from the left hand side of the screen. The TAB command moves a PRINT statement across the number of spaces specified by the number which follows the word TAB.

It's time now for a completely new program. This will be the most complicated one we've put in the computer so far. Use the NEW key (the A key) to wipe the program you have in the SPECTRUM and then enter program F.

F

```
  10 INPUT "ink? ";a
  20 INPUT "paper? ";b
  30 INPUT "position across? ";c
  40 IF c<0 OR c>31 THEN GO TO 3
0
  50 INPUT "position down? ";d
  60 IF d<0 OR d>19 THEN GO TO 5
0
  70 PRINT AT d,c; INK a; PAPER
b;"x"
  80 GO TO 10
```

Have a look at program F in your booklet. There are a number of new features in it which I will explain before we actually enter the program. First, look at line 10. It has the word INPUT which we have seen before, then the word 'ink' with a question mark within the quote marks before the letter a. When you run this you'll see that the word 'ink?' will appear at the bottom of the screen telling you that that's the INPUT required. When words are included within quote marks in an INPUT statement, those words appear at the bottom of the screen. The same goes for lines 20 and 30. Note that there is a semi-colon after the close of the marks in those three lines. This is very important.

Line 40 is also a very important line. There are a number of new things we can learn from this line. 'IF THEN' are two words which are often seen together in computer programs. When the computer finds an IF statement it checks to see whether the statement is following the IF is true and if it is then it carries out whatever command follows the THEN.

The c in line 40 refers to the c in line 30 where you've been asked to input a position across. You enter a number which becomes set equal to the variable c in line 30. Line 40 looks at the value of this variable, and *if* it finds that it is less than zero, or greater than 31, *then* it sends action back to line 30. The end of line 40 is GO TO 30 and this is exactly what it does if the condition after IF is found to be true. So, line 40 checks to see whether c is less than zero or c is greater than 31. (You will find the OR is on the U key: you get it by holding down the RED SHIFT KEY and presing down the U.) The 'less than' sign you'll get from the R key and the 'greater than' from the T key. THEN is on the same key as GO TO. Line 50 uses the by now familiar words within quote marks in an INPUT statement. Line 60 follows the same sort of routine as line 40 did checking the value of D. This time it checks that D lies between zero and 19. This brings us to the end of side one of this cassette. We'll continue the explanation of program F on side 2.

Tab x = x spaties!!

' = "nieuwe regel"

SIDE 2

We will continue now the explanation of program F. We have gone as far as line 70. Line 70 starts off PRINT AT d, c. You will find the AT on the I key. You get AT by holding down the RED SYMBOL SHIFT and pressing the I. PRINT AT prints whatever follows the PRINT AT at the location specified by the two numbers or, as in this case, variables which follow the word AT. The first number or variable is the distance down the screen in lines and the second is the distance across in spaces. So, this prints whatever follows the words PRINT AT down d and across c. It does so in ink colour a. If you look at the top row of keys you'll see the numbers one to seven, and above them the colours blue, red, magenta, green, cyan, yellow and white.

in the same way

N/w INK 5, for example, will print in cyan, and INK 2 will print in red. So INK a equals the number between one and seven which you enter in line 10. PAPER is similarly controlled. PAPER is the background you print on, so PAPER b which you entered in line 20 will print it *underneath* INK a in printing the x that comes at the end of the line 70. You can chain together ~~many~~ control statements like this in a PRINT statement by using semi-colons. a colon

Line 80 uses GO TO which, as you'll remember, you get from the G key. This sends action back to line 10 to do it all over again. Described in a vacuum like this, this 'explanation' may not appear to illuminate the program very much. ~~However~~ But, if you enter it and run it for a while – seeing the result of various inputs – and then play the tape again, you'll find that this second run should make it much clearer. As with all examples on this tape, if you don't understand something at this stage, don't go on but run the tape back through and follow ~~through~~ it again. Once you have done this sevral times you should find even the more complex things become quite clear.

Time now to continue on with program G. There are a few new things in this which I'd like to tell you about before you enter the program.

G

```
 10 BORDER 0
 20 INK 2
 30 PAPER 6
```

```
 40 CLS
 50 INPUT "give me a number ";a
 60 INPUT "and another ";b
 70 IF a>1 AND b>1 THEN PRINT a
;" and ";b;" are both positive"
 80 PAUSE 200: GO TO 40
```

Line 40 is CLS which stands for CLEAR SCREEN. You'll find this on the V key. Have a look at it now. A few keys further along on the bottom row is the M key and on this you will find PAUSE. PAUSE appears in line 80 of our program. Notice after the number 200 in line 80 there is a colon. You will get this from the Z key using the SYMBOL SHIFT then pressing Z. In line 70 you will see the word AND. This comes from the Y key and you get it by holding down the RED SYMBOL SHIFT and pressing the key.

Line 70 is fairly complicated, so make sure you get all the semi-colons and spaces correct, or it may not be accepted by the computer. Once you have entered program G and run it for a while, come back to the tape and we'll go through it line by line.

Line 10, BORDER 0, turns the frame or borded around the screen black (the colour which is written above the 0 on the top right hand corner of the keyboard). Line 20 turns the INK, or the colour in which the computer writes, red, the colour which is above the 2 key. Line 30 turns the PAPER, the background colour, to yellow. This background is not visible until the computer hits line 40 (the clear screen command).

Notice the AND command in line 70. This ensures that only if a and b are *both* positive will that sentence be printed. Try running the program again this time giving a negative number and a positive number and you'll find that, as expected, the sentence 'a and b are both positive' does not appear. In line 80, PAUSE 200 holds the screen display for a while (in this case for about four seconds; you divide the PAUSE figure by 50 in the UK, 60 in the US, to get the number of seconds PAUSE operates for). The end of line 80 sends action back to line 40 using GO TO.

Line 80 is unusual in one respect. Up to now we have had 'single statements lines' like line 20. In line 80, however, there are two statements, PAUSE 200 and GO TO 40. You can have more than one statement on a line so long as each statement is separated by a colon (which you get, as you will recall, from the Z key using the RED SYMBOL SHIFT).

Now using CAPS SHIFT and the keys 6 and 7 get the little 'greater than' sign next to line 30. When you've done this, use the EDIT key to bring line 30 down to the bottom of the screen. Move over PAPER and the 6, using the 8 key, and then delete the 6. Now I want you to press both CAPS SHIFT and SYMBOL SHIFT at once. Then take your fingers off those keys and press the T key. The word RND should appear. Follow this with the multiplication sign (the asterisk you'll find on the B key which you get by holding down the RED SYMBOL SHIFT and pressing the B), and then add a 7. Line 30 should read now as it does in program H.

H

```
  10 BORDER 0
  20 INK 2
  30 PAPER RND*7
  40 CLS
  45 PAUSE 25: GO TO 30
  50 INPUT "give me a number ";a
  60 INPUT "and another ";b
  70 IF a>1 AND b>1 THEN PRINT a
;" and ";b;" are both positive"
  80 PAUSE 200: GO TO 40
```

Stop the tape until you have done this, and have also added the new line, 45. Then come back to the tape. Now that you have line 45 in place I'd like you to press the R key (RUN), and press ENTER. This rather splendid display of changing colours comes from line 30 choosing a colour at random. The RND in line 30 stands for random. it generates a random number between zero and one. I'll try to make that clear. Stop the program using BREAK, then press NEW and clear the computer completely. Not that you've done that stop the tape and enter program I, then come back to the tape.

I

```
  10 LET a=RND
  20 PRINT a
  30 GO TO 10
```

When you run program I you'll see a series of random numbers between zero and one generated by the computer.

```
0.16993713
0.74623108
0.96762085
0.57159424
0.87005615
0.25434875
.07699585
0.77574158
0.18086243
0.56561279
0.42144775
0.60923767
0.69326782
0.99543762
0.65782166
0.33700562
0.27616882
0.71348572
0.51174927
0.38174438
0.63153076
0.36521912
```

You'll notice that when the screen is full, the question 'scroll?' appears. Touching any key (except BREAK) will start the program running again, to print up some new random numbers. RND generates numbers between zero and one as you can see, but in many cases it is more useful to have numbers that are greater than one. This can be achieved very easily by changing line 10 to what it is in program J.

J

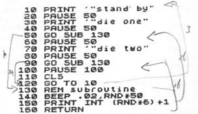
```
  10 LET a=INT (RND*10) +1
  20 PRINT a
  30 GO TO 10
```

Change line 10 so it looks like the line 10 from program J using INT (which you'll find above the R key). Press both SHIFT keys at once, then press the R key and the word INT will appear. Next hold down the RED SYMBOL SHIFT and touch the 8 key to get the left hand bracket. You get the RND as you recall by pressing both SHIFT keys, letting go and then pressing T. The asterisk (multiplication sign in BASIC) is found on the B key. You get it by holding down the RED SYMBOL SHIFT and then touching the B key. The right hand bracket you get from the 9 key using the RED SYMBOL SHIFT. The plus sign comes from the K key.

You may be beginning to think that you'll never find your way around the keyboard without an explanation of where every symbol and word is but you'll be surprised at how familiar you become with it after you have run through the tape a couple of times. Run program J and then come back to the tape for an explanation of it. Program J assigns, in line 10, the variable a to random numbers between one and ten. Note that if you didn't have the +1 at the end of that line the random numbers generated would be between zero and nine.

Now it is all very well to generate a great stack of numbers between one and ten but that may not seem at this moment particularly useful. However, random numbers can be used to simulate many things that happen in the real world such as the way raindrops fall on a particular area of ground; the way weeds grow in the garden or even the way dice fall. In fact, random numbers are used a lot in computing, particularly in games programs. Program K shows how the computer can be used to act as a pair of dice.

K



```
 10 PRINT '"stand by"
 20 PAUSE 50
 30 PRINT '"die one"
 40 PAUSE 50
 50 GO SUB 130
 60 PAUSE 50
 70 PRINT '"die two"
 80 PAUSE 50
 90 GO SUB 130
100 PAUSE 100
110 CLS
120 GO TO 10
130 REM subroutine
140 BEEP .02,RND*50
150 PRINT INT (RND*6) +1
160 RETURN
```

There is another new thing that is being taught by program K, the use of subroutines. Look at the listing of program K and you'll see that line 50 and line 90 both say GO SUB 130. You'll find the word GO SUB on the H key. When the computer comes across a GO SUB command, it goes to the line designated (in this case it goes to 130) then continues on until it comes to the word RETURN. In this case, it comes to the word RETURN in line 160. Then the computer returns to the line *after* the one which has sent it to the subroutine. When the computer came to line 50, it read GO SUB 130. Immediately it went to 130, carried out what it has to following the line numbers through in order until it came to line 160 when it struck the word RETURN. It then went back to line 60, the first line *after* the one which sent it to the subroutine, and then continued on through the program in order.

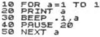
A subroutine is used if the same series of steps have to be executed several times throughout a program. It is a much more economical way of programming than including the same series of lines a number of times within a program. The only other things to notice in this particular program are the single apostrophe before the open quote marks in lines 10, 30 and 70 which space out the PRINT statements, and the BEEP command in line 140. We have used the BEEP once before in a rather restricted way in an earlier program. The BEEP command has two parameters (parameters are the numbers that follow the command). The first number, in this case .02, is the *duration* of the note. The decimal point is the same as the full stop (found on the M key using the RED SYMBOL SHIFT KEY). After the .02 (the duration), there is a comma (from the N key) and then another number for the pitch of the note. The first number after the word BEEP is the duration and the second number is the pitch. This can be from around −60 to +69. By putting in RND* 50 we get a different note every time the computer executes this subroutine.

dur, pitch

Line 150 generates a number between 1 and 6. Now when you're entering and running this program keep in mind that it is at this section of the course to illustrate two things. First of all, it's to show the production of random numbers which are used in two ways in the program, one to determine the pitch of the BEEP and the second to actually be the number that the computer is 'rolling' when it throws the dice. The second thing the program seeks to demonstrate is the use of subroutines. Run the program for a while then come back to the tape to continue on with this course.

If you look over the keys on your SPECTRUM at this point you'll probably be pleasantly surprised at how many of them you now recognise. You'll see that to program the SPECTRUM is perhaps somewhat simpler than you might have believed when you first began. Turn now to program L in your booklet.

L



```
10 FOR a=1 TO 15
20 PRINT a
30 BEEP .1,a
40 PAUSE 20
50 NEXT a
```

We are now going to look at FOR/NEXT loops. The word FOR is on the F key which makes it fairly easy to find, and the word NEXT is on the N key. If you look at line 10 in program L you will see it reads FOR A=1 to 15. The word TO is written in red on the F key (the same key as the word FOR) and you get it by pressing the RED SYMBOL SHIFT at the same time as you press the F key. Enter program L now.

A FOR/NEXT loop counts between the numbers specified in the initial FOR statement. In this case, the FOR/NEXT loop counts from one to 15 assigning the numbers 1-15 in turn to the variable a. This is shown by the fact that it prints out the numbers 1 to 15 using line 20 and in line 30 produces a series of rising tones. We can alter the FOR/NEXT loop as in program M to get tones at a different pitch.

M

```
10 FOR a=-15 TO 0
20 PRINT a
30 BEEP .1,a
40 PAUSE 20
50 NEXT a
```

Using edit, bring line 10 down to the bottom of the screen and change it so that it is the same as line 10 in program M. Run the program then come back to the tape for an explanation of it.

As you can see this program behaves much the same as L does except that the notes and the numbers start off somewhat lower in pitch. But the computer still counts up in ones, adding one each time it goes through the loop. The first time it goes through the loop a equals _15, the second time a equals 14 and so on until a equals zero. There is no reason why the computer should count only in ones or only upwards. Change line 10 using edit so that it reads the same as line 10 in program N.

Step: aantal "stappen" bij telln 2 : 2-4-6 3 : 2-5-8 etc. [illegible]

N

```
10 FOR a=48 TO 0 STEP -4
20 PRINT a
30 BEEP .1,a
40 PAUSE 20
50 NEXT a
```

You'll need the word STEP for this, and you'll find it on the D key (you get the word STEP by holding down the RED SYMBOL SHIFT and pressing the D key). When you run program N you will see that it counts downwards in jumps of four. This is what STEP does. STEP can be positive or negative and it determines the jump in counting in a FOR/NEXT loop.

Earlier on in this tape we discussed some of the ways you can effect the PRINToutput of the computer by changing the colour of the INK, or the colour of the PAPER. There are other commands that can be brought into play to add more life to the display generated by your SPECTRUM. On the bottom row of keys you will see the word FLASH underneath the V key and the word INVERSE underneath the N key. We are going to have a look at these words now.

O

```
  10 PRINT AT 0,0; FLASH 1;"test
"
  20 INPUT a$: IF a$="s" THEN ST
OP
  30 PRINT AT 0,0; FLASH 0;"test
"
  40 INPUT a$: IF a$="s" THEN ST
OP
  50 RUN
```

Enter program O. You'll see the word FLASH is used in lines 10 and 30. When you run it, you'll see the word TEST appear in the top left hand corner of the screen flashing off and on. When you press any key it will change to just sitting there–not flashing off and on. Press another key and it will start flashing again. Entering 'S' will stop the program. Run it for a while and then come back to the tape for an explanation of how it works.

Certain words such as FLASH and INVERSE are followed by a number which is either one or zero. One means 'on' and zero means 'off'. In line 10 of program O FLASH, 1 turns on the flashing which you can see quite clearly by its effect on the word TEST. Line 20 waits until you press the ENTER key before continuing. Line 30 prints the word TEST with flash turned off, that is FLASH 0, in the same position as the previous word 'test' was printed. So we see the word 'test' appear in a non-flashing, that is normal, condition. Line 40 waits for you to press ENTER again and then line 50 runs the program from the beginning. From this short program we can see how FLASH works.

Let's now look at the word INVERSE. Using the edit facility, change lines 10 and 30 to read as lines 10 and 30 in program P.

P

```
  10 PRINT AT 0,0; INVERSE 1;"te
st"
  20 INPUT a$: IF a$="s" THEN ST
OP
  30 PRINT AT 0,0; INVERSE 0;"te
st"
  40 INPUT a$: IF a$="s" THEN ST
OP
  50 RUN
```

Basically the changes only involve changing the word FLASH into the word INVERSE (which you will find underneath the M key). Run the program. This time you'll see that at first the word test printed back to front', that is white letters on a black background. Press ENTER. and it changes to normal black letters on a white background and this continues as long as you leave the program running. Stop it by entering 'S' before pressing ENTER.

By the way, you may have noticed that we've been using lower case (that is, small) letters. The SPECTRUM, unlike many other computers, does not differentiate between small and capital letters. If you like you can change lower case letters into capital ones by holding down the CAPS SHIFT KEY before you press the relevant letter key. Alternatively, if you want *all* the output to be in capitals you hold down the CAPS SHIFT and then press the 2 key (with the word CAPS LOCK above it). This will ensure that everything from that point on is in capitals. We will be using capital letters from now on for variable names but there's no reason why you can't leave them in lower case if you like.

vir julle

I would like to introduce now the word DIM. This may be the way you feel after you first hear the explanation, but don't worry–it will soon become clear. DIM is one of the more difficult things to explain which is why we have left it to this point on the tape. DIM is on the D key. You use DIM when you wish to set up an *array*, or *list* of numbers or words. We'll look at numbers first. Turn to program Q in the booklet.

Q

```
  10 DIM A(5)
  20 FOR B=1 TO 5
```

```
  30 INPUT A(B)
  40 NEXT B
  50 PAUSE 100
  60 FOR B=1 TO 5
  70 PRINT "A(";B;") IS ";A(B)
  80 NEXT B
```

The first line is the important one, DIM A(5). This tells the computer to reserve space for five numbers. We can refer to these numbers as A(1), A(2), A(3), A(4) and A(5). They are all *elements* in the A array. When you run the program you will see that line 10 *dimensions* the array to make space for these five elements, and then lines 20, 30 and 40 allow you to enter five numbers, one after another, pressing ENTER after each one. These are then stored as *elements* of the A array. Line 50 pauses for a few seconds and lines 60, 70 and 80 print out the elements of the array in a way which should help to make it clear how arrays work.

```
A(1) IS 23
A(2) IS 2342
A(3) IS -765
A(4) IS 1.0006
A(5) IS 44
```

You will find arrays of great use in programs, so if you find it hard to understand what they are from this quick explanation, I suggest you play the tape several times until it becomes clearer. You'll see arrays in use in some of the programs given at the end of the booklet. Now we've seen the use of an array to hold *numbers*. This is called a *numerical* array. You can also have an *alphanumeric*, or *string* array. You can see a string array set up in program R.

R

aantal woorde (of getalle)

maximale lengte.

```
  10 DIM A$(5,10)
  20 FOR B=1 TO 5
  30 INPUT A$(B)
  40 NEXT B
  50 PAUSE 100
  60 FOR B=1 TO 5
  70 PRINT "A$(";B;") IS ";A$(B)
  80 NEXT B
```

Notice that with a string array, not only do you have to have a dollar sign after the variable (so the dimension line reads DIM A$, or whatever letter between A and Z you decide to use), but also that the first number in brackets is the *number of elements* you are going to have in the array, and the second number is the *length* of the *longest element* in the array, that is the length of the longest word you wish to store in the array. Run program R now, entering five words in turn with ENTER after each one. Each word must be less than ten letters long. Notice that although you have to have a second number after the first in the dimension line (number 10 in line 10), when you specify the element of the array to be entered as in line 30, or to be printed as in line 70, you only need to specify the first number.

delta

Finally on this tape we'll look at READ, DATA and RESTORE. READ and DATA are very useful commands for accessing information during a program. They are often used for loading an array. Enter program S and run it and then come back to the tape for a discussion on what is going on.

S

```
 10 DIM A(3)
 20 DIM A$(3,5)
 30 FOR B=1 TO 3
 40 READ A(B)
 50 READ A$(B)
 60 NEXT B
 70 FOR C=1 TO 3
 80 PRINT A(C)
 90 PAUSE 30
100 PRINT A$(C)
110 PAUSE 30
120 NEXT C
130 DATA 99,"HELLO",4294
140 DATA "HEART",-32676,"BONES"
```

Line 10 dimensions a numeric array to hold three elements and line 20 sets up a string array to hold three elements up to five letters long. Lines 30 to 60 go through a loop reading from the DATA statements, lines 130 to 140.

