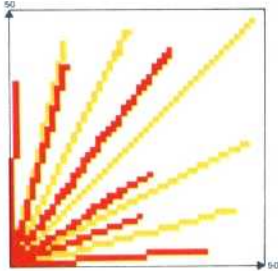
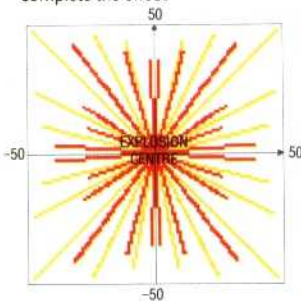


# BLOW UP



## Explosive Play

The explosion graphics are created by generating lines of random length within a range of -50 to 50 units from the centre. The explosion is built up in quadrants and mirrored to complete the effect



In the last section of our project to build up our Minefield game for the BBC Micros and the Electron, we looked at how the program detected collisions with the mines. We now take the program a step further towards completion with a discussion of the procedures we will use to create the visual and sound effects of explosions.

The BBC Micro and the Electron support 16 possible colour variations that we can use in creating an explosion effect. There are actually only eight different colours, the other eight variations being flashing effects between one colour and another. Normally, the flashing combinations of colours alternate every half a second, but the flash rate can be changed by two FX commands. \*FX9 sets the amount of time that the first colour of the flashing pair is displayed. The time is measured in units of fiftieths of a second. Thus, \*FX9,20 alters the display time of the first colour to 2/5ths of a second. The other FX command, \*FX10, can be used to alter the display time of the second colour in the same way. By altering one or both display times, interesting flickering effects can be produced.

The visual explosion effect can be produced by drawing a series of short lines in various colours out from a central point. To enhance the effect, the colour of the mines and score lettering can also be rapidly changed. As these were all originally PRINTed in logical colour 2 (which we set to green) we can use the following command to reassign logical colour 2 to a colour randomly selected from the 16 possible combinations:

```
VDU19,2,RND(15),0,0,0
```

RND(15) selects a whole number from 1 to 15. This means that colour 0 is never selected, but as colour 0 is black, the background colour, this does not matter.

The colour used to plot the explosion lines can also be randomly selected using GCOL 0,RND(3). We then use a loop to draw a series of lines from the centre of the explosion in a series of randomly selected colours. Up to now we have looked at the high resolution commands MOVE and DRAW. DRAW(X,Y) always draws a line to the point with 'absolute' co-ordinates (X,Y). However, there is another family of high resolution drawing commands that allow us to, among other things, specify points 'relative to' one another. The PLOT K,X,Y command can be used to draw lines between relative or absolute points depending on the value of K. The following table shows some of the

variations possible using PLOT:

K=0	Move relative to last point
K=1	Draw line to relative position in foreground colour
K=2	Draw line to relative position in logical inverse colour
K=3	Draw line to relative position in background colour
K=4	Move to absolute position
K=5	Draw line to absolute position in foreground colour
K=6	Draw line to absolute position in logical inverse colour
K=7	Draw line to absolute position in background colour

PLOT4 and PLOT5 are entirely equivalent to the MOVE and DRAW commands. For our explosion effect we shall use PLOT1 to draw lines relative to the centre of the explosion. Before we can draw any lines, however, we need to know where the centre of the explosion is.

In the last section of the course, when we called a dummy version of PROCexplode, we passed the values of xgraph and ygraph to x-explode and y-explode. These played no part in the dummy procedure but will now be used to specify the centre of the explosion. If this procedure is called from line 3390 of PROCmove (see page 435), then xgraph and ygraph will be the graphics co-ordinates of the centre of the character cell containing the mine. Thus, MOVEx-explode,y-explode will move the graphics cursor to the centre of the place we wish the explosion to be. The reason why we pass the co-ordinates to another pair of variables for use within the explosion procedure, instead of simply using xgraph and ygraph, is that this procedure will also be called from another part of the program where the co-ordinates of the explosion centre will be specified by a different pair of variables. By passing parameters into the explosion procedure, we have made it more flexible.

Once we have MOVED to the centre of the explosion, we shall draw a line in a random direction to, say, a maximum of 50 units:

```
PLOT 1,RND(50),RND(50)
```

will not quite do the job, as it will not specify negative co-ordinates. (A negative x co-ordinate produces a line drawn to the left, while a negative y co-ordinate gives a line drawn down from the explosion centre). Repeated use of this command will fill only one quarter of the space around the centre (where both co-ordinates are positive).

So that we can introduce negative values (and hence draw lines in all directions around the explosion centre), we must replace RND(50) with RND(100)-50. The maximum value of RND(100) is, of course, 100, so the most that RND(100)-50 can be is  $100-50 = 50$ . The minimum value of RND(100), on the other hand, is one. Hence,