159.360: Computer Graphics
Tutorial 3: Similarity Transforms in 3D

Stephen Marsland

August 2016

1 Introduction

This is a practice tutorial. It isn't worth any marks, but it will hopefully help you get a bit more practice in WebGL, and start to see how things are put together. It is meant to be easy.

2 Description

The basic code is in `similarity.html` and `similarity.js`. Most of the code is mine, but I've cribbed a few bits from various websites.

If you load `similarity.html` then you should see an upside-down 'L' shape in red, together with a collection of sliders. What isn't clear from this view is that the shape is in 3D, since the view is straight along the z-axis.

I've done most of the work for you, in that the sliders are defined and connected up. So here are some things to do:

1. read over the code and check that you understand it
2. replace the identity matrix in each of the translation, rotation, and scaling functions with the correct matrices and use the sliders to see them working
3. remove the following 2 lines:

   ```
   gl.enable(gl.CULL_FACE);
   gl.enable(gl.DEPTH_TEST);
   ```

   and see if you can work out what is happening. This is the first place where the fact that triangles have two faces will matter.
4. use `requestAnimationFrame(render);` to put the rendering command into a loop, and make the shape rotate, initially just about the x-axis
5. add some buttons to choose which axis it should rotate about
6. make the rotation smooth by using the time to control the amount of rotation
7. add a slider to make the rotation go faster or slower