Every time the computer comes across the word READ it goes to the next element in a DATA statement. You have to make sure–as in this case–that when it wants a number (as in line 40) it comes to a number in the DATA statement, and when it wants a string (as demanded by line 50) there is a string there. The computer will always READ the information from the DATA statement in order, but it doesn't matter where the DATA statements are in the program. If you doubt this, change line 130 to say line 17 and line 140 to line one or line 96. Then get rid then of lines 130 and 140 and run the program again. You'll find it performs in exactly the same way/ as it did before

The RESTORE command (which you will find above the S key) sends the computer back to the very start of the DATA data information. If you add a line such as 55 IF B=2 THEN RESTORE and then run the program, you'll see what happens when the DATA pointer is put back to the beginning of the first DATA statement.

That now brings us to the end of this tape. If there are some things which we've discussed, which still are not clear, play that section of the tape over and over again, until you understand what is going on.

You'll find a few additional commands explained at the end of the script of this talk, in the booklet. This is followed by a number of ready-to-run programs. I hope you enjoy entering them, and running them.

LEN

→ aantal letters of cijfers

The function LEN determines the length of a string, as you'll see by running the following program:

```
   10 REM LEN
   20 INPUT "ENTER A WORD ";A$
   30 IF A$="S" THEN STOP
   40 PRINT A$;" IS ";LEN A$;" LE
TTERS LONG"
   50 GO TO 20
```

VAL:

VAL returns the numeric equivalent of a string, which is shown by the next program:

```
   10 REM VAL
   20 INPUT "ENTER A SUM ";A$
   30 IF A$="S" THEN STOP
   40 PRINT A$;" EQUALS ";VAL A$
   50 GO TO 20

 22+4 EQUALS 26
 44/2-6 EQUALS 16
 1.33-5*6 EQUALS -28.67
```

INKEY$

INKEY$ reads the keyboard, and does not require you to press ENTER before acting on the keyboard input. This next short program shows INKEY$ in action. Enter 'S' to stop the program. Engage CAPS LOCK before you run it.

```
   10 REM INKEY$
   20 REM ENGAGE CAPS LOCK
   30 PAUSE 20
   40 LET A$=INKEY$
   50 IF A$="" THEN GO TO 40
   60 PRINT "YOU PRESSED ";A$
   70 IF A$="S" THEN STOP
   80 GO TO 40
```

The final program in this 'learning' section shows the use of CODE and CHR$. Each character (that is, each keyword, symbol, letter or number) printed by the computer has a CODE between zero and 255. The CODE of A is 65, and that of Z is 90. The command PRINT CODE A$ will produce the code of the first element of the string, A$. CHR$ works in the opposite way. Enter PRINT CHR$ 84 and the computer will print T, and so on. Run this program to see CODE and CHR$ in action.

```
   10 REM CHR$ and CODE
   20 INPUT "ENTER A LETTER ";A$
   30 PRINT '''"THE CODE OF ";A$;
" IS ";CODE A$
   40 IF A$="S" THEN STOP
   50 INPUT "ENTER A NUMBER BETWE
EN 65 AND   90 ";A
   60 IF A<65 OR A>90 THEN GO TO
50
   70 PRINT '''"THE CHR$ WITH COD
E ";A;" IS ";CHR$ A
   80 GO TO 10
```

chr$ → de letters
code → de cijfercode!!

PROGRAMS

Games programs:

- MAZE-MAN
- CHECKERS
- CAMEL
- THE MUSIC MASTER
- KNIGHTSBRIDGE
- CATERKILLER
- POLYMONY
- CONNECTEZ-VOUS QUATRE
- MOGUL
- COURTBALL DODGE
- HIGH ROLLER
- ECONOMY
- REACTION TEST
- DRAGONFIRE
- STOCK MARKET
- ANAGRAMS
- NOUGHTS AND CROSSES
- BOMB RUN
- DRIVER
- SPECTRAL BOUNCER
- BOMBER
- MANTRAP

Utility programs:

- DECISION-MAKER
- BILLBOARD
- SUPER SKETCH
- CALENDAR
- USER-DEFINED CHARACTER AID

Graphics demonstrations:

- DIAMOND GLORY
- SPARTAN SPANGLES
- SPIRALS
- INSTANT BLOPPO

GAMES PROGRAMS

MAZE-MAN

MAZE-MAN is a Spectrum version of the Arcade favourite PAC-MAN, and despite being written in BASIC, still provides quite a challenge. MAZE-MAN was written by Tudor Costigan.

```
Program Summary:
Lines 0010-0070:Initialisation
  of variables, etc. 'n'=High
  Score, 'b'=Score. Poke 23692,
  255 cures the computer of
  incessantly asking 'Scroll?'
  every so often. The subroutine
  at line 4300 draws the design
  shown at the beginning of the
  program:
Lines 0090-0185:Print the title
  and ask for 'Skill Level',
  with suitable error messages.
  'ww' is used as a delay in a
  loop later in the program.
Lines 0190-0210:Define User-
  Defined graphics, using data
  statements at lines 3700
  onwards. However, in the
  these graphics come out as the
  capital letter of the graphic
  that they are (eg. A,B,etc.)
Lines 0230-0440:Supply instruct-
  ions if required.
Lines 0510-1630:Setup position
  of 'Ghost' in 'l' and 'k' as
  being 6,19, and then print out
  board. Originally, each
  "Character position" of the
  board had a different data
  element, but this was too
  slow. The design of the board
  has already been tested for
  about a year at my school, and
  so far no complaints have been
  recieved. So I decided to
  incoperate the design into my
  program.
   Even in the listing the
  individual elements of the
  data statements are of
  different brightnesses. This
  is because they are, in actual
  fact, different colours, and
  the printer, being only 'black
  and white' has to show these
  colour changes as different
  shades of grey.
   The 'CHR$ 17;CHR$ 8' in line
```

```
  1420 allows the old attribute
  of the paper to show through,
  as described on pages 111 and
  114 of the Spectrum manual.
   Lines 1550-1610 print the
  title and scores in varying
  degrees of noticeability. Line
  1620 prints the amount of men
  left in the middle of the
  board. Line 1630 sets the
  paper colour to green, the
  background colour of the
  insides of the maze.
Line 1710:initialises your
  position as (16,20) and prints
  you at that position.
Line 1750:Delay according to
  skill level selected (see line
  0090.
Line 1755:Provides higher note
  in the continual sound
  effects. It has to be this
  short, because the Spectrum
  cannot 'BEEP' and do something
  else at the same time.
Lines 1760-1788:Obtain input if
  a key is being pressed and
  process it. 'd$' and 'x$' are
  used in conjunction with each
  other in order to provide
  continuous movement without
  having to keep the relevant
  key pressed down. Line 1771
  blanks out the space where the
  pacman was with green (see
  line 1630). If an unusable key
  was being pressed, then line
  1788 sends the Spectrum to
  line 2160, the routine for
  moving the alien.
Lines 1800-1875:These test the
  next position in the
  appropriate direction as to
  whether it is a space or not.
  If it is, then the pacman is
  moved to that position and
  control passes to line 2100,
  which, as explained later,
  checks for certain special
  occaisions before going on to
  line 2160, to move the ghost.
  If it is not, control is
  passed to another routine to
  find out what it is:this
  routine is explained below.
Lines 1900-2070:These routines
  check the space mentioned
  above and decide whether it is
  a star (worth 10 points), a
  half-bonus (worth 100 points),
```

a bonus (worth 200 points), or a wall (worth nothing!). For the point-gaining objects the score 'b' and the counter 'i' are increased (when 'i' reaches 186, all the stars have been eaten and all the bonus's have appeared). If a bonus of either kind has been 'eaten', then appropriate sound effects are produced. If the object is a wall; the pacman's position is unaltered and he is reprinted on the same spot.

Lines 2103-2106: A check to see if the pacman has gone down either tunnel, and if so to bring him out on the other side of the board.

Lines 2110-2120: Print or delete the bonus according to how many stars are eaten.

Lines 2130-2135: Print the score on white background, and alter the high-score if the score is higher.

Lines 2140-2149: Tests for a completely eaten-up board, and if positive prints a congratulary message and plays a tune, then going to line 510, the beginning of the main program.

Line 2161: Sounds the lower note of the continuous sound effects (see line 1755).

Lines 2165-2190: Test for possible movement towards the pacman, and accordingly goes to a subroutine. If no movement in the necessary direction is possible, then the Spectrum goes to line 2650, which checks to see if the ghost and the pacman are on the same spot.

Lines 2210-2550: Subroutines to check proposed position for a wall. If the test is postive, then that option is turned down, and the Spectrum is sent to check if another direction is possible. If there is no wall, the Spectrum goes to line 2610, explained below.

Lines 2610-2650: Check position and assign value to 'j'. This is so that the object is

remembered and can be replaced when the ghost has passed by. However, the bonus will not be replaced. The subroutine at line 2700 is called just the value of 'j' is reassigned to replace the previous object, and to print the ghost in its new position. Control then goes to line 1750, which checks for another input from the player. If the pacman and ghost are on the same spot, as tested by line 2650, the Spectrum is sent to line 2710.

Lines 2710-2770: Decrease the no. of lives left, and reset counter (but not score). Colour and sound effects are combined, and if the player still has a life left then the Spectrum is sent to line 510.

Lines 2780-2840: End game logic. Another game can be had, and your score is printed, and you are told if you have beaten the high score. If you wish to immortalise your high score, you should replace the figure in line 10 to your own. Then every time the game is run from the beginning, your high score will appear. If another game is wanted, then control goes to line 20.

Lines 3700-3750: Data statements to define the User-Defined graphics used (see lines 190-210).

Lines 4300-4420: Print the title design (see line 60).

```
Variables used:

n                    =High score
n$                   High score
j                    'CODE' of object
                     'underneath'
                     ghost
kk                   no.of lives left
i                    counter of
                     objects eaten
                     this 'frame',
b                    score
d$                   stores previous
                     key pressed
u                    loop variable
c$                   general input
                     variable
q,z                  loop variables
d                    loop variable
e$                   stores data for
                     board-printing
s                    loop variable
x,y                  position of
                     pacman
k,l                  position of
                     ghost
ss,sd                contents of
                     proposed
                     position
q$                   score+colour
                     control
aa,bb,cc,dd          length of notes
qa,qb                added to k,l
                     produce old
                     position
a                    loop variable
a$                   input variable
x,y                  loop variables
```

```
   1 REM MAZE MAN
   2 REM BY T.M.COSTIGAN
  10 LET n=47870: LET n$=STR$ n:
 POKE 23692,255: LET rc=0
  20 LET j=32: LET kk=3: LET i=0
: LET b=0: LET d$=""
  30 BORDER 6: PAPER 7: INK 2: C
LS
  60 GO SUB 4300: FOR u=0 TO 5:
PRINT : NEXT u
  70 INK 0
  90 PRINT FLASH 1; INK 1;"    MA
ZE MAN - By T.M.Costigan     "
 100 PRINT : PRINT
 110 PRINT "Skill Level?"
 120 PRINT "   Hard,Medium,Easy (
H,M,E) "
 130 PAUSE 0
 140 LET c$=INKEY$
 145 IF INKEY$<>"" THEN GO TO 14
5
 150 IF c$<>"h" AND c$<>"H" AND
c$<>"m" AND c$<>"M" AND c$<>"e"
AND c$<>"E" THEN PRINT AT 11,0;"
 H,M,or E only please!": GO TO 1
30
 160 IF c$="h" OR c$="H" THEN LE
T ww=0
 170 IF c$="m" OR c$="M" THEN LE
T ww=25
 180 IF c$="e" OR c$="E" THEN LE
T ww=50
 185 PRINT AT 10,27;c$
 190 RESTORE 3700
 200 FOR q=1 TO 6: READ x$
 205 FOR z=0 TO 7: READ x: POKE
USR x$+z,x: NEXT z
 210 NEXT q
 220 IF rc<>0 THEN GO TO 450
 230 PRINT : PRINT "Do you wish
instructions (Y/N)"
 240 IF INKEY$="" THEN GO TO 240
 250 LET c$=INKEY$
 260 IF c$="n" OR c$="N" THEN GO
 TO 450
 270 IF c$<>"y" AND c$<>"Y" THEN
 GO TO 240
 290 CLS
 300 PRINT : PRINT BRIGHT 1;"
   INSTRUCTIONS "
 310 PRINT : PRINT "To move up",
"press 'a'"
 320 PRINT "To move down","press
 'z'"
 330 PRINT "To move right","pres
s 'm'"
 340 PRINT "To move left","press
 'n'"
 360 PRINT : PRINT " · = 10 poin
ts"
```

```
 370 PRINT " O = 100 points"
 380 PRINT " ! = 200 points"
 430 PRINT : PRINT FLASH 1;"
PRESS ANY KEY TO CONTINUE    "
 435 IF INKEY$<>"" THEN GO TO 43
5
 440 IF INKEY$="" THEN GO TO 440
 510 CLS : LET l=6: LET k=19
 530 RESTORE
 650 DATA "#####################
#"
 660 DATA "#********************
#"
 670 DATA "#*###*###*##*###*###*
#"
 680 DATA "#O###************###O
#"
 690 DATA "#*****#*######*#*****
#"
 700 DATA "#*###*#********#*###*
#"
 710 DATA "#*******######*******
#"
 720 DATA "###*#*#*#    #*#*#*##
#"
 730 DATA "   *#***#    #***#*  
"
 740 DATA "###*#*#*#    #*#*#*##
#"
 750 DATA "#*****#*######*#*****
#"
 760 DATA "#*#*#***** ******#*#*
#"
 770 DATA "#O#*#*##*####*##*#*#O
#"
 780 DATA "#*#*#************#*#*
#"
 790 DATA "#******########******
#"
 800 DATA "#*####*########*####*
#"
 810 DATA "#*####*########*####*
#"
 820 DATA "#********************
#"
 830 DATA "#####################
#"
1390 PAPER 7: CLS : PAPER 4: INK
 1
1400 FOR d=1 TO 19
1410 READ e$: FOR s=1 TO 22: IF
e$(s)="#" OR e$(s)=" " THEN PAPE
R 7: INK 4
1412 IF d=12 AND s=11 THEN PAPER
 4
1415 PRINT AT d,8+s,e$(s),: PAPE
R 4: INK 1
1420 NEXT s: PRINT CHR$ 17;CHR$
8
1480 NEXT d
```

```
1490 PAPER 7: INK 2
1550 PRINT AT 1,0; FLASH 1;"Maze
 man": INVERSE 1;AT 2,3;"By"; IN
VERSE 0;AT 3,1;"T.M.C."
1600 PRINT AT 5,0; INVERSE 1;"HI
 SCORE";AT 6,1; INVERSE 0; PAPER
 7;n$
1610 PRINT AT 8,1; INVERSE 1;"SC
ORE";AT 9,1;STR$ b
1620 PRINT AT 9,19;STR$ kk;"ᗧ"
1630 PAPER 4
1660 FOR s=0 TO 200: NEXT s
1710 LET y=18: LET x=20: PRINT A
T y,x;"●"
1750 FOR u=0 TO ww: NEXT u
1755 BEEP .02,4
1760 LET x$=INKEY$
1765 IF x$="" THEN LET x$=INKEY$
1770 IF x$<>"" THEN LET d$=""
1771 PRINT AT y,x;" "
1780 IF x$="m" OR x$="M" OR d$="
m" OR d$="M" THEN GO TO 1800
1782 IF x$="n" OR x$="N" OR d$="
n" OR d$="N" THEN GO TO 1820
1784 IF x$="a" OR x$="A" OR d$="
a" OR d$="A" THEN GO TO 1840
1786 IF x$="z" OR x$="Z" OR d$="
z" OR d$="Z" THEN GO TO 1860
1788 GO TO 2160
1800 LET x=x+1
1810 LET ss=CODE (SCREEN$ (y,x))
: IF ss<>32 THEN GO TO 1900
1812 PRINT AT y,x;"ᗧ"
1815 GO TO 2100
1820 LET x=x-1
1830 LET ss=CODE (SCREEN$ (y,x))
: IF ss<>32 THEN GO TO 1950
1832 PRINT AT y,x;"ᗤ"
1835 GO TO 2100
1840 LET y=y-1
1850 LET ss=CODE (SCREEN$ (y,x))
: IF ss<>32 THEN GO TO 2000
1852 PRINT AT y,x;"ᗢ"
1855 GO TO 2100
1860 LET y=y+1
1870 LET ss=CODE (SCREEN$ (y,x))
: IF ss<>32 THEN GO TO 2050
1872 PRINT AT y,x;"ᗣ"
1875 GO TO 2100
1900 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1812
1905 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1812
1910 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1812
1915 IF ss=35 THEN LET x=x-1: GO
 TO 1812
1920 GO TO 1812
1950 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1832
```

```
1955 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1832
1960 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1832
1965 IF ss=35 THEN LET x=x+1: GO
 TO 1832
1970 GO TO 1832
2000 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1852
2005 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1852
2010 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1852
2015 IF ss=35 THEN LET y=y+1: GO
 TO 1852
2020 GO TO 1852
2050 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1872
2055 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1872
2060 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1872
2065 IF ss=35 THEN LET y=y-1: GO
 TO 1872
2070 GO TO 1872
2100 IF x$<>"" THEN LET d$=x$
2103 IF y=9 AND x=31 THEN PRINT
AT y,x;" ": LET x=9: PRINT AT 9,
9;"●"
2106 IF y=9 AND x=8 THEN PRINT A
T y,x;" ": LET x=30: PRINT AT 9,
30;"●"
2110 IF i=50 OR i=100 OR i=150 T
HEN PRINT AT 12,19; FLASH 1; PAP
ER 7;"!": LET i=i+1
2120 IF (i=60 OR i=110 OR i=160)
 AND (y<>12 AND x<>19) AND (l<>1
2 AND k<>19) THEN PRINT AT 12,19
;" "
2130 LET q$=CHR$ 17+CHR$ 7+STR$
b+CHR$ 17+CHR$ 4: PRINT AT 9,1;
INK 0;q$
2135 IF b>n THEN PRINT AT 6,1; I
NK 0;q$
2140 IF i=166 THEN PRINT AT 8,15
; BRIGHT 1; FLASH 1; PAPER 7;"WE
LL  DONE": LET I=0: GO TO 2142
2141 GO TO 2150
2142 LET aa=.125: LET bb=.25: LE
T cc=.375: LET dd=.5
2143 FOR z=1 TO 2
2145 BEEP cc,0: BEEP aa,-4: BEEP
 dd,-4: BEEP dd,0: BEEP cc,-4: B
EEP aa,-4: BEEP dd,-4: BEEP cc,0
: BEEP aa,0: BEEP dd,2: BEEP dd,
-1: BEEP bb,0: BEEP bb,4: BEEP 1
,0
2146 NEXT z
2148 BEEP cc,-1: BEEP aa,0: BEEP
 dd,2: BEEP dd,2: BEEP bb,4: BEE
```

```
P bb,0: BEEP dd,2: BEEP dd,2: BE
EP cc,-1: BEEP aa,0: BEEP dd,2:
BEEP dd,2: BEEP bb,4: BEEP bb,0:
 BEEP dd,2: BEEP dd,2: BEEP cc,0
: BEEP aa,-1: BEEP dd,-4: BEEP d
d,2: BEEP dd,0: BEEP dd,2: BEEP
dd,4: BEEP cc,0: BEEP aa,-1: BEE
P dd,-4: BEEP bb,5: BEEP bb,2: B
EEP dd,0: BEEP dd,-1: BEEP dd,0
2149 GO TO 510
2150 REM MOVING ALIEN
2160 PRINT AT y,x;"●"
2161 BEEP .02,0
2165 IF l-y>0 THEN GO TO 2410
2170 IF l-y<0 THEN GO TO 2510
2180 IF k-x>0 THEN GO TO 2210
2190 IF k-x<0 THEN GO TO 2310
2200 GO TO 2650
2210 LET k=k-1
2220 LET qa=1: LET qb=0
2240 LET sd=CODE (SCREEN$ (l,k))
2250 IF sd=35 THEN LET k=k+1: GO
 TO 2190
2260 GO TO 2610
2310 LET k=k+1
2320 LET qa=-1: LET qb=0
2340 LET sd=CODE (SCREEN$ (l,k))
2350 IF sd=35 THEN LET k=k-1: GO
 TO 1750
2360 GO TO 2610
2410 LET l=l-1
2420 LET qa=0: LET qb=1
2440 LET sd=CODE (SCREEN$ (l,k))
2450 IF sd=35 THEN LET l=l+1: GO
 TO 2170
2460 GO TO 2610
2510 LET l=l+1
2520 LET qa=0: LET qb=-1
2540 LET sd=CODE (SCREEN$ (l,k))
2550 IF sd=35 THEN LET l=l-1: GO
 TO 2180
2610 IF sd=32 THEN GO SUB 2700:
LET j=32: GO TO 1750
2620 IF sd=42 THEN GO SUB 2700:
LET j=42: GO TO 1750
2630 IF sd=79 THEN GO SUB 2700:
LET j=79: GO TO 1750
2640 IF sd=33 THEN GO SUB 2700:
LET j=32: GO TO 1750
2650 IF l=y AND k=x THEN GO SUB
2700: GO TO 2710
2660 GO TO 1750
2700 PRINT AT l+qb,k+qa;CHR$ j:
PRINT AT l,k;"●": RETURN
2710 LET kk=kk-1
2720 LET i=0
2730 FOR a=0 TO 7
2740 BEEP .02,0: BORDER a: BEEP
.02,1: PAPER 7-a: BEEP .02,4: CL
S
```

```
2750 NEXT a
2760 BORDER 6: BEEP .05,0: PAPER
 7: BEEP .05,1: INK 2: BEEP 2,3:
 CLS
2770 IF kk>0 THEN GO TO 510
2780 PRINT "You ended with ";b;"
 points."
2790 IF n<b THEN LET n=b: PRINT
"This is the new high score"
2800 PRINT ' FLASH 1; BRIGHT 1;"
      ANOTHER GAME?
"
2810 LET e$=INKEY$
2820 IF e$<>"y" AND e$<>"Y" AND
e$<>"n" AND e$<>"N" THEN GO TO 2
810
2830 LET rc=1: IF e$="y" OR e$="
Y" THEN GO TO 20
2840 STOP
3700 DATA "a".0.60.126.14.14.126
.60.0
3710 DATA "e".0,60.126,126.126,1
26.60.0
3720 DATA "f",0,60,126,126,126,8
4.84.0
3730 DATA "b",0,60,126,112,112,1
26,60,0
3740 DATA "c",0,36,102,102,126,1
26,60,0
3750 DATA "d",0,60,126,126.102,1
02,36,0
4300 FOR x=0 TO 255 STEP 5
4310 PLOT 127,87: DRAW x-127.88
4320 NEXT x
4330 FOR y=175 TO 0 STEP -5
4340 PLOT 127.87: DRAW 128,y-87
4350 NEXT y
4360 FOR x=255 TO 0 STEP -5
4370 PLOT 127,87: DRAW x-127.-87
4380 NEXT x
4390 FOR y=0 TO 175 STEP 5
4400 PLOT 127,87: DRAW -127,y-87
4410 NEXT y
4420 RETURN
```

