Chapter 3 Solutions

3.1 The general problem is how to describe a set of characters that might have thickness, curvatures and holes (such as in the letters a and q). Suppose we consider a simple example where each character can be approximated by a sequence of line segments. One possibility is to use a move/line system where 0 is a move and 1 a line. Then a character can be described by a sequence of the form $(x_0, y_0, b_0), (x_1, y_1, b_1), (x_2, y_2, b_2), ...$ where b_i is a 0 or 1. Thism approach is used in the example in the OpenGL Programming Guide. A more elaborate font can be developed by using polygons instead of line segments.

3.11 There are a couple of potential problems. One is that the application program can map different points in user coordinates to the same point in screen coordinates. Second, a given position on the screen when transformed back into user coordinates may lie outside the users window.

3.15 The problem is with the elements that are entered interactively. When we use double buffering there elements will be rendered into whichever buffer is the back buffer at the time but because they are not part of the display function they will not be the same from frame to frame. What we would like to have but don't is a way to render these elements into both buffers. One possibility is to save all the elements as they are being drawn in some data structure or a display list so they can be rerendered as desired.

3.19 Each scan is allocated 1/60 second. For a given scan we have to take 10scan line n at .9n/(60*1024) seconds from the beginning of the refresh. But allocating 10m on this line at time .81nm/(60*1024).

3.25 When the display is changing, primitives that move or are removed from the display will leave a trace or motion blur on the display as the phosphors persist. Long persistence phosphors have been used in text only displays where motion blur is less of a problem because the long persistence gives a very stable flicker-free image.