ADDING DAZZLE

At this stage in our programming project for the BBC Micro and the Electron, we have developed all the routines that form the basic skeleton of our Mines game. We can now concentrate on adding refinements to the program that will make the game visually appealing and exciting to play.

The first addition we shall make is a 'sniping' routine. This simulates a sniper firing across the minefield trying to hit either the mine detector or the assistant. The firing will be shown as a high resolution line crossing the screen from the left margin of the minefield to the right. To introduce a random element into the sniping, we shall select the co-ordinates of the starting and finishing points using the RND function. The values of xstart and xfinish are set in the initialise variables procedure. The difference between these two values is 1,024 graphics units. If the sniping line is to detect a hit on either the detector or the assistant it must draw a short segment of the line, then test the area ahead for the presence of logical colour 1 (using the POINT command) before drawing the next short segment. This sequence must be repeated until the other side of the screen is reached or a hit is made.

We must now decide on the step length we wish to use. If we choose a very short step length, then the time taken to draw the line will increase. If, however, we have too long a step length we may miss detecting the targets altogether. As each character cell is the equivalent of 64 graphics units across, a step length of half a character cell (i.e. 32 graphics units) would seem reasonable. Therefore, if we choose our step length in the x direction (dx) to be 32 units, we can draw the line in a total of 1024/32=32 steps. If we calculate the y co-ordinates of the start and finish points randomly, then the appropriate step length in the y direction (dy) can be calculated by dividing the difference between the two values by 32.

Our final problem is to find some way of erasing the line after it has been drawn. The solution lies in BBC BASIC's concept of logical colours and its ability to perform logical operations between them. In mode 5 there are four logical colours. Unless we modify them they are:

Logical Colour	0	1	2	3
Binary Equivalent	00	01	10	11
Normal Actual Colour	black	red	yellow	white

Using GCOL we can perform various logical operations between the colour we are plotting and the colour that is already there. The command has two parameters, the second of which indicates the logical colour to be plotted. The first number sets the method of plotting:

GCOL0	Plots the colour specified
GCOL1	Performs OR operation
GCOL2	Performs AND operation
GCOL3	Performs Exclusive OR operation
GCOL4	Performs NOT on the colour already there

This may sound complex, but a few examples should make the operation of the command clear. If white (logical colour 3) is present at the position that we wish to plot to and we want to plot red (logical colour 1) the various modes of operation of GCOL will produce the following results:

GCOL0,1	Will erase white and plot red	
GCOL1,1	ORs red and white to produce white	red 01 white OR 11 white 11
GCOL2,1	ANDs red and white to produce red	red 01 white AND 11 red 01
GCOL3,1	Exclusive ORs red and white to produce yellow	red 01 white EOR 11 yellow 10
GCOL4,1	NOTs white to produce black	white 11 black 00

So how does this help us with our erasing problem? We could plot the line in white and then replot in black to erase it. But if there were already something under the line, such as a mine, then this would cause a 'hole' to be left in it. However, we can Exclusive OR the red with the colour already present at each point the line crosses. When it crosses a white area, we shall get a yellow line segment. If we plot over the same area in Exclusive OR red again, the final result would be:

red	01
yellow	10
EOR	—
white	11

Thus, the original colour is returned. You may wish to verify that performing two Exclusive ORs always leaves you with the original colour. We can use this fact to erase our line. If we plot the original line using an EOR operation and then replot exactly the same line, again using Exclusive OR, we will erase the line and restore any background colours to their original condition before the first