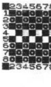
CHECKERS

In this version of the well-known board game, the Spectrum is playing down the screen from the top of the board, and you are playing from the bottom. The computer makes its multiple jumps automatically, and you are asked (see line 2125) if you can make another capture after you've taken one of the computer's pieces. The display is particularly effective on a colour television, and you'll see that the computer plays reasonably well, if a little slowly. Its end game is not strong, and you may wish to add a routine to give it a little more 'intelligence' in this area. This program is based on a Checkers game written by Graham Charlton, and subsequently modified by Tim Hartnell and Tim Rogers. You move by entering the number down the side then the number across of the piece you wish to move, then the number down and number across of the square you want to move into.

```
   5 REM CHECKERS
  10 INK 6
  20 BORDER 0
  30 PAPER 0
  35 POKE 23658,8: REM CAPS LOCK
  40 CLS
  45 GO SUB 7000
  50 GO SUB 5000
  60 GO SUB 4300
  65 GO TO 200
  70 LET r=CODE b$(b)
  80 LET s=CODE b$(b+d(x))
  90 LET t=CODE b$(b+2*d(x))
 100 RETURN
 200 LET z=0
 210 PRINT #1;AT 1,9; FLASH 1;
   "THINKING"
 215 FOR b=23 TO 86
 220 FOR x=1 TO 4
 230 GO SUB 70
 240 IF ((x<3 AND r=88) OR r=36)
     AND (s=79 OR s=96) AND t=32
     THEN GO TO 3000
 250 NEXT x
 260 NEXT b
 300 FOR a=1 TO 400
 310 LET b=INT (RND*80)+23
 320 FOR x=1 TO 4
 330 GO SUB 70
 340 IF ((x<3 AND r=88) OR r=36)
 AND s=32 THEN GO TO 4000
 350 NEXT x
 360 NEXT a
 370 PRINT AT 18,0;"You win"
 380 GO TO 370
2000 PRINT #1;"From? ";
```

```
2005 LET g=0
2010 GO SUB 6000
2020 LET g=i
2030 PRINT #1;AT 1,16;"To? ";
2040 GO SUB 6000
2050 LET h=i
2060 LET b$(12+h)=b$(12+g)
2070 IF h<20 THEN LET
     b$(h+12)="£"
2080 LET b$(g+12)=" "
2090 LET v=ABS (h-g)
2095 IF v=18 OR v=22 THEN LET b$
     (12+(h+g)/2)=" ":
     GO TO 2110
2100 GO SUB 4300
2105 GO TO 210
2110 INPUT ""
2120 GO SUB 4300
2122 BEEP .1,10
2125 PRINT #1;AT 1,0;
     "Can you take again (Y/N)?"
2130 IF INKEY$<>"N" AND INKEY$<>
     "Y" THEN GO TO 2130
2135 INPUT ""
2140 IF INKEY$="Y" THEN
     GO TO 2000
2150 GO TO 210
3000 LET b$(b+2*d(x))=CHR$ r
3010 LET b$(b)=" "
3020 LET b$(b+d(x))=" "
3030 LET b=b+2*d(x)
3040 IF b>93 THEN LET b$(b)="$":
     GO SUB 4300: GO TO 2000
3500 FOR x=1 TO 4
3510 GO SUB 70
3520 IF ((x<3 AND r=88) OR r=36)
     AND (s=79 OR s=96) AND t=32
     THEN GO TO 3000
3530 NEXT x
3540 GO SUB 4300
3550 GO TO 2000
4000 LET b$(b+d(x))=CHR$ r
4010 LET b$(b)=" "
4020 IF b+d(x)>93 THEN
     LET b$(b+d(x))="$"
4030 GO SUB 4300
4040 GO TO 2000
4300 INPUT ""
4305 PRINT AT 6,8;
4310 FOR a=13 TO 112
4315 IF b$(a)<>"X" AND b$(a)<>
     "O" AND b$(a)<>"£" AND
     b$(a)<>"$" THEN PRINT
     b$(a);: GO TO 4330
4320 PRINT INK (2*((b$(a)="O" OR
     b$(a)="£"))+5*((b$(a)="X"
     OR b$(a)="$"))); FLASH (b$
     (a)="£" OR b$(a)="$");"■";
4330 IF (a-12)/10=INT ((a-12)
     /10) THEN PRINT TAB 8;
```

```
4340 NEXT a
4350 PRINT '
4360 RETURN
5000 DIM d(4)
5010 LET d(1)=11
5020 LET d(2)=9
5030 LET d(3)=-9
5040 LET d(4)=-11
5050 READ b$
5060 LET z=1
5070 RANDOMIZE
5080 RETURN
6000 LET n$=""
6010 LET i$=INKEY$
6020 IF CODE i$<49 OR CODE i$>57
     THEN GO TO 6010
6030 BEEP .1,20
6040 LET n$=n$+i$
6050 PRINT #1;AT 1,10+16*SGN g;
     n$
6060 IF INKEY$<>"" THEN
     GO TO 6060
6070 IF LEN n$<2 THEN GO TO 6010
6080 LET i=VAL n$
6090 RETURN
7000 FOR a=0 TO 7
7010 READ b
7020 POKE USR "a"+a,b
7030 NEXT a
7040 RETURN
7050 DATA 0,60,126,102,102,126,
     60,0
9000 DATA "AAAAAAAAAAAA█23456789
█1█X█X█X█X12X█X█X█X█23█X█X█X█X34
█ █ █ █ 45█ █ █ █ █ 560█0█0█0█67█0
█0█0█0780█0█0█0█8█23456789█AAAAA
AAAAAAA"
```



CAMEL

In CAMEL, written by Chris Callender and modified by Peter Shaw, you have to try and cross a desert. If you're afraid of snakes and other things that creep and crawl, this game is not for you. If Lady Luck rides with you, you'll accumulate treasure and other goodies. Here's a key to the graphics characters used:

```
┌ - graphic a
┐ -graphic b
└ -graphic c
┘ -graphic d
▐ -graphic e
─ -graphic f
🐪 -graphic h
🯅 graphic g
```

```
  10 REM CAMEL
  20 REM CHRIS CALLENDER
         PETER SHAW
  25 LET t$=""
  30 GO SUB 9000
  40 PRINT TAB 10;"┌───────┐"'TAB
10;"▐CAMEL▐"'TAB 10;"└───────┘"
  50 INPUT "Hello, What is your
name ? "; LINE n$
  60 PRINT '''"Hello ";n$
  70 LET w=100: DIM h$(40,15): D
IM p$(10,15): DIM p(40,2): DIM s
$(22,32)
  80 FOR a=1 TO 40: LET p(a,1)=I
NT (RND*36)+2: LET p(a,2)=INT (R
ND*22)+2: NEXT a
  90 FOR a=1 TO 10: READ p$(a):
NEXT a: LET x=INT (RND*26)+2: LE
T y=INT (RND*18)+2
 100 FOR a=1 TO 40: LET h$(a)=p$
(INT (RND*10)+1): NEXT a
 110 LET s$(1)="┌───────────────
───────────────┐"
 120 LET s$(22)="└──────────────
─────────🐪────┘"
 130 FOR a=2 TO 21
 140 LET s$(a)="▐
              ▐": NEXT a
```

```
 150 LET b$="▐": FOR a=2 TO x-1:
 LET b$=b$+" ": NEXT a: LET b$=b
$+"🯅": FOR a=x+1 TO 31: LET b$=b
$+" ": NEXT a
 160 LET b$=b$+"▐"
 170 LET s$(y)=b$
 180 CLS : FOR a=1 TO 22: PRINT
INK 6;s$(a): NEXT a
 190 INPUT "Direction ";d$
 200 LET c$=d$(2 TO ): LET c=VAL
 c$
 205 IF c>10 THEN GO TO 120
 210 LET d$=d$(1)
 220 IF d$="n" THEN LET y=y-c
 230 IF d$="s" THEN LET y=y+c
 240 IF d$="e" THEN LET x=x+c
 250 IF d$="w" THEN LET x=x-c
 260 LET a=1
 270 IF ABS (x-p(a,1))<2 AND ABS
 (y-p(a,2))<2 THEN GO TO 300
 280 LET a=a+1: IF a<41 THEN GO
TO 270
 290 IF x=29 AND y=22 THEN GO TO
 7000
 292 IF w<10 THEN PRINT "Get wat
er fast": FOR a=1 TO 1000: NEXT
a
 294 LET w=w-1: IF w=0 THEN GO T
O 6000
 295 GO TO 110
 300 LET k$=h$(a)
 310 LET a=1
 320 IF k$=p$(a) THEN GO TO 340
 330 LET a=a+1: GO TO 320
 340 GO TO (((a*5)*200)/2)+500
 999 STOP
1000 PRINT "You end up in a sand
storm. You  cannot see and wande
r to"
1020 LET x=INT (RND*30)+2: LET y
=INT (RND*20)+2
1030 PRINT "(";x;",";y;")"
1040 GO TO 110
1500 PRINT "Oh dear there is a s
nake here,  I hate snakes!"
1510 PRINT "Hit it please ": LET
 h$(a)=""
1520 LET a=INT (RND*5): IF a=1 T
HEN GO TO 1560
1530 PRINT "Oh no! it bit you"
1540 INPUT "Another game ";g$: I
F g$(1)="y" THEN RUN
1550 STOP
1560 PRINT "You were very lucky
there.        Be careful!"
1570 FOR a=1 TO 1000: NEXT a: GO
 TO 110
2000 PRINT "Great-Here is some w
ater I think"
2010 IF RND>.5 THEN GO TO 2060
```

```
2020 PRINT '"Yes it is!, Take 10
0 pints"
2030 LET w=w+100
2040 GO TO 2070
2050 GO TO 9999
2060 PRINT "No, it was a mirage.
 sorry!"
2070 FOR a=1 TO 1000: NEXT a: GO
 TO 110
2500 PRINT "Here are the long lo
st Jewels"
2510 LET t$=t$+"The long lost je
wels"+CHR$ 13
2520 LET h$(a)="": FOR a=1 TO 10
00: NEXT a
2530 GO TO 110
3000 PRINT "Good ";n$;" Here is
a camel"
3010 PRINT "Where do you want it
 to take youto"
3020 INPUT x,y
3030 LET h$(a)="": GO TO 110
3500 FOR a=1 TO 61: PRINT INK (R
ND*7)+1;"TIME WARP ";: NEXT a
3510 GO TO 1540
4000 PRINT "Guess what ";n$;" he
re is a      computer! It will he
lp you find any of these"
4020 FOR a=1 TO 10: PRINT FLASH
1;a; FLASH 0,p$(a): NEXT a
4030 INPUT "Which one do you wan
t ";z
4040 LET a=1
4050 IF h$(a)=p$(z) THEN GO TO 4
070
4055 LET a=a+1: IF a<41 THEN GO
TO 4070
4060 PRINT "There is no ";p$(z):
 GO TO 4020
4070 PRINT "Try:-(";p(a,1);",";p
(a,2);")"
4080 FOR a=1 TO 750: NEXT a: GO
TO 110
4500 PRINT "Here is a box of gol
d ";n$
4510 PRINT "good!": LET t$=t$+"a
 box of gold"+CHR$ 13
4520 FOR a=1 TO 750: NEXT a: GO
TO 110
5000 PRINT "Oh dear, here is a s
corpion      ";n$
5010 LET h$(a)="": IF RND>.5 THE
N GO TO 1530
5020 PRINT "Good it didn't bite
you": FOR a=1 TO 750: NEXT a: GO
 TO 110
5500 PRINT "I see a water thief"
5510 LET h$(a)=""
5520 LET a=INT (RND*3): IF a=1 T
HEN PRINT "You overpowered him":
 FOR a=1 TO 750: NEXT a: GO TO 1
10
```

```
5530 LET b=INT (RND*w): PRINT "H
e stole ";b;" pints": LET w=w-b
5540 FOR a=1 TO 750: NEXT a: GO
TO 110
6000 PRINT "You died of thirst "
;n$
6010 PRINT "did you know that?!"
6030 GO TO 1540
7000 PRINT "You made it out of t
he desert   alive"
7010 IF t$="" THEN GO TO 7040
7020 PRINT "And managed to take:
-"
7030 PRINT t$
7040 PRINT "Prepare to take off!
": GO TO 1540
9000>BORDER 0: PAPER 0: INK 7: C
LS : FOR U=USR "a" TO USR "i"-1
9010 READ U1: POKE U,U1
9030 NEXT U
9040 DATA 0,0,0,31,31,24,24,24
9050 DATA 0,0,0,248,248,24,24,24
9060 DATA 24,24,24,31,31,0,0,0
9070 DATA 24,24,24,248,248,0,0,0
9080 DATA 24,24,24,24,24,24,24,2
4
9090 DATA 0,0,0,255,255,0,0,0
9100 DATA 8,28,8,62,8,8,20,34
9110 DATA 24,24,60,126,255,24,24
,60
9120 RETURN
9130 DATA "a sandstorm","a snake
","water","the lost jewels","a c
amel","a time warp"
9140 DATA "a computer","a box of
 gold","a scorpion","water thiev
es"
9999 STOP
```

THE MUSIC MASTER

This music program provides you with a kind of two octave organ. Written by Gwyn Dewey, the program allows you to compose a tune (giving you a visual printout as you do so), then play it back. . . or you can select one of two preprogrammed melodies (the National Anthem or a light classical piece called 'Carnival' which you'll probably recognise when you hear it). Full instructions are included with the program.

```
   1 PAPER 7: INK 1: CLS
   3 GO SUB 2000
   5 DATA 12,20,36,8,11,15,14,12
,12,20,36,8,27,47,14,12,4,4,4,8,
11,15,14,12,4,4,4,8,11,13,10,12,
0,0,0,0,3,13,10,12
   6 FOR i=USR "a" TO USR "a"+39
   7 READ a: POKE i,a
   8 NEXT i
   9 LET b$=""
  10 LET a$="zsxdcvgbhnjm#3e4rt5
y7u8i"
  13 CLS
  15 FOR g=0 TO 31
  20 IF INKEY$="" THEN GO TO 20
  30 FOR i=1 TO LEN a$
  31 IF INKEY$="l" THEN GO TO 30
00
  32 IF INKEY$="1" THEN GO TO 10
0
  33 IF INKEY$="0" THEN LET b$=b
$( TO (LEN b$)-2): BEEP .5,-20:
PRINT AT 22-i,g;"."
  40 IF INKEY$=a$(i) THEN GO TO
53
  50 NEXT i
  52 GO TO 20
  53 INK (RND*6)
  54 IF i=24 OR i=23 THEN GO TO
56
  55 PRINT AT 22-i,g;CHR$ (144+R
ND*4)
  60 BEEP 0.5,i-1
  64 IF i-1<10 THEN LET b$=b$+"0
"
  65 LET b$=b$+STR$ (i-1)
  70 NEXT g
  80 GO TO 13
 100 CLS
 105 INK 0
 106 PRINT "press key for each n
ote playback"
 107 PRINT '"and release finger
at end of    note"
 111 FOR i=1 TO 50
 112 NEXT i
```

```
 120 FOR i=1 TO LEN b$ STEP 2
 125 IF INKEY$="" THEN GO TO 125
 130 BEEP 0.05,VAL b$(i TO i+1)
 135 IF INKEY$<>"" THEN GO TO 13
0
 140 NEXT i
 150 PRINT INK 3;"End of tune"
 160 PRINT AT 21,0; INK 4;"Press
 "; FLASH 1;"ENTER"; FLASH 0;" t
o continue"
 170 IF INKEY$="" THEN GO TO 170
 175 CLS
 180 GO TO 8000
2000 CLS
2005 BORDER 4
2010 PRINT "Music"
2020 REM © 1982 G.Dewey
2030 PRINT '"This music program
has two      octaves.First you p
ut in your   tune using the foll
owing keys"
2040 PRINT INK 2;'"[illegible]"; FLASH 1;
"[illegible]"; FLASH 0;"[illegible]"; FLASH 1;"[illegible]";
 FLASH 0;"[illegible]"'" [illegible]"; FLASH 1;"[illegible]
[illegible]"; FLASH 0;"[illegible]"'" [illegible]"; FLASH
 1;"[illegible]"; FLASH 0;"[illegible]"; FLASH 1;"[illegible]
[illegible]"; FLASH 0;"[illegible]ENTER"'"©"; FLAS
H 1;"[illegible]"; FLASH 0;"©SPACE"
2050 PRINT '"1 Plays notes back.
You press a  key to control temp
o"
2060 PRINT '"0 Deletes last note
 entered and produces warning bu
zz"
2065 PRINT '"l selection of tune
s"
2070 PRINT AT 21,0; INK 6;"Press
 to compose"
2075 COPY
2080 IF INKEY$="" THEN GO TO 208
0
2090 RETURN
3000>CLS
3002 PRINT "1=UK National anthem
"
3004 PRINT "2=Carnival"
3100 INPUT "Which tune?";dd: LET
 dd=2*(dd=1)+4*(dd=2): IF dd<2 O
R dd>4 THEN GO TO 3100
3110 GO SUB 6998+(dd*2)
3120 GO TO 100
7001 RETURN
7002 LET b$="12121411121416161 71
61412141211121214161719191917 1
61717171716141617161412161719211
716141212"
7003 RETURN
7006 LET b$="1211121520192022201
92012151913121315191819201918191
31912121112152019202220192012171
```

```
71710131715131210080705071008"
7009 RETURN
8000 CLS
8010 INPUT "Do you want to compo
se again(y/n)?"; LINE a$
8020 IF a$="y" THEN RUN
```

```
Music

This music program has two
octaves.First you put in your
tune using the following keys

1234567890
 QWERTYUIOP
 ASDFGHJKLENTER
©ZXCVBNM©SPACE

1 Plays notes back.You press a
key to control tempo

0 Deletes last note entered and
produces warning buzz

l selection of tunes


Press to compose
```

KNIGHTSBRIDGE

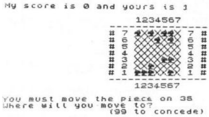
Knightsbridge is a game played on a seven by seven checkers board, with each player having seven chess knights as pieces (which explains the name). The computer throws a die which determines which piece the human player must move, then it throws a die for itself to determine which piece it must move. All pieces move like knights in chess (and you are not allowed to cheat) and a piece is captured by the opponent's piece landing on top of it. The first player to capture five of the opponent's pieces wins.

