CS4052 Computer Graphics
Assignment 0

A. Gerdelan, Trinity College Dublin

Due date: Tue 7 Oct 2014
Course Work %: 3%

1 Outline

- Set up a working project, get it running, and complete the exercises.
- 20% of the grade is for getting the project to run, 80% is for the exercises.
- Show your working programme to the demonstrators on the due date, and they will grade you.
- Submit a report on Blackboard by the due date. The report should include a written description and screen captures.
- If you finish an assignment earlier, you may demonstrate on the preceding week's lab time, and may start on the next assignment.
- 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 Project Setup

The main objective of this assignment is to make sure that get any tricky project set-up problems out of the way at the start of the course. You can find a template Visual Studio project on Blackboard, but you should also make sure that you can create a new project from scratch. If you prefer to use something other than Visual Studio that's fine - master your build system now.

- Do you know how to include external header files?
- Do you know how to link an external library?
- Do you know the difference between static and dynamic libraries?
- Do you know where to put dynamic library files so that your executable finds them?

2.1 How to Set Up a Visual Studio Project

1. Make sure your video drivers are up-to-date, which gets the latest OpenGL libraries.
2. Create a new **empty** project in Visual Studio and make sure you choose a folder for it.
3. Make a note of whether you will compile in Debug or Release mode, and 32-bit or 64-bit arch. Your project settings are unique to this combination. Your library binaries must match this build combination.
4. Download or compile the libraries required (I use GLEW and GLFW)
5. Create a folder for your library files (close to your VS solution folder)
6. Copy the library `.lib .dll` files and into this folder
7. Copy the library header folders into this folder (called GL and GLFW)
8. In Visual Studio project properties:
   - In `configuration properties->C/C++>general->Additional Include Directories` add the **relative path** to your library folder where you put the library header file sub-folders.
   - In `configuration properties->linker->general->Additional Library Directories` add the **relative path** to your library folder where you put the `.lib` files.
   - In `configuration properties->linker->input->additional dependencies` add `glew32.lib; glfw3.lib; opengl32.lib`
9. Copy the `.dll` files for libraries to the run path of your VS solution - usually a "Debug" folder inside the solution folder
10. Create a new `main.cpp` source file with starter code
11. Put a break-point at the end of your main function so that the console stays open with any error messages if something goes wrong.

3 Exercises

- **20% of grade**. Add per-vertex colours from a vertex buffer object. These should be sent to the fragment shader, and used to create a multi-coloured triangle.
- **20% of grade**. Try to make the triangle half as big on-screen **without** altering the values copied to the vertex buffer object.
- **40% of grade**. Change the triangle to a square with 2 yellow, and 2 red vertices.

4 Notes

If you are having log-in problems with Blackboard, it is essential that you get this fixed with IS services right away.