POWERS OF RESOLUTION

Although high resolution graphics do not form part of the Subhunter game being designed during this project, they are an interesting feature of any home computer. There are no easy-to-use high resolution commands in standard CBM BASIC and hence programs written in BASIC that use high resolution routines tend to be very slow.

For low resolution purposes the Commodore 64's screen is divided into 25 rows of 40 character cells, making 1,000 cells in all. As we know, each character is built up from a series of smaller dots, known as pixels, arranged in eight rows; each character cell therefore consists of 64 pixels. For high resolution purposes we need to be able to switch each pixel on or off individually using a single bit in the computer's memory to control each pixel. This idea is known as *bit mapping*. As each memory location contains eight bits, and the screen is made up of 64,000 pixels, 8,000 memory locations are needed to store the high resolution screen information.

The Commodore 64 is switched from standard low resolution mode to high resolution mode by setting bit 5 of location 53265 to one. To set this bit without disturbing any others, the following command should be used:

```
POKE53265,PEEK(53265)OR32
```

Once high resolution mode has been set, then the screen receives its information from an 8,000-byte block of memory. The start of this block of memory is pointed to by location 53272. This is the same location that was used in the construction of user-defined characters in the last article in this series (see page 232).

The area of memory normally assigned to screen memory is used to hold colour information for each eight-by-eight cell of the screen. The 16 colours available on the Commodore 64 can be represented by only four bits; so the upper four bits of any location in the screen memory are used to hold the colour of the pixels that are 'on' within a particular character cell and the lower four bits hold the colour of any pixels turned 'off'. It is therefore possible to have differing pairs of colours, one for foreground and the other for background, in every cell on the screen. If we want a purple background for the whole screen, with high resolution graphics drawn in black, we need the following codes:

Colour code for black is 0 = 0000 in binary
Colour code for purple is 4 = 0100 in binary

Putting the two parts together will give us 00000100, or 4 in decimal. POKEing 4 into every screen memory location (1024 to 2023) will produce the required black graphics on a purple background.

Before we can start to draw on the high resolution screen, the 8,000-byte area that controls what will be seen must be cleared by POKEing a zero to each location. This will take several seconds in BASIC. If this is not done, the screen display will be a mess of dots. This is because that particular memory area takes random values when the machine is switched on.

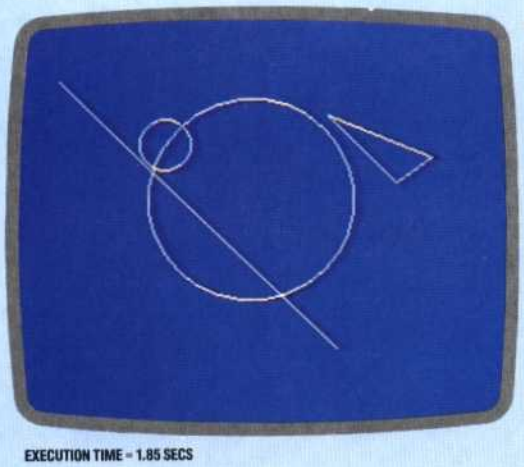
PLOTTING POINTS

A high resolution graphics program needs to be able to switch on or off individual pixels on the screen. If each point is given an X and a Y co-ordinate (X and Y are in the ranges 0 to 319 and 0 to 199 respectively) then the program can identify

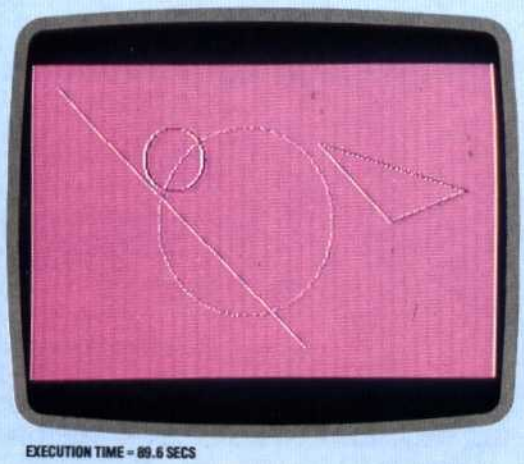
Relative Speeds

Producing this display took the Commodore 64 nearly 90 seconds and about 50 lines of program. Producing the same display on the Spectrum, however, took about two seconds and the following simple program:

```
4000 REM********Res Demo********
4050 LET HW=10: DIM UN(1): INK 6: PAPER 1:
BORDER 1: RESTORE 4100
4100 DATA 20,160,160,-160
4120 DATA 80,120,15
4140 DATA 130,90,80
4160 DATA 175,140
4180 DATA 40,-40
4200 DATA 20,15
4220 DATA -60,25
4250 FOR N=1 TO N:READ UN(N):NEXT N
4300 PLOT UN(1),UN(2):DRAW UN(3),UN(4)
4350 CIRCLE UN(5),UN(6),UN(7)
4400 CIRCLE UN(8),UN(9),UN(10)
4450 PLOT UN(11),UN(12):DRAW UN(13),UN(14)
4500 DRAW UN(15),UN(16):DRAW UN(17),UN(18)
4600 PAUSE 0
```



EXECUTION TIME = 1.85 SECS



EXECUTION TIME = 89.6 SECS

LIZ HEANEY