You move the indicated piece by telling the computer which square you wish to end up in. The board is labelled one to seven along the top and bottom, and from seven to one down the sides. You enter the number along the side (e.g. 6) then the number along the top (e.g. 4) as a single number (i.e. 64) to move your piece onto square number 64. You will be swiftly chastised if you try to cheat. You can concede at any time by entering 99. Knightsbridge was written by Tim Hartnell and modified by Peter Shaw.

```
  10 REM KNIGHTSBRIDGE
  20 REM TIM HARTNELL
         PETER SHAW
  30 GO SUB 5000: GO SUB 9000
  40 GO SUB 8000
  50 GO SUB 7000
  60 GO SUB 6000
  70 GO SUB 6000
  80 GO TO 40
  90 STOP
5000 POKE 23609,50: LET x=0: LET
 y=0: LET hu=0: LET co=0: LET q1
=0: LET q=0
5010 BORDER 6: PAPER 6: BRIGHT 1
: CLS
5020 FOR a=USR "a" TO USR "d"-1
5030 READ b: POKE a,b
5040 NEXT a
5050 LET a$="KNIGHTSBRIDGE KNIGH
TSBRIDGE KNIGHTSBRIDGE "
5060 PRINT AT 20,3; INK 1;"PRESS
 ANY KEY TO CONTINUE"
5070 FOR a=1 TO 7
5080 PRINT AT 5,0; INK a;a$( TO
32)
5090 IF INKEY$<>"" THEN RETURN
5100 LET a$=a$(2 TO )+a$(1)
5110 PAUSE 2
5115 BEEP .01,a
5120 NEXT a
5130 GO TO 5070
5020 LET q=0
```

```
6030 LET m=INT (RND*66)+12
6040 LET q=q+1
6050 IF q=500 THEN GO TO 8000
6060 IF h(m)<>146 THEN GO TO 603
0
6070 PRINT "You must move the pi
ece on ";m
6080 PRINT "Where will you move
to?"
6085 PRINT ,"(99 to concede)"
6090 INPUT n: LET p=0: IF n=99 T
HEN GO TO 6130
6095 FOR w=1 TO 8: IF m+z(w)=n T
HEN LET p=1
6096 NEXT w: IF p=0 THEN PRINT I
NK 2;"## ILLEGAL MOVE ##": GO TO
 6080
6100 IF h(n)=145 THEN LET hu=hu+
1
6110 LET h(m)=144: LET h(n)=146
6120 RETURN
6130 LET q=500
6140 GO TO 8000
7010 LET q1=0
7020 LET q1=q1+1
7030 LET k=INT (RND*66)+12
7040 IF q1=500 THEN GO TO 8000
7050 IF h(k)<>145 THEN GO TO 702
0
7060 PRINT "I must move the piec
e on ";k: PAUSE 50
7070 FOR w=1 TO 8
7075 IF k+z(w)<11 OR k+z(w)>77 T
HEN GO TO 7090
7080 IF h(k+z(w))=146 THEN GO TO
 7400
7090 NEXT w
7100 FOR w=1 TO 8
7105 IF k+z(w)<11 OR k+z(w)>77 T
HEN NEXT w
7110 IF h(k+z(w))<>144 THEN NEXT
 w
7115 IF w=8 AND h(k+z(w))<>144 T
HEN LET q1=50
7116 IF q1=500 THEN GO TO 8000
7120 LET x=k: LET y=k+z(w)
7130 LET h(x)=144: LET h(y)=145
7140 RETURN
7400 REM Capture
7410 LET co=co+1
7420 GO TO 7120
8000 REM Print board
8010 CLS
8018 IF x>0 THEN PRINT "I moved
from ";x;" to ";y
8020 PRINT '''"My score is ";co;
" and yours is ";hu: PRINT
8030 PRINT ,"    1234567"
8040 PRINT ,"-----------------"
8050 FOR j=70 TO 10 STEP -10: BE
EP .1,j/10
```

```
8060 LET a=h(j+1): LET b=h(j+2):
 LET c=h(j+3): LET d=h(j+4): LET
 e=h(j+5): LET f=h(j+6): LET g=h
(j+7)
8070 PRINT ,"# ";j/10;" ";CHR$ a
;CHR$ b;CHR$ c;CHR$ d;CHR$ e;CHR
$ f;CHR$ g;" ";j/10;" #"
8075 NEXT j
8080 PRINT ,"-----------------"
8090 PRINT ,"    1234567"
8095 PRINT
8100 IF hu=5 OR co=5 OR q1=500 O
R q=500 THEN GO TO 8400
8300 RETURN
8400 REM End
8410 IF hu=5 THEN PRINT "HUMAN D
EFEATS MACHINE!!!"
8420 IF co=5 THEN PRINT "MACHINE
 PROVES VICTORIOUS!!!"
8430 IF q=500 THEN PRINT "I ACCE
PT YOUR WISH TO CONCEDE"
8440 IF q1=500 THEN PRINT "I CON
CEDE, HUMAN"
8500 STOP
9000 REM Initialise
9010 CLS : LET hu=0: LET co=0
9020 DIM h(99): DIM z(8)
9030 FOR a=1 TO 99
9040 LET h(a)=0
9050 IF a>77 OR a=70 OR a=60 OR
a=68 OR a=69 OR a=50 OR a=59 OR
a=58 OR a=40 OR a=49 OR a=48 THE
N GO TO 9090
9055 IF a=30 OR a=38 OR a=39 OR
a=20 OR a=29 OR a=28 OR a<11 THE
N GO TO 9090
9060 LET h(a)=144
9070 IF a>70 AND a<78 THEN LET h
(a)=145
9080 IF a>10 AND a<18 THEN LET h
(a)=146
9090 NEXT a
9100 FOR a=1 TO 8
9110 READ b
9120 LET z(a)=b
9130 NEXT a
9140 FOR m=1 TO INT (RND*32)
9150 LET p=RND*m
9160 NEXT m
9500 RETURN
9600 DATA 129,66,36,24,24,36,66,
129
9610 DATA 4,28,118,254,254,28,28
,126
9620 DATA 32,56,110,127,127,56,5
6,126
9990 DATA -8,-21,-12,-19,19,12,2
1,8
```



```
My score is 0 and yours is 1

                 1234567
            ----------------
            # 7         7 #
            # 6         6 #
            # 5         5 #
            # 4         4 #
            # 3         3 #
            # 2         2 #
            # 1         1 #
            ----------------
                 1234567

You must move the piece on 35
Where will you move to?
                     (99 to concede)
```

CATERKILLER

In this program, which will not take you long to enter, you control the 'caterkiller', a squirming string of little copyright signs. You have to stay alive as long as you can. The blue hash symbols are worth ten points each if you run over them, but the white asterisks are fatal. Run into one and the game is over. You control the caterkiller's movement with the 'z' and 'm' keys (and note this program is written to by played with CAPS LOCK *off*). However, an added complication in the game is that you continue moving in one direction until you either hit the key to change the direction, or you come to the edge of the screen.

There is a high score feature (see about half way along the very long line 100; so you can try and better previous scores.

```
   3 REM Caterkiller
   4 REM Graham Charlton and
             Tim Hartnell
   5 LET highscore=0
  10 BORDER 2: PAPER 2: CLS
  15 LET b$="z"
  20 LET a=10: LET b=15: LET c=2
1: LET t=0
  30 POKE 23692,0
  40 LET t=t+1
  60 INK 1: PRINT AT c,RND*31;" #
"
  70 INK 7: PRINT AT c,RND*31;"*
"
  80 PRINT
  85 LET a$=INKEY$
  86 IF a$="" THEN LET a$=b$
  87 IF a$<>"" THEN LET b$=a$
  90 LET b=b+(a$="m" AND b<30)-(
a$="z" AND b>1)
 100 IF SCREEN$ (a,b)="*" THEN B
EEP .002,8: FOR x=1 TO 50: PRINT
 AT 6,11; INK RND*7;"You scored
";t: BEEP .005,x: NEXT x: LET hi
ghscore=(t AND highscore<t)+(hig
hscore AND highscore>t): FOR x=1
 TO 50: PRINT INK RND*6;TAB RND*
16;"Highscore is ";highscore: BE
EP .005,x/2: NEXT x: GO TO 10
 105 IF SCREEN$ (a,b)="#" THEN L
ET t=t+10
 110 PRINT AT a,b; INK 6;"©"
 120 GO TO 30
```

```
You scored 26
```

POLOMONY

POLOMONY, written by Chris Callender and Peter Shaw, is a Spectrum version of the well-known board game which has a name using the same letters, but in a slightly different order. Need we say more? If you have the board game, you may wish to move real pieces around the board, following the computer's instructions.

```
  10 REM POLOMONY
  20 REM CHRIS CALLENDER
         PETER SHAW
  30 GO SUB 8000: REM TITLE PAGE
  40 PRINT : PRINT : PRINT
  50 PRINT "HELLO THERE!!"
  60 DIM b$(40,32)
  70 FOR a=1 TO 40: READ b$(a):
NEXT a
  80 DIM p$(40,32): DIM c$(40,32
): DIM v(40): DIM p(40): DIM c(4
0)
  90 FOR a=1 TO 40: READ v(a): N
EXT a
 100 LET m=9000: LET cm=9000: LE
T p=1: LET cp=1
 130 INPUT "Press enter to roll
dice o.k."; LINE a$
 140 LET p=p+INT (RND*6)+1: IF p
>40 THEN LET p=1: LET m=m+200: G
O TO 500
 150 LET f=0: IF b$(p)( TO 2)="G
O" THEN LET m=m+200: GO TO 500
 160 IF b$(p)="Community Chest
             " THEN GO TO 1000
 170 IF b$(p)="Income tax
             " THEN GO TO 1500
 180 IF b$(p)="Chance
             " THEN GO TO 2000
 190 IF b$(p)( TO 4)="Jail" THEN
 GO TO 2500
 200 IF b$(p)="Parking
             " THEN GO TO 3000
 210 IF b$(p)="GO TO JAIL!
             " THEN GO TO 3500
 220 IF b$(p)="Super tax
             " THEN GO TO 4000
 230 PRINT "You land on ";b$(p)'
"worth $";v(p)
 240 IF c$(p)=b$(p) THEN GO TO 4
00
 250 PRINT "Will you buy ";b$(p)
: INPUT x$
 295 IF x$(1)="n" THEN GO TO 500
 300 INPUT "How many houses ";n:
 INPUT "a hotel? ";a$: IF a$(1)=
"y" THEN LET n=5
 310 LET p(p)=p(p)+(v(p)*n)
 320 LET m=m-(200*n)
 330 PRINT "Good": PAUSE 50
 340 GO TO 500
 400 PRINT "You are trespassing
on my         property- Pay $";c(p
)
 410 LET m=m-c(p): LET cm=cm+c(p
)
 420 PRINT "Ta' very much!"
 500 PRINT "You now have $";m;"
 I have $";cm
 505 PRINT "________________________
____________"
 510 IF m<0 THEN GO TO 6000
 515 IF cm<0 THEN GO TO 7000
 520 PRINT "My go!!"
 530 LET cp=cp+INT (RND*6)+1: IF
 cp>40 THEN LET cp=1: LET cm=cm+
200: GO TO 690
 550 LET f=1
 560 IF b$(cp)="Community Chest
                 " THEN GO TO 100
0
 570 IF b$(cp)="Income tax
                 " THEN GO TO 150
0
 580 IF b$(cp)="Chance
                 " THEN GO TO 200
0
 590 IF b$(cp)="Jail
                 " THEN GO TO 250
0
 600 IF b$(cp)="Parking
                 " THEN GO TO 300
0
 610 IF b$(cp)="GO TO JAIL!
                 " THEN GO TO 350
0
 620 IF b$(cp)="Super tax!
                 " THEN GO TO 400
0
 621 IF p$(cp)=b$(cp) THEN LET c
m=cm-p(cp): LET m=m+p(cp): PRINT
 "I trespass on ";b$(cp): GO TO
590
 625 IF v(cp)=0 THEN GO TO 690
 627 IF b$(cp)=c$(cp) THEN GO TO
 690
 629 PRINT "I landed on ";b$(cp)
 630 IF cm/v(cp)>10 OR cm<1000 T
HEN GO TO 690
 640 LET n=INT (RND*6)+1
 650 IF n=5 THEN PRINT "I buy ";
b$(cp)'"With 4 houses and 1 hote
l": GO TO 680
 670 PRINT "I buy ";b$(cp)'"With
 ";n;" houses"
 680 LET c(cp)=(v(cp)/10): LET c
(cp)=c(cp)+(n*200): LET c$(cp)=b
$(cp)
 690 PAUSE 50
 700 PRINT "________________________
____________"
 710 GO TO 130
1000 IF f=0 THEN PRINT "You land
 on Community chest"
1010 IF f=1 THEN PRINT "I land o
n community chest"
1020 LET c=INT (RND*5)+1
1030 GO SUB (1060 AND c=1)+(1100
 AND c=2)+(1140 AND c=3)+(1180 A
ND c=4)+(1220 AND c=5)
1040 IF f=1 THEN GO TO 690
1050 IF f=0 THEN GO TO 500
1060 IF f=0 THEN PRINT "You inhe
rit $100": LET m=m+100
1070 IF f=0 THEN PRINT "I inheri
t $100": LET cm=cm+100
1080 RETURN
1100 IF f=0 THEN PRINT "Your ann
uity matures-collect      $200": L
ET m=m+200
1110 IF f=1 THEN PRINT "My annui
ty mature-I collect $200": LET c
m=cm+200
1120 RETURN
1140 IF f=0 THEN PRINT "Bank err
or in your favour        collect
$200": LET m=m+200: RETURN
1150 PRINT "Bank error in my fav
our. I get  $200": LET cm=cm+200

1160 RETURN
1180 IF f=0 THEN PRINT "You have
 won $10 in a beauty    contest"
: LET m=m+10: RETURN
1190 PRINT "I have won $10 in a
beauty          contest": LET cm=cm+
10
1200 RETURN
1220 IF f=0 THEN PRINT "HA!, Pay
$50 insurance": LET m=m-50: RETU
RN
1230 PRINT "Ugh I have to pay $5
0 insurance": LET cm=cm-50
1240 RETURN
1500 IF f=0 THEN PRINT "Hee Hee-
Pay $200 income tax ": LET m=m-2
00: GO TO 1040
1510 PRINT "Oh no! I must pay $2
00 in incometax": LET cm=cm-200
1520 GO TO 1040
2000 IF f=0 THEN PRINT "You land
 on Chance"
2010 IF f=1 THEN PRINT "I land o
n chance"
2020 LET c=INT (RND*5)+1: IF c=5
 THEN GO TO 2170
2030 GO SUB (2050 AND c=1)+(2080
 AND c=2)+(2110 AND c=3)+(2140 A
ND c=4)
2040 GO TO 1040
2050 IF f=0 THEN PRINT "You won
```

```
a crossword puzzle & get$100": L
ET m=m+100: RETURN
2060 IF f=1 THEN PRINT "I win a
crossword puzzle & get  $100": L
ET cm=cm+100
2070 RETURN
2080 IF f=0 THEN PRINT "You must
 go back three places": LET p=p-
3
2090 IF f=1 THEN PRINT "I must g
o back three place": LET cp=cp-3
2100 RETURN
2110 IF f=0 THEN PRINT "Advance
to Mayfair!": LET p=40
2120 IF f=1 THEN PRINT "I advanc
e to Mayfair": LET cp=40
2130 RETURN
2140 IF f=0 THEN PRINT "You have
 been speeding again-   that's $
15": LET m=m-15: RETURN
2150 PRINT "I am fined $15 for s
peeding": LET cm=cm-15
2160 RETURN
2170 IF f=0 THEN PRINT "GO TO JA
IL,Do not pass go!": GO TO 3500
2180 PRINT "I must go to jail, O
h dear!": GO TO 3500
2500 IF f=0 THEN PRINT "You are
just visiting jail": GO TO 1040
2510 PRINT "I am in jail, but ju
st visiting": GO TO 1040
3000 IF f=0 THEN INPUT "Do you w
ant to park";d$
3010 IF f=1 THEN PRINT "I will n
ot park here": GO TO 1040
3020 IF d$(1)="n" THEN PRINT "Wi
se choice": GO TO 1040
3030 LET f=0: GO TO 1040
3500 IF f=0 THEN PRINT "You're i
n jail,$50 to get out": LET m=m-
50: GO TO 1040
3510 PRINT "I'm in jail & must p
ay $50": LET cm=cm-50
3520 GO TO 1040
4000 IF f=0 THEN PRINT "You must
 pay $100 super tax": LET m=m-10
0
4010 IF f=1 THEN PRINT "I must p
ay $100 super tax": LET cm=cm-10
0
4020 GO TO 1040
6000 PRINT "You have no money! S
orry you      lose": INPUT "anothe
r game?";a$
6020 IF a$(1)="y" THEN RUN
6030 STOP
7000 PRINT "Boo I lose. Well don
e": GO TO 6010
8000 BORDER 0: PAPER 0: INK 6: C
LS
8005 DRAW 0,175: DRAW 255,0: DRA
```

```
W 0,-175: DRAW -255,0
8010 PLOT 24,151: DRAW 207,0
8020 DRAW 0,-127: DRAW -207,0: D
RAW 0,127
8030 LET a$="POLOMONY POLOMONY
POLOMONY "
8035 FOR a=1 TO 7
8036 PRINT AT 16,10; PAPER 8-a;
INK 0;"PRESS A KEY"
8037 BORDER a
8040 PRINT AT 6,4; INK a;A$( TO
24): PAUSE 2
8050 LET A$=A$(2 TO ) +A$(1)
8060 IF INKEY$<>"" THEN GO TO 81
00
8070 NEXT a
8080 GO TO 8035
8100 BORDER 0: CLS
8110 RETURN
9000 DATA "GO","Old Kent Road","
Community Chest","Whitechapel Ro
ad","Income tax"
9010 DATA "King's Cross Station"
,"The Angel Islington","Chance",
"Euston Road"
9020 DATA "Pentonville Road","Ja
il","Pall Mall","Electric compan
y","Whitehall"
9030 DATA "Northumberland avenue
","Marylebone Station","Bow Stre
et"
9040 DATA "Community Chest","Mar
lbourgh Street","Vine Street","P
arking","Strand"
9050 DATA "Chance","Fleet Street
","Trafalgar Square","Fenchurch
Station"
9060 DATA "Leicester square","Co
ventry Street","Water works","Pi
ccadily"
9070 DATA "GO TO JAIL!","Regents
 Street","Oxford Street","Commun
ity Chest"
9080 DATA "Bond Street","Liverpo
ol Street Station","Chance","Par
k lane"
9090 DATA "Super tax","Mayfair"
9100 DATA 0,60,0,60,0,200,100,0,
100,120,0,140,150,140,160,200,18
0,0,180,200,0
9110 DATA 220,0,220,240,220,260,
260,150,280,0,300,300,0,320,200,
0,350,0,400
```

CONNECTEZ-VOUS QUATRE

If your French is as bad as mine, you'll immediately recognize that this program is a version of 'Four in a Row' or 'Connect Four'. It is very simple to play, and you are given (see line 100) the choice of making the first or second move. The Spectrum does not put up a particularly strong defence, and does not always know when the game is over. You may wish to improve the program to compensate for these two weaknesses. Connectez-vous Quatre was written by Tim Rogers.

```
  5 REM Connectez-vous Quatre
 10 BORDER 0
 20 INK 6
 30 PAPER 0
 40 CLS
 45 REM © TdR 82
 50 INPUT
    "What is your name, kiddo?"
    ,a$
 60 IF a$="" THEN LET
    a$="Human"
 65 DIM r(8)
 70 PRINT "██████████"
 80 FOR a=1 TO 8
 85 PRINT "█++++++++█"
 90 NEXT a
 95 PRINT "█12345678█"
100 PRINT #1;"First or second"
    ,a$;"?"
110 PRINT #1;"Press '1' OR '2'"
120 IF INKEY$<>"1" AND INKEY$<>
    "2" THEN GO TO 120
125 INPUT ""
130 IF INKEY$="1" THEN
    GO TO 300
140 LET i=INT (RND*8)+1
150 GO TO 400
300 IF INKEY$<>"" THEN
    GO TO 300
310 PRINT AT 12,0;"Your go  ";
    a$
320 PRINT "Which slot?"
330 LET i$=INKEY$
335 IF INKEY$=" " THEN
    GO TO 590
340 IF CODE i$<49 OR CODE i$>56
    THEN GO TO 330
350 LET i=VAL i$
360 IF r(i)=8 THEN GO TO 330
370 LET r(i)=r(i)+1
380 PRINT AT 9-r(i),i;
    PAPER 2;" "
```

```
400 PRINT AT 12,0; FLASH 1;
    "THINKING"
405 FOR a=1 TO 5
410 FOR d=1 TO 8
420 LET h=0
430 LET c=0
440 FOR b=a TO a+3
450 LET c=c+(ATTR (d,b)=14)
460 LET h=h+(ATTR (d,b)=22)
470 NEXT b
480 IF c=4 OR h=4 THEN
    GO TO 600
490 IF c=3 AND ATTR (d,b)=6
    THEN PRINT AT d,b; PAPER 1;
    " ": GO TO 600
500 IF c=3 AND ATTR (d,a-(a>1))
    =6 THEN PRINT AT d,a-1;
    PAPER 1;" ": GO TO 600
510 IF b<9 THEN IF b=3 AND ATTR
    (d,b)=6 THEN LET j=b:
    GO TO 560
520 IF h=3 AND ATTR (d,a-(a>1))
    =6 THEN LET j=a-(a>1):
    GO TO 560
525 BEEP .01,a*d
530 NEXT d
540 NEXT a
550 LET j=i+INT (RND*2)*(i<8)-
    INT (RND*2)*(i>1)
555 IF r(j)=8 THEN GO TO 550
560 LET r(j)=r(j)+1
570 PRINT AT 9-r(j),j; PAPER 1;
    " "
580 GO TO 300
590 POKE 23658,8: REM CAPS LOCK
595 INPUT "Who wins? (H/C)",a$:
    IF a$="C" THEN GO TO 610
600 IF h>2 THEN PRINT AT 0,16;
    "You win ";a$: GO TO 620
610 IF c>2 THEN PRINT AT 0,16;
    "I win ";a$
620 IF INKEY$<>"" THEN
    GO TO 620
625 IF INKEY$="" THEN GO TO 620
630 POKE 23658,0: REM CS LK OFF
635 CLS
640 GO TO 65
```

MOGUL

This simulation program, written by Peter Shaw and Tim Hartnell, allows you to run a stall selling a product (which you can determine) which sells better in warm weather than in cold. Many programs of this type sell lemonade, or ice cream. You determine at the beginning how many days you want the simulation to last, and how many different traders you want to take part.

