

CPSC 314

Assignment 1: Introduction to OpenGL and Vertex Shaders

Due 4PM, Jan 24, 2013

1 Introduction

The main goals of this assignment are to setup your OpenGL environment, including installation of freeGLUT, GLEW, and GLM, and an initial exploration of the uses of vertex shaders. For this exploration you will be using a rather complicated template provided by the instructor. You do not have to look at most of this template, with the exception of the shader code (`.glsl` files) and `main.cpp`.

We will describe what is going on in the C++ code a bit later in the course. You are of course welcome to take a peek now, especially for the last part of the assignment. Many of the concepts are explained in Appendix A of your textbook, and in the web resources listed on the course web page.

When programming a shader you use a programming language called GLSL (OpenGL Shading Language). For this course we will be using GLSL version 3.30, which is indicated by adding the line `#version 330` to the top of your shaders. Make sure that any code you find while trying to learn GLSL is the correct version (or higher in this case) as there are substantial differences between the versions.

Template: The template code is found in the main assignment directory. It includes three vertex shaders, for the “axes,” “gem,” and “armadillo”. You will modify these shaders to manipulate the scene. You will not need to make changes to most source files for this assignment, except `main.cpp` (though you can if you wish to implement extra functionality). The source code is mainly used to load the shaders and the model data to display.

Execution: The `README` contains instructions for compiling and running the template, for Windows and Linux. We will try to put together specific instructions for the Mac soon.

2 Work to be done (100 pts)

First, ensure that you can compile and run the template in your preferred computing environment. See “README.txt.” Type ‘h’ to get help on the controls for viewing and other functions.

- **10 pts** Gem motion.

The variable `gem_position` in the `display()` function is changed using the keyboard, and passed to the gem shader (in `gem.vs.glsl`) using an `uniform` variable. Change the gem shader to move the gem.

- **25 pts** Color by Location.

Modify `armadillo.vs.glsl` to use the position of the current vertex to set the color of the vertex. That is to say, color code (x,y,z) coordinates using (r,g,b) colors in some way that is obvious to see.

- **35 pts** Proximity Deformation.

For this part you will need to change the armadillo shader as well as `main.cpp`. The mesh should deform to the surface of a sphere of a given radius around the gem if it is within the gem’s radius. The template code already changes the variable `gem_radius` using keyboard input. You should pass the information about the gem’s position and radius to the armadillo shader (in Section “Mesh Code”). Hint: see “Gem Code”. Use it to compute the new location of the vertex.

- **30 pts** Creative License

For this part we want to see what you can do. Your ideas should use at least one new shader, and should be of a similar complexity to the previous tasks. If you have any doubts, make sure to OK it with a prof or TA. Some possible suggestions might be:

- deform the vertices in your object in a wave over time.
- explode the model along face normals to view all the triangles that make it up.

Bonus marks may be given at the discretion of the marker for particularly noteworthy explorations.

Hand-in Instructions: You do not have to hand in any printed code. Create a `README.txt` file that includes your name, student number, and login ID, and any information you would like to pass on to the marker. Create a folder called “`assn1`” under your “`cs314`” directory. Within this directory have two subdirectories named “`part1`” and “`part2`” respectively, and put all the source files, your makefile, and your `README.txt` file for each part in the respective folder. Do not use further sub-directories. The assignment should be handed in with the exact command:

```
handin cs314 assn1
```