**Drawing Clock Face**

To draw the clock face, I used GL\_TRIANGLE\_FAN with 362 vertices. The initial vertex is located at (256,256) and the rest of the vertices are declared inside of a for loop. I used the knowledge that a point along the circumference of a circle’s position is

**(center vertex x position + radius \* cos (angle of point),**

**center vertex y position + radius \* sin (angle of point))**

When I originally tried to create the clock face there was a small portion of the clock missing. This was because I was only using 361 vertices for the GL\_TRIANGLE\_FAN and the last vertex and first vertex were not connected. I fixed this by increasing the for loop to include one more vertex.

**Drawing Hands of Clock**

To draw the hands of the clock I used the coordinates provided in the assignment document.

The hours hand was created using GL\_TRIANGLE\_STRIP. I looked at using GL\_QUADS and GL\_POLYGON for creating the hours hand and it actually created a 5-sided concave polygon instead of a 4-side convex polygon. I think that it has to do with the ordering of the points in OpenGL, but I am not entirely sure.

The minutes hand was created using GL\_TRIANGLES.

The seconds hand was created using GL\_LINES.

**Rotating the Hands of the Clock**

To rotate the hands of the clock I used OpenGL matrix transformations, specifically translation and rotation. First, I needed to translate each hand of the clock back to the origin of the window. Then, I rotated each hand individually at differing speeds using the formulas provided in the assignment document. I needed to ensure that each time that was taken was converted to floats since the variable that was being calculated was of type float. The rotation axis was around the z plane since I was dealing with a strictly 2-dimensional clock. After the rotation, I translated each hand back to the center of the clock face.

When I began to look at the rotations, I did not include the matrix push and pop but was able to do some research to figure out that was my main issue. One interesting thing that I learned was that it was OpenGL runs faster when using popMatrix and pushMatrix instead of loadMatrix and loadIdentity. For some reason when I tried to use loadIdentity and loadMatrix, my program would not work properly and I think that has to do with using the double buffer in this project instead of the single buffer that we looked at in class.