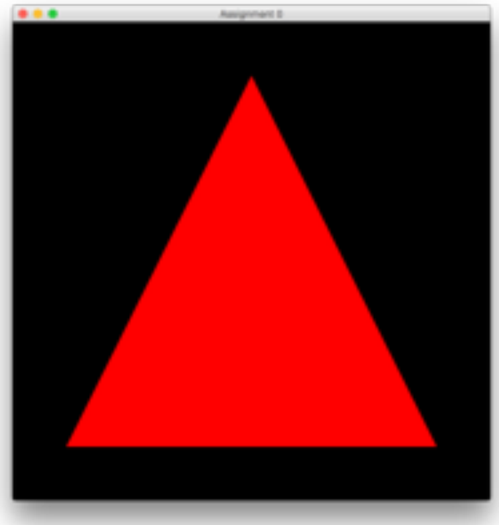
CS4052: Computer Graphics – Lab 1

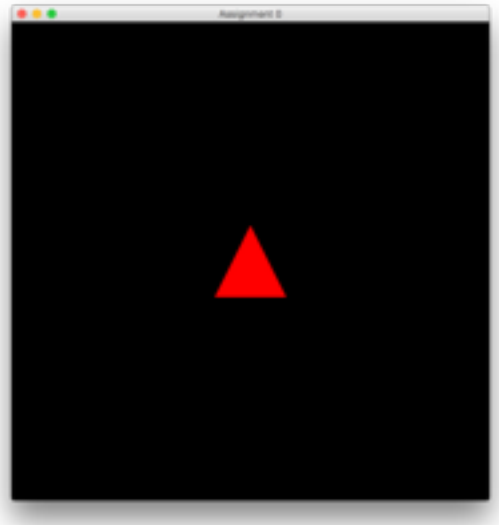
Matt Donnelly (11350561)

For this lab we were first asked to apply various different transformations to a triangle when a key is pressed. To do this I started by first modifying my vertex shader to include 3 mat4 uniforms respresting translations, scale and rotation transformations to be applied. The gl_Position output of the vertex shader is then set to the product of the translation, scale and rotation matrices and a vec4 containing the vertex position.

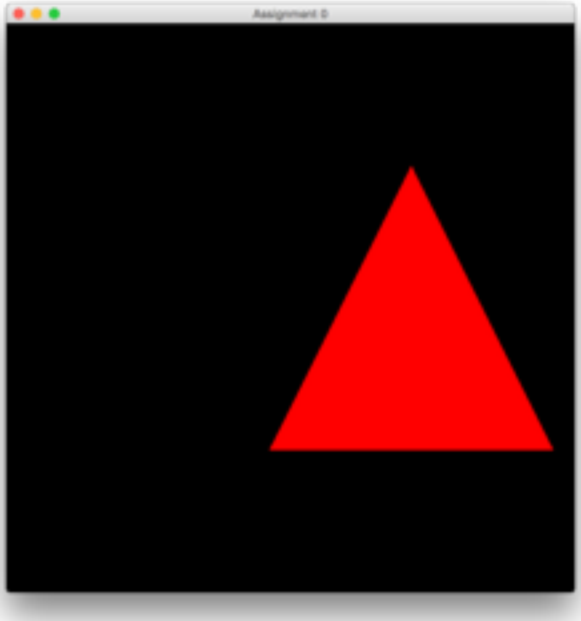
In the case of each transformation, the value affecting it will oscillate between two set values, and is controlled by a speed variable representing how much the transformation value should change by per second. This is done adding the speed value multiplied by the the amount of time it took for the last frame to draw.

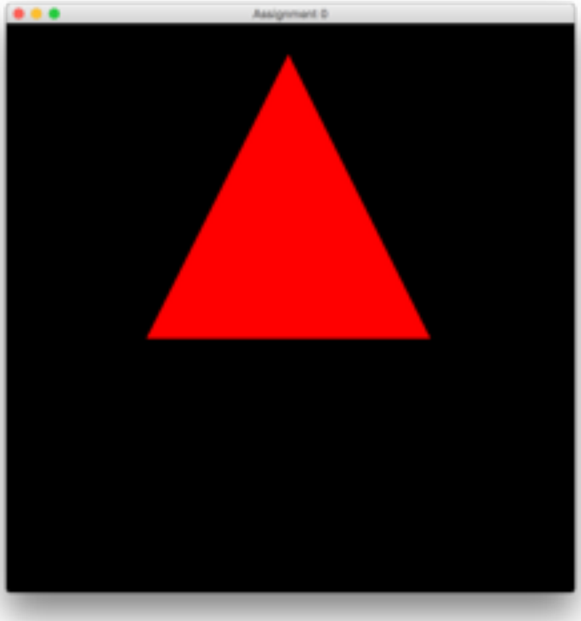
Starting with the simplest, the application of a scale transformation is done by adjusting the values along the main diagonal in an identity matrix. For example, if each value is set to 0.5 then, the triangle render at half it's size. The results of this are shown below:



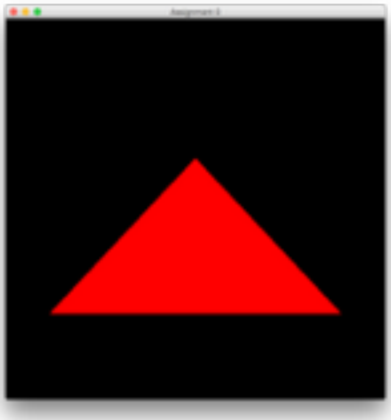


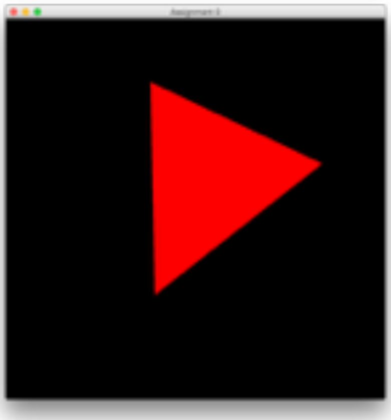
The application of a translation transformation is done by adjusting the values down the right most column of the matrix when in row-major format – although OpenGL handles matrices in column major format. For example if the top right corner of the matrix is set to 1.0, then the x-coordinate position of the triangle will be offset by 1.0. The results of this are shown below:

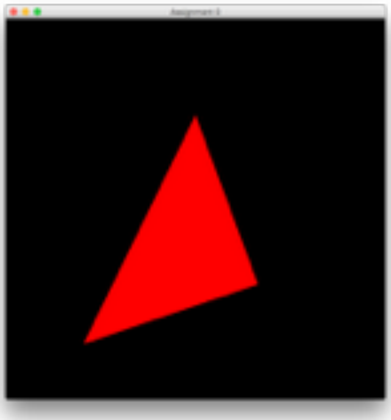




Applying a rotation transformation is done differently depending on which axis you're rotating around however, all transformations are done by setting 4 different values in the transformation matrix to values in terms of cos and sine of the angle of rotation. The results of this are show below:


Assignment 0


Assignment 0


Assignment 0