be inserted in the main program:

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

JOYSTICK PORT SCREEN

In Digitaya, if a player strays into the joystick port location, then he or she is in danger of being hit by a laser beam. The design of our screen display, therefore, involves drawing a joystick port with laser beams emanating from its centre. The joystick port is drawn using several full stop characters PRINTed to the top left corner of the screen, and a typical D-type surround is then drawn using high resolution graphics and PLOT statements. Notice that after MOVEing to the start position, all of the succeeding PLOT statements that create the port surround are PLOT 1 commands — which means they draw relative to the last point plotted. This is extremely convenient, because if shapes are drawn using a series of relative commands then, if it is decided to move the position of the whole shape, only the first MOVE statement has to be altered.

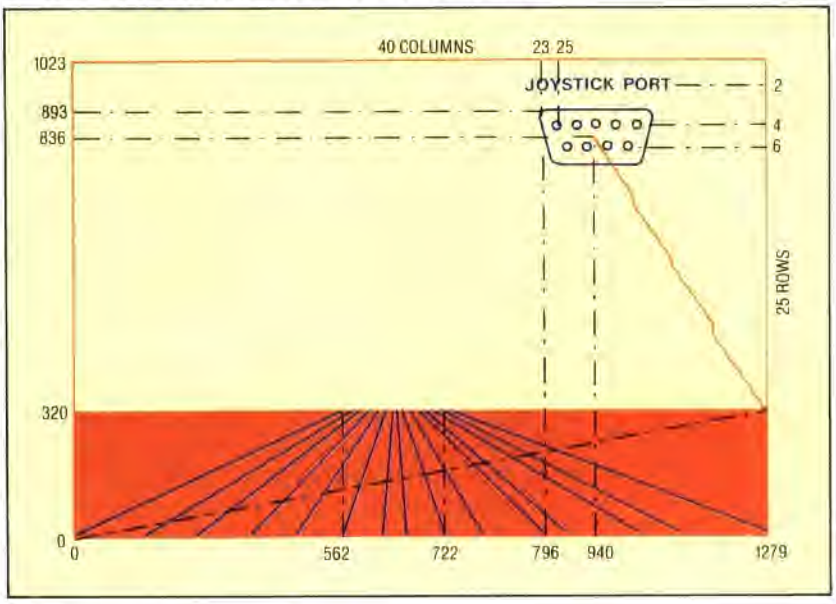
The foreground consists of a rectangular block of colour, once again drawn using two triangular fill primitives. To give an impression of depth, a series of converging lines is drawn over this, using a FOR . . . NEXT loop (lines 8170-8200). The loop sets up values of X from 0 to 1280 — the width of the screen in graphics units. A series of lines is drawn to the bottom of the screen, the start point on the horizon for each point increasing as X increases. However, the step of 32, used between consecutive lines at the bottom of the screen, is reduced to a step of 4 at the horizon (by dividing each X value by 8 in the MOVE command that defines the start point of each line).

The laser beam effect is produced by drawing a line from the centre of the joystick port to a randomly-chosen point on the horizon, in a random colour. The line is subsequently rubbed out — without disturbing the background — by plotting the same line in the Exclusive OR plotting mode, set by GCOL 3. The drawing and rubbing out of lines is placed within a REPEAT. . . UNTIL loop, together with a test to see if a key is pressed on the keyboard. Use of INKEY$, instead of GET$, allows program execution to continue while it is testing for a keypress within the loop. This loop is terminated when a key is pressed, the screen is then cleared, the original text colour restored and program control handed back to the main joystick port routine. To call this graphics subroutine the following line should be inserted:

```
3845 GOSUB 8000:REM JOYSTICK PORT PICTURE
```

Taking Some Stick

In both the joystick port and the ALU screens extensive use is made of the relative plotting facility since it permits easy erasure and movement of whole graphics shapes. Another plotting option is used to DRAW and FILL solid blocks of hi-res colour



