Lab 5– 2D Transforms

This lab is intended to start familiarizing you with geometric transforms in OpenGL. Create a simple OpenGL program which draws a some-what equilateral triangle at the center of the screen (of any color).

1. Write code such that when the user clicks with the left mouse button, the triangle is moved along with the mouse. Use glReadPixel to make sure that you only apply the transform when the user clicks on the triangle (i.e. check the color of the pixel the user has clicked to make sure it is the color of your triangle). Make sure that the translation and the mouse movement correspond (i.e. the triangle "follows" the mouse and does not lag or zoom too quickly).

2. Add a keyboard event that triggers an "animation" setting. When the animation setting is active, the idle() function (which you can register a callback for using glutIdleFunc(idle()) in main), will increment the angle by which the triangle is rotated.
   For example:
   ```
   void idle() {
     theta += 1.0;
     if (theta >= 360.0)
       theta-=360;
     glutPostRedisplay();
   }
   ```
   You may need to play with the amount of the theta update settings to generate a nice smooth rotation. You will also need to add the appropriate transform calls to your drawing routines to make sure that the triangle rotates.

3. Now modify your lab such that the triangle rotation animation can be applied when the triangle is anywhere on the screen. In other words, you can rotate even after the triangle has been transformed via a mouse translation. Make sure that the triangle rotates about its center each time you start the rotation. Allow the user to apply any number of translations, then rotation animations, then translation, then rotation animations, etc.