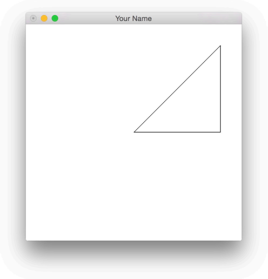
Lab #01 – CSC/CPE 474 – Parametric Curves

Today we're going to draw parametric curves in OpenGL.

Please download the code from PolyLearn for the lab and go over the code. Make sure you understand the flow of information between main.cpp and the .glsl files and what the vertex and fragment programs are doing.

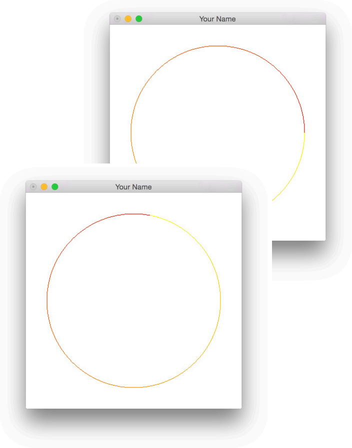
We're going to use old-style OpenGL today to specify the vertex data. This is not recommended for most cases, but is acceptable for drawing curves with a few (<1000) vertices. To draw a primitive, we use the following command:

```
        glBegin(TYPE);
        glVertex3f(X, Y, Z);
        …
        glEnd();
```



You can specify what to draw by changing the TYPE. Options include GL_POINTS, GL_LINES, etc. See http://www.opengl.org/sdk/docs/man2/xhtml/glBegin.xml for more details. Once you specify the type in glBegin(), you can add any number of vertices using the glVertex() call. To finish the primitive, call glEnd(). The provided code draws a black triangle. The color is specified with the call glColor(). Since OpenGL is a state machine, the last color specify is going to be the color of the vertex.

Task 0:
Put your name on the window title.

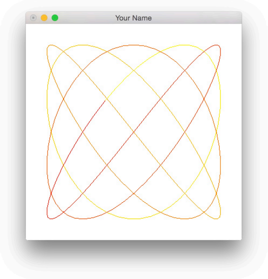


Task 1:
Use a for-loop to draw a circle of radius 1. The parametric equation is:

$$x = \cos(s)\,, y = \sin\,(s),$$

where $s \in [0,2\pi]$ is the parameter of the parametric equation. Once you have the circle on the screen, change its color so that it smoothly varies from $s = 0$ to $s = 2\pi$. (In the figure to the right, I'm using red to yellow.)

Task 2:
Use the global variable, t, to change the color over time. This global variable measures time and is incremented in the idleGL() function. When this task is complete, the circle should look like it is rotating. For example, in the figure to the right, the red part of the circle rotates counter clockwise.



Task 3:
Draw an animated Lissajous curve (http://en.wikipedia.org/wiki/Lissajous_curve).