If you want to run it just by yourself, enter 1 when asked how many are to take part. You'll be lead through the program, to enter your decisions, and the computer will decide what the outcome of each day's trading it. At the end of the simulation, the computer will go through the profits (or otherwise) made by each of the traders, and will decide who has made the most, and thus is worth the title 'Mogul'. There is a lot of room within this progam to adapt it to your own wishes and include your own ideas.

```
  10 REM Mogul
  20 REM Based on program
          by Peter Shaw
  30 LET AC=INT (RND*100)+100: L
ET CP=INT (RND*40)+30
  40 PAPER 7: BRIGHT 1: CLS
 210 INPUT INK 2; FLASH 1;"For h
ow many days do you want   the s
ales simulation to         conti
nue? ";D
 220 INPUT INK 1; FLASH 1;"How m
any traders are            compe
ting?";P
 230 DIM I(P)
 240 DIM A(P)
 280 FOR J=1 TO P
 290 LET A(J)=20000: LET I(J)=20
000
 310 NEXT J
 320 FOR X=1 TO D
 350 FOR J=1 TO P
 360 PRINT AT 1,4; INK 1;"This i
s day "; FLASH 1;X; FLASH 0;AT 3
,3;"Stall number "; INVERSE 1; F
LASH 1;J
 380 LET I(J)=A(J)
 390 PRINT '' INK 7; PAPER 1;"AS
SETS $";A(J)/100
 400 PRINT '' INK 6; PAPER 2;"Ad
vertising signs: $";AC/100;" eac
h"
 410 PRINT ' INK 7; PAPER 3;"Raw
 materials cost ";CP;"c each"
 420 GO SUB 1220
 430 INPUT FLASH 1; PAPER 4; BRI
GHT 1;"How many signs do you wan
```

```
t? ";SI
 440 INPUT FLASH 1; PAPER 3; BRI
GHT 1;"And how many units wholes
ale      will you buy? ";QU
 450 IF QU<1 THEN GO TO 440
 510 LET CO=(SI*AC)+(QU*CP)
 520 IF CO<=A(J) THEN GO TO 570
 530 GO TO 430
 570 INPUT INK 2; BRIGHT 1; FLAS
H 1;"What price (in c) will you
     sell them for? ";SP
 580 LET SA=INT (CH*(SI+1)/SP)
 590 IF SA>QU THEN LET SA=QU
 610 LET TA=SA*SP
 620 LET A(J)=(A(J)+TA)-CO
 625 PRINT '' INK 3;"-----------
--------------------"
 630 PRINT ' INK 2;"Sales: ";SA'
"Day's profit: $";(A(J)-I(J))/10
0'"Current assets: $";A(J)/100
 640 INPUT FLASH 1; INK 2; PAPER
 6;"Press ENTER to continue
";q$
 650 CLS
 710 NEXT J
 720 NEXT X
 730 PRINT INK 3; FLASH 1;"Stand
 by for a summary..."
 740 FOR q=1 TO 30: BEEP q/1000,
q: BEEP q/1500,60-2*q: NEXT q
 750 CLS
 755 LET W1=0: LET W=0
 760 FOR J=1 TO P
 770 PRINT TAB 1; INK 2;"Stall:
";J;TAB 12; INK 1;"Assets: $";A(
J)/100
 780 IF A(J)>W1 THEN LET W1=A(J)
: LET W=J
 790 NEXT J
 800 PRINT '' INK 3; FLASH 1; IN
VERSE 1;"The Mogul is the operat
or of     stall number "; INVERSE
 0;W
 810 PRINT ' INK 3; FLASH 1;"who
 ended up with "; INVERSE 1;"$";
W1/100
1210 STOP
1220 PRINT ''"Here is the weathe
r forecast:"
1230 FOR g=1 TO 30: BEEP g/1000,
40-g: BEEP g/1000,g: NEXT g
1240 LET ch=INT (RND*40)
1250 IF ch>25 THEN PRINT ' INK 2
;"Heat wave": RETURN
1260 IF ch>15 THEN PRINT ' INK 1
;"Fine and sunny": RETURN
1270 IF ch>5 THEN PRINT ' INK 1;
"Cloudy, but mild": RETURN
1280 PRINT INK 2;"Cold and wet"
1290 RETURN
```

```
   Stall number 1

ASSETS $200

Advertising signs: $1.13 each

Raw materials cost 34c each

Here is the weather forecast:

Fine and sunny

-------------------------------

Sales: 18
Day's profit: $-29.67
Current assets: $170.33
```

```
 Stall: 1   Assets: $149.4
 Stall: 2   Assets: $123.5
 Stall: 3   Assets: $176.4

The Mogul is the operator of
stall number 3

who ended up with $176.4



 Stall: 1   Assets: $149.4
 Stall: 2   Assets: $123.5
 Stall: 3   Assets: $176.4

The Mogul is the operator of
stall number 3

who ended up with $176.4
```

COURTBALL DODGE

In this fast-moving game by Peter Shaw, you use the "5" and "8" to move your base in the direction indicated by the arrows over those keys. The zero key is used to fire. Every time your missile hits the bouncing ball your score rises, but you are killed (!) if the ball hits you. The ball is graphic A, the base is graphic B, C and D and the missile is graphic E.

```
  10 REM COURTBALL DODGE
  20 REM PETER SHAW
  30 LET hi=0
  40 LET s=0: LET a=13: LET b=IN
T (RND*5): LET c=INT (RND*30): L
ET d=1: LET e=1
  50 BORDER 7: PAPER 7: BRIGHT 1
: INK 1: CLS
  60 GO SUB 9000
  70 IF b<>21 THEN GO TO 90
  80 IF c=a+1 OR c=a+2 OR c=a+3
THEN GO TO 500
  90 LET a=a+(INKEY$="8" AND a<2
7)-(INKEY$="5" AND a>-1)
 100 PRINT AT 21,a;" ▄█▄ "
 110 IF INKEY$="0" THEN GO SUB 1
000
 120 PRINT AT b,c;" "
 130 LET b=b+d: IF b>20 OR b<1 T
HEN LET d=-d: BEEP .008,15
 140 LET c=c+e: IF c>30 OR c<1 T
HEN LET e=-e: BEEP .008,20
 150 PRINT AT b,c; INK 2;"●"
 160 PAUSE 1
 170 GO TO 70
 500 PRINT AT 1,5;"GAME OVER you
 scored ";s
 510 IF s>hi THEN LET hi=s
 520 PRINT ''TAB 5;"high today "
;hi
 530 INPUT "Would you like anoth
er game "; LINE a$: IF a$(1)="y"
 THEN GO TO 40
 540 STOP
1000 LET g=a+2
1001 FOR f=20 TO 0 STEP -1
1002 PRINT AT f,g;"¦"
1005 IF b<>21 THEN GO TO 1020
1010 IF c=a+1 OR c=a+2 OR c=a+3
THEN GO TO 500
1020 LET a=a+(INKEY$="8" AND a<2
7)-(INKEY$="5" AND a>-1)
1030 PRINT AT 21,a;" ▄█▄ "
1040 PRINT AT b,c;" "
1050 LET b=b+d: IF b>20 OR b<1 T
HEN LET d=-d: BEEP .008,15
1060 LET c=c+e: IF c>30 OR c<1 T
HEN LET e=-e: BEEP .008,20
```

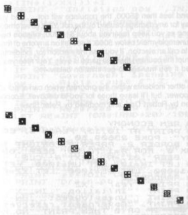
```
1070 PRINT AT b,c; INK 2;"●"
1080 PRINT AT f,g;" "
1085 IF f=b AND g=c THEN GO TO 2
000
1090 NEXT f
1100 RETURN
2000 LET s=s+1
2010 BEEP .5,s*2: BEEP .2,-s*2
2020 CLS : RETURN
9000 RESTORE : FOR z=USR "a" TO
USR "g"-1
9010 READ x: POKE z,x
9020 NEXT z
9030 RETURN
9040 DATA 60,126,255,255,255,255
,126,60
9050 DATA 0,0,0,31,63,127,127,12
7
9060 DATA 0,24,60,255,255,255,25
5,255
9070 DATA 0,0,0,248,252,254,254,
254
9080 DATA 16,8,16,8,16,8,16,8
9090 DATA 145,82,0,195,0,74,137,
0
```

HIGH ROLLER

In this program, which features some nice user-defined dice graphics, you and the Spectrum take it in turns to roll two dice. You add the total of the pips, and add this to your score. You can roll the dice as many times as you like, but if you roll a seven, you automatically lose. The game continues for nine winning rounds (that is, dead heats do not count). Clear player prompts are included within the program, so you should have no problems understanding how to play it when you get the program running. HIGH ROLLER was written by Tim Hartnell.

```
   5 REM HIGH ROLLER
  10 GO SUB 625
  15 PAPER 7: BRIGHT 1: CLS : BO
RDER 2
  17 LET hs=0: LET cs=0
  20 PRINT AT 3,1; INK 9; PAPER
RND*7; FLASH 1;"Press 'R' to rol
l, 'Q' to quit"
  30 LET w$=INKEY$
  40 IF w$<>"q" AND w$<>"Q" AND
w$<>"r" AND w$<>"R" THEN GO TO 3
0
  50 PRINT AT 3,0;"
"
  60 IF w$="q" OR w$="Q" THEN GO
 TO 300
  70 GO SUB 800
  80 IF z=7 THEN GO TO 520
  90 LET hs=hs+z
 100 PRINT AT 7,3; FLASH 1; INK
2;"Your total is "; INVERSE 1;hs
 110 GO TO 20
 400 FOR g=1 TO 40
 405 PRINT AT 5,3; FLASH 1; INK
RND*7;" Stand by "
 410 BEEP g/1000,40/g: NEXT g
 430 GO SUB 800
 440 IF z=7 THEN GO TO 500
 450 LET cs=cs+z
 460 PRINT AT 9,4; INK 2;"My tot
al is "; INVERSE 1; FLASH 1;cs;
INVERSE 0; FLASH 0; INK 1;AT 11,
2;"Your total is "; FLASH 1;hs
 470 IF cs<hs THEN GO TO 400
 480 IF cs=hs THEN PRINT AT 18,1
2; INK 2; PAPER 6; FLASH 1;"Dead
 heat!"
 485 IF cs>hs THEN GO TO 520
 490 GO TO 1000
 500 PRINT AT 12,12; INK 2; PAPE
R 6; FLASH 1;"You win!": LET h=h
+1
 510 GO TO 1000
 520 PRINT AT 12,12; INK 2; PAPE
R 6; FLASH 1;"I win!": LET cz=cz
+1
 530 GO TO 1000
 620 STOP
 625 DIM e(6): LET h=0: LET cz=0
 630 FOR a=1 TO 6
 640 READ d
 650 FOR b=0 TO 7
 660 READ c
 670 POKE USR CHR$ d+b,c
 680 NEXT B
 685 LET E(A)=D
 690 NEXT A
 700 RETURN
 710 DATA 144,255,255,255,231,23
1,255,255,255
 720 DATA 145,255,159,159,255,25
5,249,249,255
 730 DATA 146,255,159,191,231,23
1,253,249,255
 740 DATA 147,255,153,153,255,25
5,153,153,255
 750 DATA 148,255,153,189,231,23
1,189,153,255
 760 DATA 149,255,153,255,219,21
9,255,153,255
 800 REM Roll dem bones
 805 CLS
 810 FOR q=1 TO 20
 820 LET x=INT (RND*6)+1
 830 LET y=INT (RND*6)+1
 840 PRINT AT 6,6; INK 1;CHR$ e(
x)
 850 PRINT AT 8,8; INK 2;CHR$ e(
y)
 855 BORDER RND*7
 860 BEEP .001,50-2*q: BEEP .001
,2*q
 870 NEXT q
 875 BORDER 2
 880 LET z=x+y
 890 PRINT AT 12,3; FLASH 1; INK
 4;"Roll was "; INVERSE 1;x; INV
ERSE 0;" and "; INVERSE 1;y
 895 PAUSE 200
 900 PRINT AT 12,3;"
     ";AT 6,6;" ";AT 8,8;" "
 910 RETURN
1000 PRINT FLASH 1; INK 4;"Score
:"
1010 PRINT INK 2;'"You ";h
1020 PRINT INK 1;'" Me ";cz
1025 IF cz+h=9 THEN GO TO 1060
1030 IF h>cz THEN PRINT '' FLASH
 1; INK 2;"You are leading"
1040 IF cz>h THEN PRINT '' FLASH
 1; INK 2,"I am leading"
1060 FOR g=1 TO 100: BEEP .002,g
-50: NEXT g
```

```
1070 CLS
1075 IF cz+h=9 THEN GO TO 1090
1080 GO TO 17
1090 CLS
1095 PRINT AT 4,4; INK 3; INVERS
E 1;"End of the game..."
1100 IF h<cz THEN PRINT INK 2; F
LASH 1;'''TAB 5;"I win!!"
1110 IF h>cz THEN PRINT INK 2; F
LASH 1;'''TAB 5;"You win!!"
```



ECONOMY

Do you think you could run the economy better than the government? Don't we all. Here's a program which puts you in the hot seat to control government spending. If you spend more than one sixth of your reserves, unemployment will be reduced, but spending less than one eight will increase it, although inflation will fall. Like governments in real life, one of your perpetual battles will be to balance unemployment and inflation in such a way as to keep the economy in the best health possible.

If you spend less then $5000, the populace will riot. . . and you'll have to pay for any damage caused by the riot. Your government is safe so long as you keep reserves about $10,000, inflation below 40% and unemployment below 5000. If you fail on any one of these, you have to call an election. If you lose the election (or, incidentally, you bankrupt the country) then the game is over. Your reserves will be boosted if you manage to get inflation below 4%.

There are other incidents within the program which can affect your period in power, but I'll leave those for you to discover. Economy was written by Robert Day, and modified by Peter Shaw.

```
  10 REM ECONOMY
  20 PRINT AT 10,13; FLASH 1;"EC
ONOMY": POKE 23609,50
  30 BORDER 2: PAPER 2: BRIGHT 1
: INK 0: FOR A=1 TO 50: BEEP .00
8,A: BEEP .01,-A: NEXT A: CLS
  40 LET IN=10: LET UN=1000: LET
 GR=10000: LET GS=1000: LET XZ=0
  50 RANDOMIZE
  60 PRINT "Year ";XZ
  70 PRINT '"Inflation ";IN;"%"
  80 PRINT '"Unemployment ";UN
  90 PRINT '"Government reserves
 £";GR: IF GR<0 THEN PRINT "NO M
ORE RESERVES": STOP
 100 PRINT '"Government spending
 ";GS
 110 INPUT "What will Government
 spending benext year ? £";G: IF
 G<GR THEN LET GS=G
 120 IF G<GR+1 AND G>0 THEN GO T
O 150
 130 IF G>GR+1 THEN PRINT '"INSU
FFICIENT RESERVES": PAUSE 200
 140 CLS : GO TO 70
 150 CLS
 160 LET GR=GR-G
 170 LET ZZ=(RND*5)+2
```

```
 180 PRINT "You spent £";G
 190 PRINT "Reserves now £";GR
 200 IF G>(GR/6) THEN LET UN=INT
 (UN*(1/ZZ))+1: PRINT "High Gove
rnment spending cuts  unemploym
ent to ";UN
 210 IF G<(GR/8) THEN LET UN=INT
 (UN+(UN*(2/ZZ))): PRINT "Low go
vernment spending raises  unempl
oyment to ";UN
 220 LET XX=(RND*3)+1
 230 IF G<(GR/8) THEN LET IN=INT
 (IN*(XX/4))+1
 240 IF G>(GR/8) THEN LET IN=INT
 (IN+(IN*(1/XX)))+1
 250 PRINT '"Inflation now ";IN;
"%"
 260 LET V=INT (RND*5000)+1
 270 IF IN>3 THEN GO TO 310
 280 PRINT "Good inflation rate,
 boost to  reserves of £";V
 290 LET GR=GR+V
 300 PRINT "Reserves now £";GR
 310 LET Z=INT (RND*10000)+1
 320 IF G>4999 THEN GO TO 370
 330 PRINT "Government spending
too low,    Riots cause £";Z;" w
orth of": PRINT "damage"
 340 LET GR=GR-Z
 350 IF GR<1 THEN GO TO 510
 360 PRINT "Reserves now £";GR
 370 LET RE=INT (GR*(RND*20)/100
)+1
 380 PRINT "Investment gain £";R
E
 390 LET GR=GR+RE
 400 PRINT "Reserves now £";GR
 410 LET OIL=INT (RND*4)+1
 420 IF OIL<>4 THEN GO TO 490
 430 LET DR=INT (RND*2500)
 440 PRINT "****OIL CRISIS****"
 450 PRINT "Drain on reserves ";
DR
 460 LET GR=GR-DR
 470 PRINT "Reserves now £";GR
 480 IF GR<1 THEN PRINT "█████BA
NKRUPT█████": STOP
 490 IF GR<10000 THEN PRINT "████
█VERY LOW RESERVES█████": GO TO 5
50
 500 IF IN>40 THEN PRINT "████VER
Y HIGH INFLATION████": GO TO 560
 510 IF UN>5000 THEN PRINT "████V
ERY HIGH UNEMPLOYMENT████": GO TO
 560
 520 PAUSE 700
 530 CLS
 540 LET XZ=XZ+1
 550 GO TO 60
 560 PRINT "YOU MUST CALL FOR AN
 ELECTION!"
 570 PRINT "PRESS ENTER FOR A RE
SULT"
 580 LET W=INT (RND*3)+1
 590 INPUT LINE A$
 600 IF W<>1 THEN GO TO 640
 610 PRINT "▪▪▪▪▪▪▪▪YOU LOST▪▪▪▪▪
▪▪"
 620 PRINT "You lasted ";XZ;" ye
ars"
 630 STOP
 640 PRINT "▬▬!!YOU WON!!▬▬ Pr
ess newline"
 650 INPUT LINE a$
 660 CLS
 670 GO TO 60
```

REACTION TEST

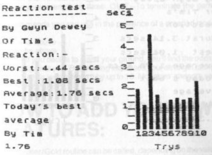
This program pits your reflexes against the relentless Spectrum timer. Written by Gwyn Dewey, the program features a very clever printout at the end of each round, showing how well you did in that round, with your best and worst times.

