

3220 Assignment 5 Report  
William Bolton  
David Benas

To draw a triangle in the FPGA board we implemented edge functions to calculate if a specific pixel is within a given triangle or not. To start we declared the appropriate regs – A0... B2, x, y, x\_min, y\_min, x\_max, y\_max, e0, e1, e2. In addition we provide two regs – screenclear\_x, screenclear\_y, for the purpose of clearing the screen before drawing a triangle.

Most of the computations take place in the initial phase. We check for a special case where  $p1x == p2x$  and  $p2y > p1y$ . We make a few substitutions for the variables in this case. The reason for this is that we found for test cases 2 & 3 there would be an extra line drawn to the screen that was not part of the triangle, so we make substitutions to avoid this. Next, we calculate all the min and max values of x/y, then set  $x = x\_min$ ,  $y = y\_min$ . We then calculate A0... B2 with their respective formulas. For the purpose of making sure that the first pixel of the triangle to be drawn has the correct positive sign for its e0, e1, e2 edge functions, we calculate e0, e1, e2 – if any of these functions is less than zero, we flip the signs for A & B. In addition to the hack mentioned earlier to avoid extra lines being drawn, we check if any of A0... B2 is less than or equal to zero. If it is we add one to it. Our guess as to why this works is because it provides a correct boundary for the edge functions, not allowing them all to return positive when they shouldn't.

Outside of the initial phase the algorithm is straightforward. We first draw to the screen twice with black to clear the screen completely. After this, we cycle through  $x = x\_min$ ,  $x < x\_max$ ,  $y = y\_min$ ,  $y < y\_max$ , calculating e0, e1, e2 at each point and checking if all are greater than or equal to 0, then drawing to the screen at the current pixel if they are.

For fun, when drawing a pixel to the screen instead of using a specific color (i.e. 0xffff), we put the count variable which is automatically incremented each clock cycle into the O\_GPU\_DATA output. The effects of this can be seen when drawing the triangle – it will gradually change color over the course of about 10 – 20 seconds. Pretty neat!