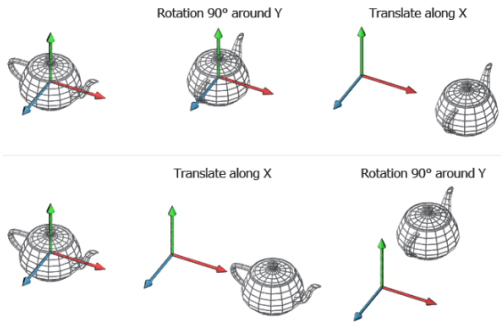
Lab #5

The purpose of this lab is to introduce you to working with transformations in OpenGL, and to get you familiar with the model/view, and projection, stages of the vertex pipeline.



1. This lab is strictly **individual** (no groupwork).

2. You are required to create a transformation matrix to move an object around the scene. Your code must use matrices to calculate the transformations and perspective projection. Details provided below.

3. Be sure to post to the lab forum and attend the in-person lab to ask the demonstrator for help.

4. Your code from Lab #4 should provide you with a good starting point. Note: the shaders have been externalised so you need to put them (simpleVertexShader.txt and simpleFragmentShader.txt) in a "Shader" folder in your project. ----------

5. Your program should have the following features:
   - Keyboard control
     a. Keypress to show: Rotation around the x- y- and z-axis
     b. Keypress to show: Translation in the x- y- and z- direction
     c. Keypress to show: Uniform and non-uniform Scaling
     d. Keypress to show: Combined Transformations

   - Virtual Camera: Create an interactive walkthrough that allows the user to move around the scene using the keyboard and/or the mouse. At a minimum, implement moving forwards and backwards, turning left and turning right.

Common mistakes:

- ***My shape disappeared!*** - check how far you are moving it; maximum -1 to 1 on x and y. Also check your matrix is being sent to the shader - all uniform variables are zero by default.
- ***Shader compilation error!*** - Trying to multiply a `vec3` by a `mat4`. Recast like this for a point: `vec4 result = matrix * vec4 (input_vector, 1.0);`
- ***My transformation is inside-out!*** - Row and column-order multiplication are mixed up. Check the order of multiplication, and compare to the layout of your matrix.
- ***My translation isn't doing anything!*** - `vec4` point has a 0 for the last component, when it should have a 1.
- ***My translation still isn't doing anything!*** - check that your matrix uniform location is correct, and is being sent with `glUniform`.

Notes

- You will need to refer to the documentation of your supporting library e.g. GLFW or FreeGLUT to find the functions and call-backs to use for keyboard input.
- Make sure that you have implemented error checks. Are you checking for shader compilation and linking errors? Do you print the logs in this case? Are you checking that the result of glGetUniformLocation() is not negative?
- Remember to read the OpenGL 4 man-pages for every function that you use from OpenGL - know what the parameters do, and what other functions they depend on. The http://docs.gl/project is a very much improved presentation of the man-pages, and has examples.
- This lab is where you may first encounter 3d geometry chaos and lose track of what you are computing. Use your pencil and paper to draw the scene. Know what the numbers should be. Use your calculator to hand calculate the transformations for the first vertex. Compare this to the computer's result to help diagnose any problems. You should be able to hand-calculate a matrix multiplication - use a cheat sheet to help you if you are unsure. I added print() functions to my maths code to help debug calculation problems. Think about the efficiency of your code - what calculations should be done in C, and what should be computed in shaders?