When you are ready to start, press ENTER, and the first of ten characters–chosen at random from the Spectrum keyboard and including punctuation marks, and upper and lower case letters–will appear. You have to press the relevant key (using SHIFTs as needed) as quickly as you can. The screen will then clear and after a short delay, a new character will appear.

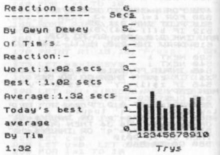
At the end of the round, a bar graph, as shown in the sample printout, will tell you how well you did. If you want to use the program for several people, to get the best result from among them, change the GOTO 90 at the end of line 560 to GOTO 10.

```
   7 LET h=1000000
   8 LET h$=""
   9 LET b$="0  1  2  3  4  5  6
 7"
  10 BORDER 0: PAPER 0: INK 7: C
LS
  20 PRINT AT 0,9; INK 2; PAPER
6;"REACTION TEST"''
  30 PAUSE 50
  40 PRINT INK 4;"When you are r
eady press 'ENTER'& a random cha
racter will appearon the screen(
33-90&94-122).You must then pres
s this character  on the keyboar
d & your time is  recorded.You h
ave ten goes & if your average i
s good you could  beat the "; FL
ASH 1;"HIGH SCORE!"
  50 PRINT ' INK 6;"REMEMBER: tim
e is against you."
  60 PRINT INK 6;"So ACT QUICKLY
!!"
  70 INPUT "Enter your name?"; L
INE a$: IF LEN a$>13 THEN GO TO
70
  80 RANDOMIZE
  90 CLS
 100 PRINT INK 2; PAPER 6;"Keybo
ard test"
 110 PRINT '; PAPER 2;"123456789
0"'" QWERTYUIOP"'" ASDFGHJKLen"'
"©ZXCVBNM©sp"
 115 FOR i=1 TO 2
 120 FOR g=1 TO 30: BEEP g/1000,
g: BEEP g/1500,60-3*g: NEXT g
 125 NEXT i
 130 DIM b(10)
 140 FOR i=1 TO 10
 150 FOR g=1 TO 30+RND*40: BEEP
.001,g: NEXT g
 155 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
 160 LET a=INT (RND*90)+33: IF a
>=91 AND a<=93 THEN GO TO 160
 170 PRINT AT 15,10;"Press: ";CH
R$ a
 175 LET w$=INKEY$
 176 PRINT AT 0,20; FLASH 1;w$
 180 IF w$<>CHR$ a THEN GO TO 17
5
 190 LET b(i)=(65536*PEEK 23674+
256*PEEK 23673+PEEK 23672)
 195 FOR g=1 TO 30: BEEP .009,g:
 NEXT g
 200 PRINT AT 15,10;"          ";A
T 0,20;" "
 210 NEXT i
 220 LET c=0
 225 LET y=0: LET z=100000
 230 FOR i=1 TO 10
 240 LET c=c+b(i)
 241 IF b(i)>y THEN LET y=b(i)
 242 IF b(i)<z THEN LET z=b(i)
 250 NEXT i
 260 LET c=INT (c/10)
 270 LET c=c/50
 275 IF c<h THEN LET h=c: LET h$
=a$
 280 PRINT AT 21.0
 285 POKE 23692,255
 290 PRINT INK 1; PAPER 6;"React
ion test"'"---------------"
 300 PRINT INK 3'"By Gwyn Dewey"
 310 PRINT INK 3'"Of ";a$;"'s"
 320 PRINT INK 3'"Reaction:-"
 330 PRINT ' INK 4;"Worst:";y/50
;" secs"
 340 PRINT ' INK 5;"Best  :";z/50
;" secs"
 350 PRINT ' INK 6;"Average:";c;
" secs"
 360 PRINT INK 2'"Today's best"
 365 PRINT INK 2'"average"
 370 PRINT INK 2'"By ";h$
 380 PRINT INK 2'h
 390 FOR i=0 TO 18
 400 PRINT AT i,19;"_"
 410 NEXT i
 420 FOR i=18 TO 0 STEP -1
 430 PRINT AT i,18;b$(19-i)
 440 NEXT i
 445 FOR i=20 TO 29
 450 PRINT AT 19,i; INK 4+(3*(b(
i-19)<>y AND b(i-19)<>z));i-19
```

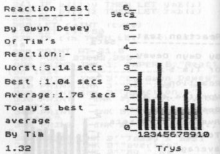
```
 455 NEXT i
 460 PRINT AT 21,23;"Trys"
 465 PRINT AT 1,16; OVER 1;"Secs
"
 470 FOR i=162 TO 238 STEP 8
 500 FOR k=i TO i+3
 510 PLOT INK 2;k,25
 512 IF b((i-154)/8)>=300 THEN D
RAW INK 2;0,150: GO TO 516
 515 DRAW INK 2;0,(b((i-154)/8))
/2
 520 NEXT k
 540 NEXT i
 545 INPUT "Printer Copy?"; LINE
 z$: IF z$="" THEN GO TO 545
 546 IF z$(1)="Y" OR z$(1)="y" T
HEN COPY
 547 CLS
 550 PRINT AT 0,0;"Press'p' to p
lay again or 'q' toquit"
 560 IF INKEY$="p" OR INKEY$="P"
 OR INKEY$=CHR$ 34 THEN GO TO 90
 570 IF INKEY$="q" OR INKEY$="Q"
 THEN GO TO 9010
 580 GO TO 560
```



```
Reaction test
---------------
By Gwyn Dewey
Of Tim's
Reaction:-
Worst:4.44 secs
Best  :1.08 secs
Average:1.76 secs
Today's best
average
By Tim
1.76
```



```
Reaction test
--------------
By Gwyn Dewey
Of Tim's
Reaction:-
Worst:1.82 secs
Best :1.02 secs
Average:1.32 secs
Today's best
average
By Tim
1.32
```



```
Reaction test
--------------
By Gwyn Dewey
Of Tim's
Reaction:-
Worst:3.14 secs
Best :1.04 secs
Average:1.76 secs
Today's best
average
By Tim
1.32
```

DRAGONFIRE

How do you fit 65535 ADVENTURE games, each with over 1 x 10^{38} locations into the memory of a Spectrum? Just key in this amazing program by Adam Waring and you'll see. The program as listed is only the framework of an adventure. It is not complete for the simple reason that it never can be. There's always something more to be added. . . magic swords, robbers, secret massages, torches and so on. The possibilities are endless. In its present form, the aim of the game is to find five or more pieces of gold and get them back to the exit before your strength runs out. There are ten different types of monster out to thwart you in this laudable aim, and all of the monsters can sap your strength. You'll find that making maps as you progress will help you finish the game successfully.

The commands you can use are:

NORTH–Moves you north, if there is no wall in the way

EAST, SOUTH, WEST–As the NORTH command

HELP–Prints the available commands, and your status (gold, strength and postion)

JUMP–Transports you to any cave within the system. The entrance/exit cave has the co-ordinates 0, 0. You first enter the horizontal co-ordinate, then the direction (EAST or WEST). Follow this with the vertical co-ordinate (NORTH or SOUTH). Jumping reduces your strength by three. QUIT–To terminate the game

The commands you can use in the presence of a dreaded monster are:

HELP–As above

JUMP–As above

FIGHT–If you decide to stand your ground, you'll find the odds are stacked against you. But winning increases your strength rating by up to 30, while losing drops it by up to 10. Monsters' strength varies from one to 100.

HOW TO ADD YOUR OWN FEATURES:

The Monster/Gold routine can be called, depending on the value of Y. To add your own features, use the same technique as in lines 1170 and 1180. If you feel like a real challenge, you could even add a third dimension.

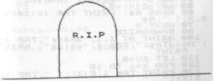
```
  10 REM DRAGON FIRE
  20 REM ADAM WARING
  30 REM PETER SHAW
  40 LET H=100
  50 LET V=100
  60 LET F=0
  70 DIM F(50)
  80 LET S=50
  90 LET G=0
 100 RANDOMIZE INT (RND*65536)
 110 LET X=RND
 120 RANDOMIZE
 130 CLS
 140 POKE 23692,255
 150 BORDER 1: PAPER 1: BRIGHT 1
: INK 9: CLS
 160 FOR m=0 TO 20: PRINT TAB m;
INK (m/4)+2;" DRAGONFIRE": NEXT
m
 170 PRINT AT 2,16; INK 7; FLASH
 1;"PRESS ANY KEY": PAUSE 0
 180 CLS
 190 LET Y=0
1000 IF H=100 AND V=100 AND G>4
THEN GO TO 9000
1010 IF S<1 THEN GO TO 9100
1020 PRINT "Exits are-"
1030 GO SUB 5100
1040 IF NOT Z THEN PRINT "North
";
1050 GO SUB 5300
1060 IF NOT Z THEN PRINT "East "
;
1070 GO SUB 5200
1080 IF NOT Z THEN PRINT "South
";
1090 GO SUB 5400
1100 IF NOT Z THEN PRINT "West"
1110 IF Y<.1 THEN GO TO 4000
1120 IF Y<.15 THEN GO TO 2700
2000 PRINT "What shall I do?"
2010 PAUSE 0
2020 LET A$=INKEY$
2030 IF A$="h" THEN GO TO 4600
2040 LET S=S-1
2050 IF A$="n" THEN GO TO 3000
2060 IF A$="s" THEN GO TO 3100
2070 IF A$="e" THEN GO TO 3200
2080 IF A$="w" THEN GO TO 3300
2090 IF A$="j" THEN GO TO 2500
2100 IF A$="q" THEN STOP
2110 PRINT INVERSE 1;"I CANT DO
THAT"; INVERSE 0: GO TO 1000
2500 PRINT "Jump"
2510 LET S=S-2
2520 INPUT "Horizontal co-ordina
tes";C
2530 IF C<>INT C OR C<0 THEN GO
TO 2520
2540 INPUT "East or west?";A$
2550 IF A$(1)="w" THEN LET C=-C
```

```
2560 LET H=C*2+100
2570 INPUT "Vertical co-ordinate
s ";C
2580 IF C<>INT C OR C<0 THEN GO
TO 2570
2590 INPUT "North or South ";A$
2600 IF A$(1)="n" THEN LET C=-C
2610 LET V=C*2+100
2620 PRINT INVERSE 1;"ZAP"; INVE
RSE 0
2630 GO TO 1000
2700 GO SUB 4500
2710 LET F(N)=Y
2720 LET F=N
2730 LET G=G+1
2740 LET S=S+6
2750 PRINT INVERSE 1;"YAHOO, IVE
 FOUND A GOLDEN NUGGET"
2760 GO TO 2000
3000 PRINT "North"
3010 GO SUB 5100
3020 IF Z THEN GO TO 3500
3030 LET V=V-2
3040 GO TO 1000
3100 PRINT "South"
3110 GO SUB 5200
3120 IF Z THEN GO TO 3500
3130 LET V=V+2
3140 GO TO 1000
3200 PRINT "East"
3210 GO SUB 5300
3220 IF Z THEN GO TO 3500
3230 LET H=H+2
3240 GO TO 1000
3300 PRINT "West"
3310 GO SUB 5400
3320 IF Z THEN GO TO 3500
3330 LET H=H-2
3340 GO TO 1000
3500 PRINT "I can't walk through
 walls twit!"
3510 GO TO 1000
4000 GO SUB 4500
4010 READ A$: IF A$="." THEN RES
TORE : GO TO 4010
4020 PRINT "A";
4030 PRINT A$;
4040 PRINT "blocks your path."
4050 PRINT "What you gonna do no
w?"
4060 PAUSE 0
4070 LET A$=INKEY$
4080 IF A$="h" THEN GO TO 4600
4090 IF A$="f" THEN GO TO 6100
4100 LET S=S-1
4110 IF A$="j" THEN GO TO 2500
4120 GO TO 2110
4500 FOR N=1 TO Z
4510 IF Y=F(N) THEN GO TO 2000
4520 NEXT N
4530 RETURN
```

```
4600 PRINT "Help"
4610 PRINT FLASH 1;"N"; FLASH 0;
"orth", FLASH 1;"S"; FLASH 0;"ou
th"
4620 PRINT FLASH 1;"E"; FLASH 0;
"ast", FLASH 1;"W"; FLASH 0;"est
"
4630 PRINT FLASH 1;"J"; FLASH 0;
"ump", FLASH 1;"F"; FLASH 0;"igh
t"
4640 PRINT "Strength ";S
4650 PRINT "Gold ";G
4660 LET A$=" West"
4670 IF H>100 THEN LET A$=" East
"
4680 PRINT "Horizontal position:
 ";ABS (H-100)/2;A$
4690 LET A$=" North"
4700 IF V>100 THEN LET A$=" Sout
h"
4710 PRINT "Vertical position: "
;ABS (V-100)/2;A$
4720 GO TO 1000
5150 LET V=V-1
5160 GO SUB 6000
5170 LET V=V+1
5190 RETURN
5250 LET V=V+1
5260 GO SUB 6000
5270 LET V=V-1
5290 RETURN
5350 LET H=H+1
5360 GO SUB 6000
5370 LET H=H-1
5390 RETURN
5450 LET H=H-1
5460 GO SUB 6000
5470 LET H=H+1
5490 RETURN
6050 LET Z=SQR ABS (H/X*(V/X))
6060 LET Z=Z-INT Z
6070 LET Y1=Z
6080 IF Z<.5 THEN LET Z=0
6090 RETURN
6130 PRINT "FIGHT!!!"
6150 IF RND*5<20 THEN GO TO 6500
6170 PRINT INVERSE 1;"CONGRATULA
TIONS YOU'VE SLAIN THEGRUESOME C
REATURE"
6200 LET S=S+Y*3
6210 LET F(N)=Y1
6220 LET F=N
6290 GO TO 1000
6500 PRINT INVERSE 1;"YOU'VE BEE
N BEATEN, YOU'RE       NOTHING BU
T A BIG WEED!
                     "
6550 LET S=S-Y
6590 GO TO 2000
6690 RETURN
9000 CLS
9050 PRINT INK 2;"▌▌▌ ▛▀ ▌  ▌
```

```
DONEWELL"
9060 PRINT '"You have escaped -
I didn't       think this was poss
ible"
9090 STOP
9100 BORDER 0: PAPER 0: INK 7
9140 CLS
9150 PRINT INK 6; PAPER 1;"You h
ave joined the many dead."
9160 PLOT 0,40: DRAW 255,0
9170 PLOT 71,40: DRAW 0,56
9180 DRAW 70,0,-PI
9190 DRAW 0,-56
9200 PRINT AT 10,11;"R.I.P"
9900 DATA "n evil spirit "," blo
ated beast "," brainless brute ",
" ghastly ghoul "," deadly devi
l "," mad pirate "," huge horror
 "," terrifying troll "," facele
ss fiend ","n orrible ogre","."
```

```
What you gonna do now?
FIGHT!!!
CONGRATULATIONS YOU'VE SLAIN THE
GRUESOME CREATURE
Exits are-
North East South West
A brainless brute blocks your pa
th.
What you gonna do now?
Jump
CAP
Exits are-
A ghastly ghoul blocks your path
.
What you gonna do now?


You have joined the many dead.
```



STOCK MARKET

Here's your chance to tackle Wall Street, with a portfolio of stocks in ten companies (including IBM, Shell, Toshiba and Xerox).

When you run the game, you'll see three columns. The first is the share value (in dollars), the second the number of shares you hold and the third column contains the names of the companies, along with a number (the number is used to refer to the companies when you trade).

You start the game with $5000, and you can continue as long as your money and stock value combined are above $200. After the share values and holdings are printed out, you'll be asked to nominate a firm in which you wish to trade. The prompt is Firm? and to this you reply by entering the number which refers to the company. Next the prompt Deals? will appear. Enter a positive number if you wish to buy that number of shares, a negative number if you wish to sell. Once you've traded as you wish on the prices you see, enter a zero for the Firms? prompt. You'll be taxed, and the share values will change. Your portfolio total value will also change, of course.

If you wish to modify this program, based on one written by Ian Turtle, to stop when you reach a predetermined value (say $10,000) add: 185 IF M + SV > 10000 THEN STOP
By all means change the names of the firms (in the DATA statement, line 380) into which ever names you choose.

```
  10 REM Stock Market
  15 REM Ian Turtle
  17 PAPER 1: CLS : INK 7: BORDE
R 1
  20 DIM P(20): DIM S(10)
  30 LET M=5000
  35 PRINT AT 5,0;
  40 FOR X=1 TO 10
  50 LET P(X)=21+INT (RND*6)
  60 LET S(X)=0
  65 READ A$: PRINT TAB (X(10)+2
0;X;"  ";A$
  70 NEXT X
  80 RANDOMIZE
 100 PRINT AT 1,1;"Share";TAB 14
;"Holdings";TAB 1;"value": PRINT
 : PRINT
 110 LET SV=0
 120 FOR X=1 TO 10
 130 PRINT TAB 1;P(X);"  ";TAB 14
;S(X);"  "
```

```
 140 LET SV=SV+P(X)*S(X)
 150 NEXT X
 160 PRINT ''"Money: $";M;"      "
 170 PRINT ''"Stock value: $";SV
;"   "
 180 IF M+SV<200 THEN STOP
 190 INPUT "Firm? ";A
 215 IF A>10 THEN GO TO 190
 220 IF A=0 THEN GO TO 300
 230 INPUT "Deals? ";D
 240 IF M-D*P(A)<0 THEN GO TO 23
0
 250 IF S(A)+D<0 THEN GO TO 230
 270 LET M=M-D*P(A)
 280 LET S(A)=S(A)+D
 290 GO TO 80
 300 FOR V=1 TO 3
 310 FOR X=1 TO 10
 320 LET P(X)=P(X)+INT (RND*3)-I
NT (RND*3): BEEP .008,P(X)*(P(X)
<60)
 330 LET M=INT (M-M/400)
 340 IF P(X)<5 THEN LET P(X)=5
 350 NEXT X
 360 NEXT V
 370 GO TO 80
 380 DATA "Toshiba","Shell","IBM
","Exxon","Rolls R.","BP","Gulf"
,"Total","Elf","Xerox"
```

```
 Share          Holdings
 value


 26             0         1 Toshiba
 21             2         2 Shell
 17             47        3 IBM
 26             37        4 Exxon
 21             6         5 Rolls R.
 29             0         6 BP
 24             0         7 Gulf
 22             0         8 Total
 26             20        9 Elf
 17             2        10 Xerox

Money: $2130

Stock value: $2463
```

```
 Share          Holdings
 value

 26             32    1 Toshiba
 21             2     2 Shell
 17             49    3 IBM
 26             37    4 Exxon
 21             50    5 Rolls R.
 29             0     6 BP
 24             6     7 Gulf
 22             0     8 Total
 26             2     9 Elf
 17             2    10 Xerox

Money: $664

Stock value: $3949


 Share          Holdings
 value

 23             32    1 Toshiba
 20             2     2 Shell
 17             49    3 IBM
 22             37    4 Exxon
 18             50    5 Rolls R.
 28             20    6 BP
 24             6     7 Gulf
 24             2     8 Total
 23             2     9 Elf
 18             2    10 Xerox

Money: $0

Stock value: $4157
```

