CS4052 Computer Graphics
Assignment 1

A. Gerdelan, Trinity College Dublin

Due date: Tue 14 Oct 2014
Course Work %: 10%

1 Outline

- You must create a **transformation matrix** to move your triangle around the scene. Your code must use matrices to calculate the transformations.
- The marking scheme provided shows the maximum marks that can be obtained for each section if completed perfectly. Merely attempting a section does not imply the full score indicated below.
- You can download my maths function header from Blackboard, use the GLM library, or create your own functions (recommended).
- Show your programme to the demonstrators on the due date and they will grade you.
- You must also submit a zip file with your **report and code** on Blackboard only. This should include a short pdf report with screen captures, as well any cpp and h files written. You must acknowledge the author of any code written by other people (such as GLM).
- This assignment is strictly individual (no group work).
- If you fail to attend the lab or to submit the report on time, you will receive 0%.
- Demonstrating a project that was not created by you is considered cheating and must be reported as such.
- This assignment should be straightforward, but leave lots of time for unexpected problems. Be sure to attend labs, and ask demonstrators for help.

2 Programme Features

Your programme should have the following features:

- Keyboard control
- Key-press to show: Rotation around the x- y- and z-axis (˜20%)
- Key-press to show: Translation in the x- y- and z- direction (˜20%)
- Key-press to show: Uniform and non-uniform Scaling (˜20%)
- Key-press to show: Combined Transformations (˜20%)
- Multiple triangles in the scene, using the same buffer but creating a new transformation matrix for each one. (˜20%)

3 Common Mistakes

1. *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.
2. *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);
   ```

3. *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.
4. *My translation isn't doing anything!*- `vec4` point has a 0.0 for the last component, when it should have a 1.0.
5. *My translation still isn't doing anything!*- check that your matrix uniform location is correct, and is being sent with `glUniform`.

4 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 assignment 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?