DRAWING PICTURES

In the last instalment of our adventure game programming project, we looked at the design of two graphic screens for the BBC Micro. These screens depicted two locations of importance in our Digitaya game — the ALU and the joystick port. We now look at the design of the same screens for the Sinclair Spectrum.

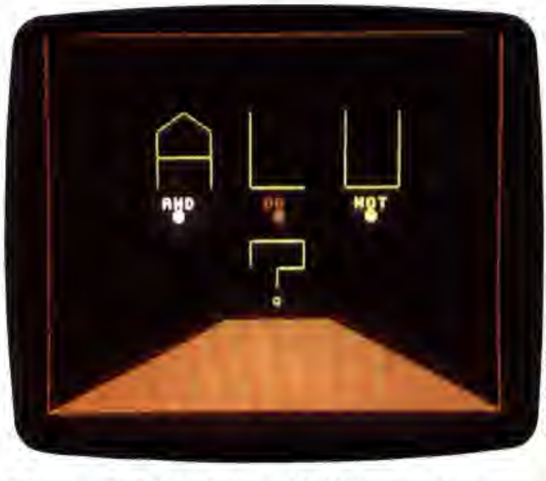
The design of the ALU screen involves the scrolling of the letters A, L and U down to the centre of the screen using high resolution graphics. On the BBC Micro, this scrolling was performed by drawing the letter from a specified start point using relative drawing commands, then rubbing it out, moving the start point and repeating the whole procedure (see page 904). The same idea can be used in the Spectrum version.

The Spectrum's DRAW command allows relative drawing only — that is, starting from the last point specified — but this is ideally suited to this particular scrolling application. By PLOTting an initial start point and then carrying out a series of DRAW commands to create the shape of the letter, we can easily move the entire letter design around the screen by simply changing the co-ordinates of the point initially PLOTted. Rubbing out can be accomplished by drawing the same shape in the same position, but with all the colours inverted. This effect is turned on by using INVERSE 1, and turned off again with INVERSE 0. Thus, for each position that the letter takes up, we shall draw it twice: once with INVERSE 0, to make the shape appear, and again with INVERSE 1 to rub it out.

If we take the example of the letter A, which scrolls on from the left, we can place all these instructions within a FOR . . . NEXT loop. This loop increases the value of the x co-ordinate of the initially-plotted point for the shape. Nested inside this loop is a second FOR . . . NEXT structure that simply carries out the drawing commands twice. The last value of x is 55, which denotes the final resting position of the letter on the screen. Obviously, we do not want to rub out the final version of the letter, so a test is inserted to ensure that the letter will be erased (by switching to INVERSE 1) only if the x co-ordinate is less than 55. The principles discussed here are also applied to the other two letters to make L scroll up the screen and U scroll on from the right.

ALU ROUGH DESIGN

When designing a graphic screen it is important to rough out a design on paper and make an initial estimate of the co-ordinate values that each shape on the screen will have. In addition, any letters that have to be PRINTed to the screen should also be positioned, in terms of rows and columns. The screen shot shows such a design, with the screen dimensions in graphics and character units.



IAN McKINNELL

The words AND, OR and NOT are positioned on the screen using the PRINT AT r,c command: r being the number of rows from the top of the screen, and c indicating the number of columns from the left hand margin. The buttons are drawn using CIRCLE x,y,r where the co-ordinates of the centre and the length of the radius are specified.

On completion of the drawing routines, the program waits for a keypress before resetting the INK and PAPER to the original colours and clearing the screen. It then RETURNs to the main ALU routine. The keypress is tested by INKEY$; if no key is pressed then the test is simply repeated.

To call this subroutine, the following line should be inserted into the Digitaya program:

```
4565 GOSUB 7000:REM ALU PICTURE S/R
```

THE JOYSTICK PORT

The joystick port screen is designed to shoot laser beams from the centre of a joystick connector socket. The pins for the socket are full stop characters PRINTed to the screen and the D-type surround is drawn using high resolution graphics. To give the picture a sense of depth, a series of tapering lines is drawn in the foreground. The start point for each line is on the horizon line, and is selected by a PLOT command. The end of each line is at the bottom of the screen. The lines are spaced at one-unit intervals on the horizon, and widen to seven units at the bottom of the screen.

The fact that Spectrum BASIC's DRAW command is relative makes the routine slightly more complex than if we were able to specify an absolute end

ERRATA

1) In the program at the top of the right-hand column on page 636, the last condition in lines 260 and 370 should read:

```
OR fire=1
```

2) In the program on page 700, the last statement in line 30210 should read:

```
:GOTO 30050
```

3) In Basic Flavours on page 848, replace the first Flavour by:

Replace SN$ by S$, IV$(,) by V$(,), IC$() by I$() and NN$ by R$