ANAGRAMS

In this anagram program, written by Paul Toland, the computer will choose a word at random from its DATA statements (from line 330) and then jumble up the letters. You enter the letters in another order, and little arrows will appear under the letters in the correct position (the little arrow, of course, comes from the H key, using the red SYMBOL SHIFT key). You have a limited number of guesses (related to the length of the word). Change the contents of the DATA statements when you know all the words. You'll find the program is more effective when the words within the DATA statements bear some similarity to one another (such as FRIEND, and FRIED).

```
   1 DIM D$(25,10)
   2 FOR I=1 TO 25
   3 READ D$(I)
   4 NEXT I
  10 RANDOMIZE
  20 PRINT INK 2;TAB 4;"Anagrams
"
  30 LET R=INT (RND*25)+1
  40 LET W$=""
  50 FOR I=1 TO 10
  60 IF D$(R,I)>" " THEN LET W$=
W$+D$(R,I)
  70 NEXT I
  80 LET L=LEN W$
  90 LET S$="                "( TO L)
 100 FOR I=1 TO L
 110 LET R=INT (RND*L)+1
 120 IF S$(R)>" " THEN GO TO 110
 130 LET S$(R)=W$(I)
 140 NEXT I
 150 PRINT INK 7; PAPER 1;" Your
 anagram is ";S$;" "
 160 FOR J=1 TO 9
 170 PRINT J;" ";
 180 INPUT G$
 190 PRINT G$
 200 IF G$=W$ THEN GO TO 280
 205 PRINT "  ";
 210 FOR I=1 TO LEN G$
 215 IF I>L THEN GO TO 245
 220 IF G$(I)=W$(I) THEN PRINT I
NK 2;"↑";
 230 IF G$(I)<>W$(I) THEN PRINT
" ";
 240 NEXT I: PRINT
 250 NEXT J
 260 PRINT INK 2;"Time is up";TA
B 3;"The word is ";W$
 270 GO TO 290
 280 PRINT INK 1; FLASH 1;"That'
```

```
s it!!"
 285 IF INKEY$<>"" THEN GO TO 28
5
 290 PRINT FLASH 1; INK 2;"Would
 you like to try again?"
 295 LET A$=INKEY$: IF A$="" THE
N GO TO 295
 300 IF A$<>"y" THEN STOP
 310 RESTORE
 320 RUN
 330 DATA "fiend","friend","frie
d","trifle","trifled"
 340 DATA "ordinary","bicycle","
bifocal","between","sequel"
 350 DATA "paradise","parasol","
paragon","antique","article"
 360 DATA "sandwich","savage","s
atisfy","startle","salvation"
 370 DATA "partridge","pardon","
particle","personal","practice"
```

```
    Anagrams
 Your anagram is infde
 1 fiden
   ↑↑
 2 finde
   ↑↑
 3 fied
   ↑↑↑
 4 fined
   ↑↑  ↑
 5 fiend
 That's it!!
 Would you like to try again?
```

NOUGHTS AND CROSSES

This version of the well-known game, written by Ian Turtle and Tim Hartnell, makes good use of the Spectrum's sound and colour. All you need to know to play this is that you enter the number of the position where you wish to move. You are the noughts, the Spectrum is the crosses. The computer decides (line 30) whether it, or you, will have first move.

```
  10 REM Noughts and Crosses
  15 REM Ian Turtle/T Hartnell
  20 DIM A(9): DIM B(9)
  30 IF RND>.5 THEN GO TO 110
  40 GO SUB 700
  50 GO SUB 900
  60 BEEP .03,0: INPUT INK 9; PA
PER RND*7;"Enter your move ";M
  80 IF M<1 OR M>9 OR A(M)<>0 TH
EN GO TO 60
 100 LET A(M)=1
 110 GO SUB 900
 120 GO SUB 700
 130 GO SUB 600
 140 FOR I=1 TO 8
 150 BEEP .008,RND*50: GO SUB 80
0
 160 IF Z=-2 THEN GO TO 340
 170 NEXT I
 180 GO SUB 600
 190 FOR I=1 TO 8
 200 GO SUB 800
 210 IF Z=2 THEN GO TO 340
 220 NEXT I
 230 FOR I=1 TO 10
 240 LET P=2*INT (RND*4)+1
 300 IF A(P)<>0 THEN GO TO 240
 310 LET A(P)=-1
 320 GO SUB 900
 330 GO TO 40
 340 FOR I=1 TO 3
 350 LET V=B(I)
 360 IF A(V)=0 THEN LET A(V)=-1
 370 NEXT I
 380 GO TO 320
 600 LET A$="147258369123456789159357"
 610 RETURN
 700 GO SUB 600
 710 FOR I=1 TO 8
 720 GO SUB 800
 730 IF Z=3 THEN PRINT AT 15,3;
FLASH 1; INK 2;"You win!!!": STO
P
```

```
 740 IF Z=-3 THEN PRINT AT 15,3;
FLASH 1; INK 2;"I win!!!": STOP

 750 NEXT I
 760 FOR Z=1 TO 9
 770 IF A(Z)=0 THEN RETURN
 780 NEXT Z
 790 PRINT AT 15,3; FLASH 1; INK
2;"It's a draw!!!": STOP
 800 LET Z=0
 810 FOR J=1 TO 3
 820 LET B(J)=VAL (A$(1))
 830 LET A$=A$(2 TO )
 840 LET V=B(J)
 850 LET Z=Z+A(V)
 860 NEXT J
 870 RETURN
 900 PRINT AT 4,2; INK 9; PAPER
RND*7;"Spectrum Noughts and Cros
ses";AT 8,6;
 930 FOR Z=1 TO 9
 940 IF A(Z)=1 THEN PRINT INVERS
E 1; INK 2;" O ";: BEEP .02,-10
 950 IF A(Z)=-1 THEN PRINT INVER
SE 1; INK 1;" X ";: BEEP .02,20
 960 IF A(Z)=0 THEN PRINT INVERS
E 1; INK 4;" ";Z;" ";: BEEP .02,
50
 970 IF INT (Z/3)*3=Z THEN PRINT
 : PRINT : PRINT TAB 6;
 980 NEXT Z
 990 RETURN
```

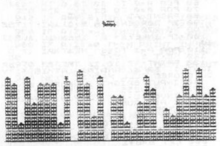
```
  Spectrum Noughts and Crosses


         O   2   X

         4   X   6

         X   O   O


    I win!!!
```

BOMB RUN

In this version of CITY BOMBER, written by Peter Shaw, you pilot a rather elderly biplane over the city, doing your best to demolish the skyscrapers which lie in your path. You drop a bomb by pressing the '0' key. The plane is graphic H and graphic B, the bomb is graphic C, the top of the building is graphic D and the main body of the building is graphic E.

```
  10 REM BOMB RUN
  20 REM PETER SHAW
  30 CLS
  40 GO SUB 9000
  50 GO SUB 8000
  60 FOR A=0 TO 20
  70 FOR B=0 TO 31
  80 PRINT AT A,B;" ✈"
  90 IF SCREEN$ (A,B+3)<>" " THE
N GO TO 270
 100 IF INKEY$="0" THEN GO SUB 1
60
 110 FOR P=1 TO 5: NEXT P
 120 NEXT B: NEXT A
 130 PRINT AT 10,10; FLASH 1; BR
IGHT 1;"SAFE LANDING";AT 15,6;"Y
OU SCORED FULL MARKS"
 140 INPUT BRIGHT 1; FLASH 1; IN
VERSE 1;"PRESS ": INVERSE 0;"ENT
ER"; INVERSE 1;" TO PLAY AGAIN "
; LINE A$: RUN
 150 STOP
 160 LET B1=B+3
 170 FOR X=A+1 TO 20
 180 PRINT AT X-1,B1;" "
 190 PRINT AT X,B1; INK 2;"*"
 200 LET B=B+1
 210 IF B>31 THEN LET A=A+1: LET
 B=0
 220 PRINT AT A,B;" ✈": IF SCRE
EN$ (A,B+3)<>" " THEN GO TO 270
 230 BEEP .008,-(X-20)
 240 NEXT X
 250 PRINT AT X-1,B1;" "
 260 RETURN
 270 FOR C=A TO 19: PRINT AT C,B
+1;"  ";AT C+1,B+1;"✈"
 280 BEEP .1,-(C+2): NEXT C
 290 PRINT AT 10,11; FLASH 1; BR
IGHT 1;"CRASHED";AT 15,9; INVERS
E 1;"YOU SCORED ";(A*10)+B
 300 GO TO 140
8000 FOR K=0 TO 31
8010 LET R=INT (RND*6)+2: LET J=
INT (RND*10)+10
8020 PRINT AT J,K; INK R;"▲"
8030 FOR J=J+1 TO 20
```

```
8040 PRINT AT J,K; INK A;"▦"
8050 NEXT J: NEXT K
8060 PLOT 0,7: DRAW 255,0. RETUR
N
9000 BORDER 0: PAPER 0: INK 7: C
LS
9010 FOR U=USR "a" TO USR "e"+7
9020 READ USER. POKE U,USER
9030 NEXT U: RETURN
9040 DATA 199,225,242,255,127,63
,64,0
9050 DATA 252,32,66,250,255,250,
34,0
9060 DATA 0,36,60,24,60,60,24,0
9070 DATA 0,8,28,63,127,73,73,12
7
9080 DATA 127,73,73,127,127,73,7
3,127
```



```
✈ (PLANE) - GRAPHIC HB
¥ (BOMB) - GRAPHIC C
⌂ (TOP OF BUILDING) - GRAPHIC D
▦ (BUILDING) - GRAPHIC E
```

DRIVER

DRIVER puts you behind the wheel, amid the screech of brakes and the smell of burning rubber. Written by Chris Callender and modified by Peter Shaw, this game is a real challenge to play. We won't make it easy for you, and tell you how to win this one. That's for you to work out. The prompts are clear, and it is up to you to work out the best values to enter.

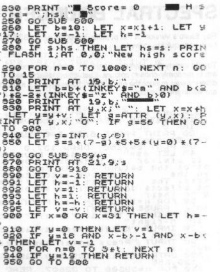
```
  10 REM DRIVER
  20 REM CHRIS CALLENDER
         PETER SHAW
  30 LET p=0: LET r=0: LET s=0:
LET x=0: LET y=0: LET g=0
  40 BORDER 6: PAPER 6: BRIGHT 1
: INK 0: CLS
  50 FOR a=0 TO 21: PRINT TAB a+
2; INK (a/4);"DRIVER": NEXT a
  60 FOR a=1 TO 60: BEEP .008,a:
 BEEP .008,-a: NEXT a: CLS
  70 FOR i=1 TO 30
  80 LET m=INT (20+(RND*120))
  90 PRINT "Part ";i;" Max ";INT
 m
 100 PRINT "Pos ";INT p;" Avge "
;INT (x/i)
 110 PRINT
 120 PRINT "Gear ";
 130 INPUT g1
 140 IF ABS (g1-g)>1 OR g1<1 OR
g1>5 THEN GO TO 120
 150 LET g=g1: PRINT g
 160 PRINT "Accel ";
 170 INPUT a: PRINT a
 180 IF a<1 THEN GO TO 160
 190 PRINT "Brake ";
 200 INPUT b: PRINT b
 210 IF b<0 OR b>2 THEN GO TO 20
0
 220 LET b=b+1
 230 CLS
 240 LET s=s/(b+1)+g*a*2
 250 IF s/g>7 THEN GO TO 280
 260 PRINT "STALLED!!-Select 1st
gear"
 270 LET s=0: LET r=0: LET g=0
 275 GO TO 110
 280 IF s<m+m/3 THEN GO TO 310
 290 BORDER 0: INK 7: PAPER 0: C
LS : PRINT "Crashed": FOR a=1 TO
 20: BEEP .1,a: BEEP .05,-a: NEX
T a
 300 STOP
 310 LET r=r/4+a*2000/9
 320 IF r<8500 THEN GO TO 350
 330 BORDER 0: INK 7: PAPER 0: C
LS : PRINT "Engine blew": FOR a=
10 TO 1 STEP -1: BEEP .1,a: BEEP
 .05,-a: NEXT a
 340 STOP
 350 LET x=x+s
 360 LET y=y+m
 370 LET j=y-x
 380 IF j<0 THEN LET j=0
 390 LET p=(j-14)/15+1
 400 PRINT "Gear   ";INT g
 410 PRINT "Accel  ";INT a
 420 PRINT "Tacho  ";INT r
 430 PRINT "Speed  ";INT s
 440 PRINT "Dist   ";INT (y-x)
 450 PRINT : NEXT i
 460 CLS
 470 PRINT "Race over"
 480 PRINT "Final place ";INT p
 490 PRINT "Average speed was ";
INT (x/30)
 500 STOP
```

SPECTRAL BOUNCER

Roy Easto's version of the popular computer game BREAKOUT runs very quickly indeed.

To add variety from game to game, the bricks are randomly, and are in randomly-chosen colours. Each colour affects the ball in a different way, making the game more challenging. Full instructions are within the program.

```
  10 LET hs=0
  15 RESTORE : PAPER 7: BORDER 5
: INK 0: CLS
  20 PRINT PAPER 6;AT 2,8;"Spect
ral Bouncer."''
  30 PLOT 64,150: DRAW 137,0
  40 LET a$="   The game Bouncer
 is similar  to Breakout, you co
ntrol the batand try to knock do
wn the        various coloured br
icks. The     bricks have differe
nt effects onthe bricks accordin
g to their    colour."
  50 FOR n=1 TO LEN a$: PRINT a$
(n);: PAUSE 1: NEXT n
  60 PRINT ''TAB 10;"Z-left"'TAB
 10;"M-right"
  70 INPUT "Enter speed(1-fast,3
-slow)";t
  90 INPUT "Press enter when rea
dy";a$
 100 DATA 255,129,129,129,129,12
9,129,255
 110 FOR n=32600 TO 32607: READ
a: POKE n,a: NEXT n
 120 CLS
 130 PRINT "▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯"
 140 FOR x=0 TO 5: FOR y=1 TO 6:
 POKE 22527+y+x*6,y*8
 150 IF x*6+y=32 THEN GO TO 170
 160 NEXT y: NEXT x
 170 FOR n=1 TO 50
 180 LET x=INT (RND*28+2): LET y
=INT (RND*15+2)
 190 IF ATTR (y,x)<>56 THEN GO T
O 180
 200 PRINT PAPER RND*5+1;AT y,x;
"□";: NEXT n
 210 LET b=10: LET x=INT (RND*8)
+11: LET x1=x: LET y=17: LET v=-
1: LET h=1: LET s=0
 220 PRINT AT 20,0;"▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
```



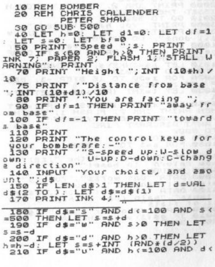
```
 230 PRINT "■ Score= 0      ■■ H s
core= ";hs;" ■"
 250 GO SUB 800
 260 LET b=10: LET x=x1+1: LET y
=17: LET v=-1: LET h=-1
 270 GO SUB 800
 280 IF s>hs THEN LET hs=s: PRIN
T FLASH 1;AT 0,0;"New high score
"
 290 FOR n=0 TO 1000: NEXT n: GO
 TO 15
 800 PRINT AT 19,b;"    "
 810 LET b=b+(INKEY$="m" AND b<2
7)*2-2*(INKEY$="z" AND b>0)
 820 PRINT AT 19,b;"▀▀▀▀"
 830 PRINT AT y,x;" ": LET x=x+h
: LET y=y+v: LET g=ATTR (y,x): P
RINT AT y,x;"O": IF g=56 THEN GO
 TO 900
 840 LET g=INT (g/8)
 850 LET s=s+(7-g)*5+5*(y=0)*(7-
g)
 860 GO SUB 889+g
 870 PRINT AT 21,9;s
 880 GO TO 910
 890 LET v=-1: RETURN
 891 LET h=-1: RETURN
 892 LET v=1: RETURN
 893 LET h=1: RETURN
 894 LET h=-h: RETURN
 895 LET v=-v: RETURN
 900 IF x=0 OR x=31 THEN LET h=-
h
 910 IF y=0 THEN LET v=1
 920 IF y=16 AND x-b>-1 AND x-b<
4 THEN LET v=-1
 930 FOR n=0 TO 3*t: NEXT n
 940 IF y=19 THEN RETURN
 950 GO TO 800
```

BOMBER

Here's your chance to fly a bomber, using the sophisticated computer console of the Spectrum. The keys U, D, S and W are your controls, and they do the following:

U–Moves you up
D–Moves you down
C–Changes directions
S–Speed up
W–Slow down

They have to be following by a number which represents the change you want. For example, U90 will make the bomber rise by 90 feet. Landing for the first time can be an exceedingly tricky business. BOMBER was written by Chris Callender and modified by Peter Shaw.



```
  10 REM BOMBER
  20 REM CHRIS CALLENDER
         PETER SHAW
  30 GO SUB 500
  40 LET h=0: LET d1=0: LET df=1
: LET s=0: LET bf=0
  50 PRINT "Speed ";s: PRINT
  60 IF s<60 AND h>0 THEN PRINT
INK 7; PAPER 2; FLASH 1;"STALL W
ARNING": PRINT
  70 PRINT "Height ";INT (10*h)/
10
  75 PRINT '"Distance from base
";INT (10*d1)/10
  80 PRINT '"You are facing ";
  90 IF df=1 THEN PRINT "away fr
om base"
 100 IF df=-1 THEN PRINT "toward
s base"
 110 PRINT
 120 PRINT "The control keys for
 your bomberare:-"
 130 PRINT '"S-speed up:W-slow d
own:          U-up:D-down:C-chang
e direction"
 140 INPUT "Your choice, and amo
unt ";d$
 150 IF LEN d$>1 THEN LET d=VAL
d$(2 TO ): LET d$=d$(1)
 170 PRINT INK 4;"▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁
▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
 180 IF d$="s" AND d<=100 AND s<
=500 THEN LET s=s+d
 190 IF d$="w" AND s>0 THEN LET
s=s-d
 200 IF d$="d" AND h>0 THEN LET
h=h-d: LET s=s+INT (RND*(d/2))
 210 IF d$="u" AND h<=100 AND d<
```

```
=200 THEN LET h=h+d: LET s=s-INT
 (RND*(d/2))
 220 IF d$="c" THEN LET df=-df
 230 IF h>0 AND s<50 THEN GO TO
1000
 240 IF d1>3 AND h=0 THEN GO TO
2000
 250 IF d1<3 AND h=0 AND s=0 AND
 d1>-3 THEN GO TO 3000
 260 IF d1>30 AND bf=0 THEN GO S
UB 4000
 270 LET d1=d1+(s*1/60*df): GO T
O 50
 500 BORDER 0: PAPER 6: INK 0: C
LS
 510 LET d=0
 515 LET t=INT (RND*10)+1
 520 FOR a=1 TO t: READ z$: NEXT
 a
 990 RETURN
1000 PRINT INK 2; FLASH 1;"YOU S
TALLED"
1010 GO TO 8000
2000 PRINT INK 2; FLASH 1;"YOU M
ISSED THE RUNWAY!"
2010 GO TO 8000
3000 PRINT "You have landed."
3010 PRINT "You hit the ";z$;" w
ell done": GO TO 8000
4000 PRINT "You are over the tar
get now !!!!"'"PRESS 'B' TO DROP
 YOUR BOMB"
4010 LET b$=INKEY$: IF INKEY$=""
 THEN GO TO 4010
4020 FOR z=30 TO 15 STEP -1: BEE
P .05,z: NEXT z
4030 LET t=INT (RND*10): LET bf=
1
4040 LET x=INT (RND*5): IF x=0 T
HEN PRINT INK 2; FLASH 1;"YOU HA
VE JUST BEEN DESTROYED BY
 ANTI AIRCRAFT FIRE          ": GO
TO 8000
4050 RESTORE
4060 FOR a=1 TO t: READ z$: NEXT
 a
4070 PRINT "You hit the ";z$;",
now return  to base!"
4080 RETURN
4090 DATA "Gas works","runway","
aircraft hangers","house"
4100 DATA "power station","water
 tanks","main road","shops","blo
ck of flats"
8000 INPUT "Another game?";g$
8010 IF g$(1)="y" THEN RUN
8020 STOP
```

```
Speed 86

Height 10

Distance from base -1.4

You are facing towards base

The control keys for your bomber
are:-

S-speed up:W-slow down:
U-up:D-down:C-change direction

YOU STALLED
```

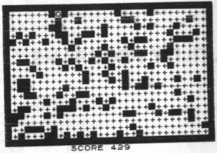
MANTRAP

The fiendish Spectrum is creating blobs to trap you. Your score is related to how long you manage to keep from being caught in a blob trap, with one each side of you so you cannot move. You only score when you are moving, so sitting in the middle will achieve nothing. Full instructions are included within the program. To make it more colourful, change line 240 to:
PRINT INK RND*6 + 1; AT x + c, y + d; "■"

MANTRAP was written by Graham Charlton.