KEVIN JONES

ALU Screen

```
7000 REM **** ALU SCREEN S/R ****
7010 CLS :
7015 REM ** CURSOR OFF **
7017 VDU23,1,0;0;0;0;
7020 REM ** BORDER **
7030 GCOL 0,1
7040 MOVE 0,0
7050 DRAW 0,1023
7060 DRAW 1279,1023
7070 DRAW 1279,0
7080 DRAW 0,0
7090 :
7100 REM ** PATH **
7110 MOVE 1279,0
7120 PLOT 85,500,250
7130 PLOT 85,800,250
7140 REM ** LETTER A **
7150 GCOL 3,2
7160 FORX=0 TO 300 STEP 10
7170 FOR I=1 TO 2
7180 PROCletter_a
7190 NEXT I,X
7200 PROCletter_a
7210 :
7220 REM ** LETTER L **
7230 FOR X=300 TO 590 STEP 10
7240  FOR I=1 TO 2
7250 PROCletter_l
7260 NEXT I,X
7270 PROCletter_l
7280 :
7290 REM ** LETTER U **
7300 FOR X=1260 TO 850 STEP -10
7310 FOR I=1 TO 2
7320 PROCletter_u
7330 NEXT I,X
7340 PROCletter_u
7350 :
7360 REM ** BUTTONS **
7370  PROCbutton
7380 COLOUR3
7390  PRINT TAB(11,15)button$
7400 COLOUR1
7410 PRINT TAB(20,15)button$
7420 COLOUR2
7430 PRINT TAB(28,15)button$
7440 REM ** COMMANDS **
7450 COLOUR3
7460 PRINT TAB(10,14)"AND"
7470 COLOUR1
7480 PRINT TAB(19,14)"OR"
7490 COLOUR2
7500 PRINT TAB(27,14)"NOT"
7510 :
7520  MOVE 575,400
7530  PROCg_mark
7540 :
7550 REM ** WAIT FOR KEY **
7560 A$=GET$
7562 REM ** CURSOR ON **
7564 VDU23,1,1;0;0;0;
7566 COLOUR3:CLS
7570 RETURN
7580 :
7590 DEF PROCletter_a
7600 MOVE X,600
7610 PLOT 1,0,150
7620 PLOT 1,75,50
7630 PLOT 1,75,-50
7640 PLOT 1,0,-150
7650 PLOT 0,0,80
7660 PLOT 1,-150,0
7670 ENDPROC
7680 :
7690 DEF PROCletter_l
7700 MOVE 725,Y
7710 PLOT 1,-150,0
7720 PLOT 1,0,200
7730 ENDPROC
7740 :
7750  DEF PROCletter_u
7760 MOVE X,800
7770 PLOT 1,0,-200
7780 PLOT 1,150,0
7790 PLOT 1,0,200
7800 ENDPROC
7810 :
7820 DEF PROCbutton
7830 VDU 23,240,60,126,255,255,255,255,126,60
7840 button$=CHR$(240)
7850 ENDPROC
7860 :
7870 DEF PROCg_mark
7880 PLOT 1,0,60
7890 PLOT 1,150,0
7900 PLOT 1,0,-70
7910 PLOT 1,-75,0
7920 PLOT 1,0,-50
7930 PLOT 0,-8,-30
7940 PLOT 1,16,0
7950 PLOT 1,0,-16
7960 PLOT 1,-16,0
7970 PLOT 1,0,16
7980 ENDPROC
8000 REM **** JOYSTICK PORT PICTURE ****
8010 :
8020 REM ** CURSOR OFF **
8030 VDU23,1,0;0;0;0;
8040 CLS
8050 REM ** BORDER **
8060 GCOL 0,1
8070 MOVE 0,0
8080 DRAW 0,1023
8090 DRAW 1279,1023
8100 DRAW 1279,0
8110 DRAW 0,0
8120 :
8130 REM ** HORIZON **
8140 PLOT 85,1279,320
8150 PLOT 85,0,320
8160 GCOL0,2
8170 FOR X= 0 TO 1280 STEP 32
8180 MOVE 562+X/8,320
8190 DRAW X,0
8200 NEXT X
8210 :
8220 REM ** JOYSTICK **
8230 COLOUR 2
8240 PRINTTAB(23,2)"JOYSTICK PORT"
8250 PRINTTAB(25,4)"
8260 PRINTTAB(25,6)"
8270 GCOL 0,2
8280 MOVE 796,893
8290 PLOT 1,288,0
8300 PLOT 1,4,-4
8310 PLOT 1,4,-4
8320 PLOT 1,0,-4
8330 PLOT 1,-4,-4
8340 PLOT 1,-30,-80
8350 PLOT 1, 4,-4
8360 PLOT 1,-4,-4
8370 PLOT 1,-216,0
8380 PLOT 1,-4,4
8390 PLOT 1,-4,4
8400 PLOT 1,-30,80
8410 PLOT 1,-4,4
8420 PLOT 1,0,4
8430 PLOT 1,4,4
8440 :
8450 REM ** SHOOT **
8460 REPEAT
8470 A$=INKEY$(10)
8480 X=RND(1279):Y=320
8490 GCOL 3,RND(3)
8500 FOR I=1 TO 2
8510 MOVE 940,856
8520 DRAW X,Y
8530 NEXT I
8540 UNTIL A$<>"":REM wait for keypress
8550 :
8560 REM ** CURSOR BACK ON **
8570 VDU23,1,1;0;0;0;
8575 COLOUR3:CLS
```