RANGE OF VIEW

This is the second instalment of a project to build up a graphics game for the BBC Models A and B, and the Electron. Here, we look at high resolution line drawing, the BBC's internal timer, and the remaining procedures to set up the game's scenario.

In the first instalment, we defined an area of the BBC mode 5 screen as the 'minefield' on which our game will be played. We defined the shapes of the mines, the detector and the assistant, developed procedures to lay a number of the mines at random in the minefield, and printed the detector and the assistant in their starting positions. To make the display more attractive we can draw a border around the minefield area. The simplest way to do this is to use the high resolution commands MOVE and DRAW.

There is a problem associated with mixing high resolution graphics with low resolution characters on the BBC/Electron: the different displays use different co-ordinate systems. We have already looked at the low resolution display in detail, when we used it to position the mines and characters with the PRINT TAB(X,Y) command. In this system, the origin (the point where both X and Y are zero) is the top left-hand corner. The X values in this system increase from 0 to 19 from left to right, and the Y values increase from 0 to 31 from top to bottom. Mode 2 also uses a 20 by 32 character display, but all other modes show a different number of characters and so use different co-ordinates for each character.

The eight display modes of the BBC/Electron offer three different resolutions (640 by 256, 320 by 256 and 160 by 256) and yet they all use the same co-ordinate system. Although this system is rather confusing at first, it is a real help in programming because graphics designed for one mode will fit on the screen in the other modes.

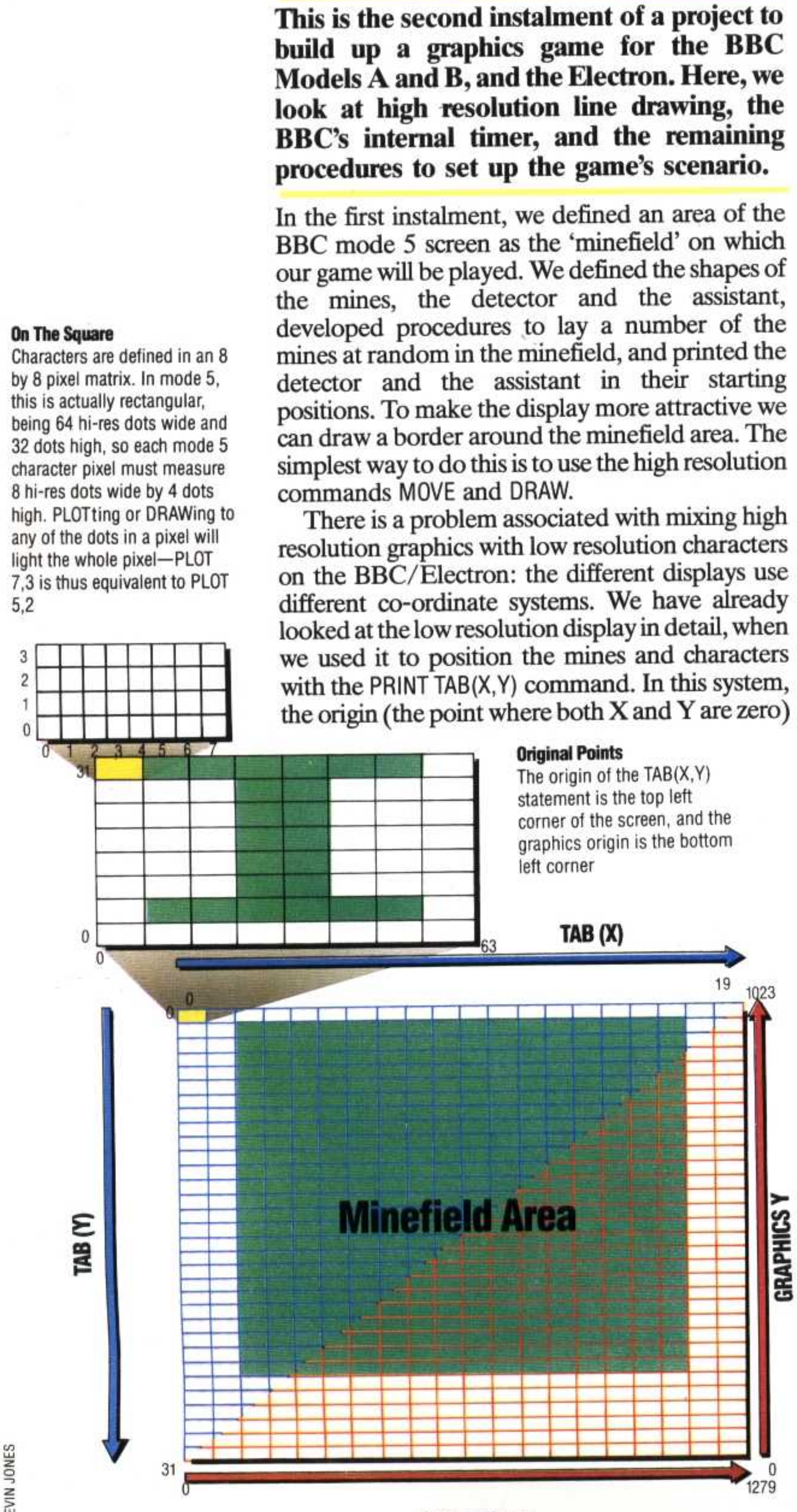
The BBC/Electron co-ordinate system treats the screen as if it has a resolution of 1280 by 1024. This is, of course, a higher resolution than the two machines can produce, apparently to allow for future developments of the BBC! All co-ordinates are given as numbers in the range 0 to 1279 across the screen and 0 to 1023 down the screen. BBC BASIC 'automatically converts these into the appropriate values for whichever of the three screen resolutions is being used.

Mixing graphics with characters is made a little harder because the graphics origin is at the bottom left of the screen rather than the top left used in the character co-ordinate system.

The 160 by 256 pixel resolution in mode 5 is directly linked to the number of characters that can be shown. Each character is formed from an eight by eight pixel grid. As there is enough room for 20 characters across the screen, there must be 8 × 20 = 160 pixels across the screen. Similarly, there must be 32 × 8 = 256 pixels down the screen. To relate this to the high resolution co-ordinate system, we must remember that a pixel is the smallest area of light on the screen that can be individually turned on or off. In the high resolution system, there are 1280 different co-ordinates in the X direction. If we divide this by the number of pixels in the X direction we get 1280/160 = 8. Similarly, in the Y direction, dividing the high resolution co-ordinates by the number of pixels gives 1024/256 = 4. This means that each pixel can be turned on by plotting to any of several positions in the 1280 by 1024 co-ordinate system. The illustration shows how a range of co-ordinates can be used to turn on (or off) one pixel. Plotting (7,3) would light the same pixel as plotting (0,0) or (5,2) and so on.

We can use this fact to relate character positions to high resolution co-ordinates. On the horizontal axis, if one pixel is equivalent to eight units, then one character width will be equivalent to 64 units. Four units are equivalent to one pixel in the

On The Square
Characters are defined in an 8 by 8 pixel matrix. In mode 5, this is actually rectangular, being 64 hi-res dots wide and 32 dots high, so each mode 5 character pixel must measure 8 hi-res dots wide by 4 dots high. PLOTting or DRAWing to any of the dots in a pixel will light the whole pixel—PLOT 7,3 is thus equivalent to PLOT 5,2



Original Points
The origin of the TAB(X,Y) statement is the top left corner of the screen, and the graphics origin is the bottom left corner

KEVIN JONES