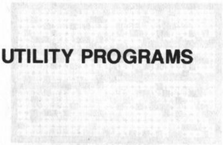
```
  10 BORDER 2: PAPER 2: INK 7: C
LS
  20 GO SUB 260
  30 PRINT AT 0,0;"███████████████
█████████████████"
  40 FOR a=1 TO 19: PRINT "■"; I
NK 0;"+++++++++++++++++++++++++++
++++"; INK 6;"■": NEXT a
  50 PRINT "█████████████████████
███████████"
  60 LET g=0: LET x=10: LET y=16
  70 PRINT FLASH 1;AT x,y;"O"
  80 PRINT AT 21,10;"SCORE 0
 "
  90 INPUT "LEVEL? (1-HIGHEST) "
;l
 100 FOR t=1 TO l
 110 LET a$=INKEY$
 120 IF a$="s" THEN GO TO 30
 130 LET c=(a$="b")-(a$="y")
 140 LET d=(a$="h")-(a$="g")
 150 IF SCREEN$ (x+c,y+d)<>"+" T
HEN GO TO 200
 160 PRINT INK RND*7;AT x,y;"+"
 170 LET x=x+c: LET y=y+d: LET g
=g+ATTR (x,y)
 180 PRINT AT 21,16;g
 190 PRINT FLASH 1;AT x,y;"O"
 200 NEXT t
 210 LET z=INT (RND*4)
 220 LET c=(z=0)-(z=1)
 230 LET d=(z=2)-(z=3)
 240 PRINT AT x+c,y+d;"■"
 250 GO TO 100
 260 PRINT "AVOID THE 'O' BEING
TRAPPED BY"''
 270 PRINT "USING THE FOLLOWING
KEYS:-"''
 280 PRINT "'y' - UP"''
 290 PRINT "'b' - DOWN"''
 300 PRINT "'g' - LEFT"''
 310 PRINT "'h' - RIGHT"''
 320 PRINT "YOU ONLY SCORE WHEN
MOVING TO"''
```

```
 330 PRINT "A POSITION WITH A '+
', YOU SCORE"''
 340 PRINT "POINTS RELATIVE TO T
HE COLOUR"''
 350 PRINT "WHEN YOU BECOME TRAP
PED PRESS"''
 360 PRINT "'s' TO RESTART"''
 370 INPUT "PRESS ENTER WHEN REA
DY "; LINE a$
 380 BORDER 0: PAPER 0: INK 6: C
LS
 390 RETURN
```



SCORE 429

UTILITY PROGRAMS



DECISION-MAKER

If the effort of making decisions is proving too much for you, DECISION-MAKER will prove a real boon. It should act as an genuine aid in making balanced decisions. Here's a sample of how it works. From this sample run, you'll see how to use the program for your own decisions. DECISION-MAKER was written by Chris Callender, and modified by Peter Shaw. The sample run shows the program being used to help decide on a pet.

```
NUMBER OF OPTIONS? 3
NUMBER OF IMPORTANT FACTORS? 3
DECISION MAKER II
INPUT NAME OF OPTION 1
PIG
INPUT NAME OF OPTION 2
SWAN
INPUT NAME OF OPTION 3
HAMSTER

DECISION MAKER II
INPUT NAME OF FACTOR 1
COST
INPUT NAME OF FACTOR 2
AFFECTION
INPUT NAME OF FACTOR 3
COLOUR

DECISION MAKER II
HOW WOULD YOU RATE COST ON THE PIG?
HOW WOULD YOU RATE COST ON THE SWAN?
100
  . . .etc

HOW WOULD YOU RATE AFFECTION ON THE PIG?
99
  .. .etc

THANKS. . .HERE IS WHAT I THINK
PIG  326
SWAN  293
HAMSTER 162
```

```
   1 POKE 23692,255
  10 REM DECISION MAKER II
  20 REM CHRIS CALLENDER
  30 REM PETER SHAW
  40 BORDER 1: PAPER 1: BRIGHT 1
: INK 5: CLS : POKE 23692,255: G
O SUB 8000
  50 PRINT "Number of options ?"
;: GO SUB 9000
  60 LET Z=VAL B$
  70 PRINT "Number of important
factors ?";
  80 GO SUB 9000
  90 LET F=VAL B$
 100 DIM Z$(Z,32)
 110 DIM F$(F,32)
 120 CLS
 130 GO SUB 8000
 140 FOR A=1 TO Z: BEEP .01,A
 150 PRINT "Input name of option
";A
 160 GO SUB 9000
 170 LET Z$(A)=B$
 180 NEXT A
 190 CLS
 200 GO SUB 8000
 210 FOR A=1 TO F: BEEP .01,A
 220 PRINT "Input name of factor
";A
 230 GO SUB 9000
 240 LET F$(A)=B$
 250 NEXT A
 260 DIM M(Z,F): CLS
 270 GO SUB 8000
 280 FOR A=1 TO Z: BEEP .01,A: F
OR B=1 TO F: BEEP .01,B
 290 CLS : GO SUB 8000
 300 PRINT "How would you rate "
;F$(B)
 310 PRINT "on the ";Z$(A)
 320 GO SUB 9000
 330 LET M(A,B)=VAL B$
 340 NEXT B
 350 NEXT A
 360 CLS
 370 GO SUB 8000
 380 PRINT "Thanks...Here's what
I think"
 390 FOR A=1 TO Z: BEEP .01,A
 400 LET C=0
 410 FOR B=1 TO F: BEEP .01,B
 420 LET C=C+M(A,B)
 430 NEXT B
 440 PRINT Z$(A);C
 450 NEXT A: FOR A=1 TO 50: BEEP
.008,A: BEEP .008,-A: NEXT A
 460 PRINT "Another run ";
 470 GO SUB 9000
 480 CLS
 490 IF B$(1)="y" THEN RUN
 500 STOP
```

```
8000 PRINT TAB 5;"DECISION MAKER
 II"
8010 PRINT TAB 5;"▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄"
8020 RETURN
9000 POKE 23609,50
9010 LET B$=""
9020 PAUSE 0
9030 LET A$=INKEY$
9040 IF A$=CHR$ 13 THEN GO TO 90
80
9050 PRINT A$;
9060 LET B$=B$+A$: BEEP .01,20
9070 GO TO 9020
9080 PRINT
9090 RETURN
```

BILLBOARD

Derek Cook's program BILLBOARD allows you to magnify a message from two to seven times. You can get 16 letters into your message if it is enlarged to double its original size, 10 for triple enlargment, eight for four times larger, six for five times, five for six times and just four if you want each letter printed at seven times its original size.

```
   30 REM "© D.L.Cook 1982"
   40 FLASH 0: CLS : BORDER 1: IN
K 0
   50 PRINT "This program will ma
gnify a      message from 2 to 7 
times.The    maximum number of ch
aracters in the message is:
                2x - 16
                3x - 10
                4x - 8
                5x - 6
                6x - 5
                7x - 4"
   80 FLASH 0: PRINT : PRINT "Ent
er number of characters"
   85 INPUT nch
   86 INK 2: PRINT AT 11,27;nch
   87 INK 0: PRINT : PRINT "Enter
 message"
   90 INPUT a$
   91 INK 2: PRINT AT 13,14;a$
   92 INK 0: PRINT : PRINT "Enter
 magnification 2-7"
   95 INPUT n
   96 INK 2: PRINT AT 15,24;n
   97 PAUSE 100: CLS
  100 INK 0: FLASH 1: PRINT AT 0,
0;a$
  110 FOR x=0 TO nch*8-1
  120 FOR y=175 TO 168 STEP -1
  125 LET z=175-y
  130 IF POINT (x,y)=1 THEN GO SU
B 200
  140 NEXT y
  150 NEXT x
  160 INK 1: PRINT AT 20,5;"Anoth
er message? y/n"
  170 INPUT b$
  180 IF b$="y" THEN RUN
  190 STOP
  200 FOR c=0 TO n-1
  205 FOR d=0 TO n-1
  210 INK n-1: PLOT n*x+c,175-(80
+d+z*n)
  220 NEXT d
  230 NEXT c
  240 RETURN
```

SUPER SKETCH

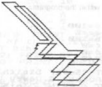
This flexible program from Gwyn Dewey shows you just how much you can do with your Spectrum. The program helps you to draw pictures in colour.

The program is menu-driven, and there is a wide range of options:
Ø – 6 – colour control
Q, W, E, A, D, Z, X, C – control the direction of the line
8 – erases
B – controls brilliance
o – (the small letter o) cancels
9 – sends a copy of the screen to the ZX printer
V – exit from the program
K – saves picture on tape (J loads a picture from tape)

Full instructions are within the program.

```
   10 LET a=100
   15 LET aa=0
   20 LET b=100
   30 LET c=0
   32 LET d=0
   33 LET z=0
   35 CLS
   40 PRINT "Super Sketch"
   50 REM Author G.Dewey
   60 PRINT : PRINT "This program
 helps you to draw  pictures usi
ng the following    keys.NOTE:on
ly one colour is    allowed in a
ny one square"
   70 PRINT '"0-6 colour control"
   80 PRINT '"Q,W,E,A,D,Z,X,C DRA
W"
   90 PRINT '"8 erases o cancels"
  100 PRINT '"B controls brillian
ce"
  110 PRINT '"9 sends a copy of t
he screen to the ZX PRINTER"
  120 PRINT '"V exits"
  130 PRINT '"K saves picture(J l
oads picture)"
  140 IF INKEY$="" THEN GO TO 140
  150 CLS
  151 POKE 23658,0
  154 INVERSE 0: PLOT a,b
  155 INVERSE aa: PLOT a,b
  156 LET b$=INKEY$
  157 IF b$="" THEN GO TO 154
  160 LET a=a-(b$="q" OR b$="a" O
R b$="z")+(b$="e" OR b$="d" OR b
$="c")
```

```
 170 LET b=b-(b$="z" OR b$="x" O
R b$="c")+(b$="q" OR b$="w" OR b
$="e")
 210 IF b$>="0" AND b$<="7" THEN
 INK (VAL b$)
 215 IF b$="8" THEN LET aa=1
 216 IF b$="o" THEN LET aa=0
 220 IF b$="b" AND d=0 THEN LET
d=1
 230 IF b$="b" AND d=1 THEN LET
d=0
 240 BRIGHT d
 250 IF b$="9" THEN COPY
 260 IF b$="v" THEN GO TO 9100
 270 IF b$="k" THEN SAVE "pictur
e"SCREEN$
 280 IF b$="j" THEN LOAD "pictur
e"SCREEN$
 290 GO TO 154
```



CALENDAR

Graham Charlton's CALENDAR program allows you to print out a calendar of the month of your choice – or even a whole year. When you run the program, you'll be asked to enter the year you want, then the number of the month. You enter a zero in response to this prompt if you want the whole year printed out. The program in its present form simply dumps the calendar to the ZX Printer, but can be modified (by changing all the L PRINTS from line 250 to 400 into PRINTs). Note that there are two apostrophes (available with the SYMBOL SHIFT key from the seven key) in line 400.

```
  10 LET a$=" JANUARY FEBRUARY
MARCH    APRIL    MAY      JUNE
   JULY     AUGUST   SEPTEMBER O
CTOBER NOVEMBER DECEMBER "
  20 INPUT "Year?";y
  30 INPUT "Month? (1-12,0 for a
ll) ";m
  40 IF m<0 OR m>12 THEN GO TO 3
0
  50 LPRINT TAB 12;y''
  60 IF m=0 THEN GO TO 90
  70 GO SUB 130
  80 GO TO 20
  90 FOR m=1 TO 12
 100 GO SUB 130
 110 NEXT m
 120 GO TO 20
 130 LET a=y+1+31*(m-1)+INT ((y-
1)/4)-INT (3*INT ((y+99)/100)/4)
 140 IF m>2 THEN LET a=y+1+31*(m
-1)-INT ((4*m+23)/10)+INT (y/4)-
INT (3*INT (y/100+1)/4)
 150 LET a=a-7*INT (a/7)
 160 IF a=0 THEN LET a=7
 170 LET a=a-1
 180 IF m=2 THEN GO TO 220
 190 LET d=31
 200 IF m=4 OR m=6 OR m=9 OR m=1
1 THEN LET d=30
 210 GO TO 240
 220 LET d=28
 230 IF y-4*INT (y/4)=0 AND NOT
y-100*INT (y/100)=0 OR y-400*INT
 (y/400)=0 THEN LET d=29
 240 CLS
 250 LPRINT TAB 10;a$(9*m-8 TO 9
*m)
 260 LPRINT '" SUN MON TUE WED T
HU FRI SAT"''"    ";
 270 IF a=0 THEN GO TO 310
 280 FOR i=1 TO a
 290 LPRINT "    ";
 300 NEXT i
```

```
310 FOR i=1 TO d
320 IF i<10 THEN LPRINT " ";
330 LPRINT i;
340 LET a=a+1
350 IF a-7*INT (a/7)<>0 THEN GO
TO 380
360 LET a=0
370 LPRINT
380 LPRINT "  ";
390 NEXT i
400 LPRINT ""
410 RETURN
```

USER-DEFINED CHARACTER GENERATOR AID

This brief program will save you time if your are producing a number of user-defined characters. Draw out the characters you want, on eight by eight grids, then enter the strings of ones and zeroes relating to a single line of the grid. The program will turn this string of ones and zeroes into a number, which is the number you use to get the character you want. Run it once, and you'll see how easy it is to use, and how useful. Enter 's' to stop it. It is short enough to be worth typing in each time you have several characters to define. Although it is possible to achieve the same thing in shorter ways, you lose the printout of the cumulative total.This growing total can help you learn to 'visual' binary values.

```
  5 REM Character numbers
 10 INPUT a$
 15 IF a$="s" THEN STOP
 17 IF LEN a$<>8 THEN GO TO 10
 20 LET b=0
 25 LET x=1
 30 FOR a=8 TO 1 STEP -1
 40 LET b=b+VAL (a$(a))*x
 45 PRINT b
 50 LET x=x+x
 60 NEXT a
 70 PRINT a$;" ";b
 80 GO TO 10
```

```
1
3
3
3
19
19
83
211
11010011 211
```

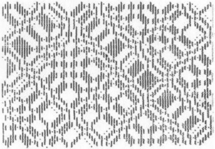
```
0
2
2
10
10
42
42
170
10101010 170
```

GRAPHICS DEMONSTRATIONS

DIAMOND GLORY

Graham Charlton's program treats you to the delights of a moving cursor creating constaly evolving patterns, many of them based on diamond shapes.

```
   5 BORDER RND: PAPER RND+2: IN
K RND*3+4: OVER 1: CLS
  10 LET x=0: LET y=INT (RND*175
)
  20 LET a=0: LET b=INT (RND*175
)
  30 LET g=2+INT (RND*2): LET c=
g: LET h=2+INT (RND*2): LET d=2+
INT (RND*2)
  40 IF a+g>255 OR a+g<0 THEN LE
T g=-g
  50 IF x+c>255 OR x+c<0 THEN LE
T c=-c
  60 IF b+h>175 OR b+h<0 THEN LE
T h=-h
  70 IF y+d>175 OR y+d<0 THEN LE
T d=-d
  80 LET a=a+g
  90 LET b=b+h
 100 LET x=x+c
 110 LET y=y+d
 120 PLOT a,b: DRAW x-a,y-b
 130 GO TO 40
```



SPARTAN SPANGLES

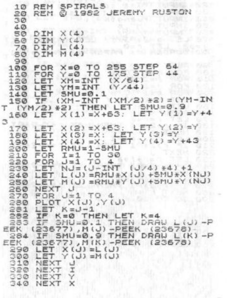
Another great graphics demonstration program from Graham Charlton, which allows you to specify such things as the colours used. Once you've seen it running in its listed form, add the lines shown at bottom for greater variations on the basic process.

```
  10 INPUT "Paper colour",p
  20 INPUT "Ink colour",g
  30 INPUT "Symmetry",s
  40 INPUT "Length",l
  50 INPUT "M",m
  60 INPUT "F",f
  70 LET a=2*PI/s: LET x=128: LE
T y=88: LET r=0
  80 BORDER p: PAPER p: INK g: C
LS
  90 PLOT x,y
 100 FOR h=1 TO s
 110 FOR i=1 TO s
 120 FOR j=1 TO 2
 130 FOR k=1 TO m
 140 LET x=l*SIN r
 150 LET y=l*COS r
 160 DRAW x,y
 170 LET r=r+a
 180 IF r>2*PI THEN LET r=r-2*PI
 190 NEXT k
 200 LET r=r-a+PI
 210 IF r>2*PI THEN LET r=r-2*PI
 220 NEXT j
 230 LET r=r+f*a
 240 IF r>2*PI THEN LET r=r-2*PI
 250 NEXT i
 260 LET r=r+f*a
 270 IF r>2*PI THEN LET r=r-2*PI
 280 NEXT h
 290 GO TO 10
```

```
  65 INPUT "c",c
  68 LET z=2*PI/c
 100 FOR h=1 TO c
 260 LET r=h*z
```

SPIRALS

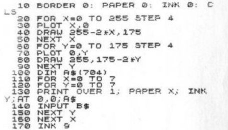
Fill your TV screen and your life with an elaborate pattern in this program written by Jeremy Ruston. Note that the blank lines (such as 30 and 40) are achieved by entering a space after the line number, before using ENTER to move it into the body of the program.



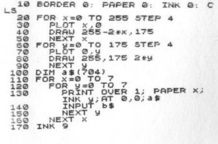
```
  10 REM SPIRALS
  20 REM © 1982 JEREMY RUSTON
  30
  40
  50 DIM X(4)
  60 DIM Y(4)
  70 DIM L(4)
  80 DIM M(4)
  90
 100 FOR X=0 TO 255 STEP 64
 110 FOR Y=0 TO 175 STEP 44
 120 LET XM=INT (X/64)
 130 LET YM=INT (Y/44)
 140 LET SMU=0.1
 150 IF (XM-INT (XM/2)*2)=(YM-IN
T (YM/2)*2) THEN LET SMU=0.9
 160 LET X(1)=X+63: LET Y(1)=Y+4
3
 170 LET X(2)=X+63: LET Y(2)=Y
 180 LET X(3)=X: LET Y(3)=Y
 190 LET X(4)=X: LET Y(4)=Y+43
 200 LET RMU=1-SMU
 210 FOR I=1 TO 30
 220 FOR J=1 TO 4
 230 LET NJ=(J-INT (J/4)*4)+1
 240 LET L(J)=RMU*X(J)+SMU*X(NJ)
 250 LET M(J)=RMU*Y(J)+SMU*Y(NJ)
 260 NEXT J
 270 FOR J=1 TO 4
 280 PLOT X(J),Y(J)
 281 LET K=J-1
 282 IF K=0 THEN LET K=4
 283 IF SMU=0.1 THEN DRAW L(J)-P
EEK (23677),M(J)-PEEK (23678)
 284 IF SMU=0.9 THEN DRAW L(K)-P
EEK (23677),M(K)-PEEK (23678)
 290 LET X(J)=L(J)
 300 LET Y(J)=M(J)
 310 NEXT J
 320 NEXT I
 330 NEXT Y
 340 NEXT X
```

INSTANT BLOPPO

Graham Charlton's program INSTANT BLOPPO draws a design on the screen which you cannot see. When the prompt appears, you press ENTER and the pattern appears almost instantaneously. Every press of ENTER moves into you into a different version of the pattern. The program is listed twice. In the second listing, as you can see, the loops are indented which makes it much easier to work out what is going on in the program. This indenting process is useful to remember for short programs if you want to make it perfectly clear what is occurring in the program.



```
  10 BORDER 0: PAPER 0: INK 0: C
LS
  20 FOR X=0 TO 255 STEP 4
  30 PLOT X,0
  40 DRAW 255-2*X,175
  50 NEXT X
  60 FOR Y=0 TO 175 STEP 4
  70 PLOT 0,Y
  80 DRAW 255,175-2*Y
  90 NEXT Y
 100 DIM A$(704)
 110 FOR X=0 TO 7
 120 FOR Y=0 TO 7
 130 PRINT OVER 1; PAPER X; INK
Y;AT 0,0;A$
 140 INPUT B$
 150 NEXT Y
 160 NEXT X
 170 INK 9
```



```
  10 BORDER 0: PAPER 0: INK 0: C
LS
  20 FOR x=0 TO 255 STEP 4
  30    PLOT x,0
  40    DRAW 255-2*x,175
  50    NEXT x
  60 FOR y=0 TO 175 STEP 4
  70    PLOT 0,y
  80    DRAW 255,175 2*y
  90    NEXT y
 100 DIM a$(704)
 110 FOR x=0 TO 7
 120    FOR y=0 TO 7
 130       PRINT OVER 1; PAPER x;
           INK y;AT 0,0;a$
 140       INPUT b$
 150       NEXT y
 160    NEXT x
 170 INK 9
```