



THE PLOT THICKENS

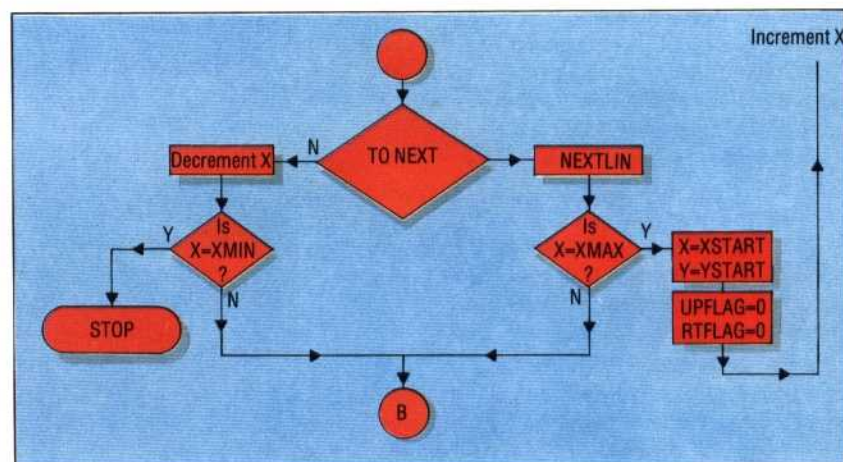
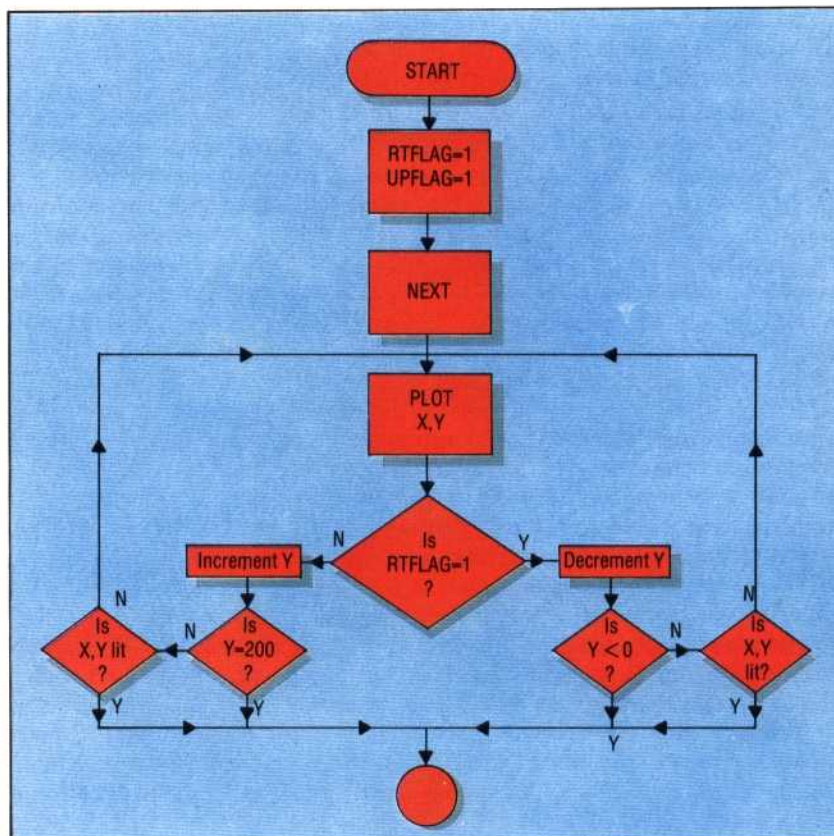
In the course so far, we have developed a series of machine code routines that utilise the high resolution capabilities of the Commodore 64. Plotsub (see page 337), Linesub (see page 416) and Circsub (see page 457) allow us to plot points and draw lines and circles, respectively. Here, we conclude the series with a routine that enables us to fill in the shapes drawn on the screen with the other programs.

There are many ways of designing algorithms to fill shapes on a screen, but what, at first sight, seems to be a fairly simple task is in fact more complex than one might imagine. Shapes that contain 'reflex' angles (interior angles greater than 180 degrees) or shapes that contract and then expand all present their own particular difficulties. It is possible to design a single routine to cope with some of these difficulties but not all of them. Giving a program the 'intelligence' to assess what constitutes a closed shape is not an easy matter.

The method we shall use for our routine begins filling in the shape from a starting point, designated by the user, anywhere within the shape's outline. The routine then progresses up the screen, plotting points until it reaches a boundary, whereupon it moves right one pixel and progresses down the screen until it meets another boundary. Again, a one pixel move to the right is made and the routine starts moving up. This procedure continues until the right-hand side of the shape is filled. The whole process is then repeated, beginning again at the starting point and moving left, until the entire shape is filled.

The first part of the routine is straightforward. Two flags, UPFLAG and RTFLAG, are used to indicate the direction in which the filling routine should move at any stage of the program. The first flowchart segment shows the plot-increment-test part of the routine. The main loop of this section increments or decrements the Y co-ordinate value depending on the state of UPFLAG. After testing for the edge of the screen and assessing whether the next pixel is already lit, the routine loops back to plot the next point. If a lit pixel or a screen edge is encountered, then the routine moves on to the next stage.

The routine will then move either right or left depending on the state of RTFLAG. Detecting the left or right boundary of the shape is difficult. Rather than doing this, the routine allows the user to set maximum and minimum values for the X co-ordinate. This facility also gives the user the option of filling in strips within a shape.



The second flowchart shows that if X is incrementing (i.e. moving right) and reaches its maximum value, then the values of X and Y are reset to the start co-ordinate values and the direction flags are both set to zero in preparation for filling the left part of the shape. If X is decrementing and reaches XMIN, then this indicates the completion of the routine. If the maximum or minimum values of X have not been reached, then the routine must fill in the next line.