Assignment 4: Hello, Shaders

Computer Graphics – CS 4300/5310

Due: *March 19th, 11:59pm*

Important: I developed this assignment and solution code using Processing v2.0b8. Processing has been drastically changing their shader support from version to version in the beta. Processing 1.5 has shader support, but is not quite as easy to use. When asking for help, please make sure to specify which version of Processing you are using. It may be helpful to upgrade to 2.0b8 so that everyone is on the same page.

Educational Objectives

- Be able to describe the capabilities of vertex and fragment shaders
- Gain experience with writing short shader programs
- Understand how shaders can be combined

Assignment Description

This assignment description comes in two parts. In part one, you will be writing both a vertex and a fragment shader to manipulate and shade objects on the screen. In part two, you will use the output from those shaders as input to a third shader that will produce an embossed effect.

Part 1: Moving a Square and Shading with Concentric Circles

First, write a short Processing program that draws one or more squares to the screen. Also create the infrastructure for loading and activating a vertex shader and a fragment shader that will be applied to your scene.

- Your vertex shader should rotate the square(s) as a function of time.
- Your fragment shader should draw different-colored concentric circles that are radiating outwards from the center of the screen over time.

Your Processing code should not be performing any tasks beyond simply drawing the square(s) to the screen at a static location and communicating with the shaders. If you ever find yourself moving or shading the square using Processing, you are on the wrong track!

You should feel free to play with the effects you have at this point. The requirements above are a guideline for the amount of effort I expect you to put in. As in previous assignments, you are welcome to alter your effects to produce something more interesting or aesthetically pleasing.

Part 2: Image Processing with Shaders

Now that you have part 1 working, modify your Processing code so that it is drawing the scene to an image. Now let's use shaders to perform some image processing! You should need only one shader for this portion of the assignment.

Recall that *embossing* is an effect that can be performed by convolving an image with a filter; a 3x3 version of an emboss filter is below.

$$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Write a fragment shader that applies this filter and draw the results to the screen.

As before, you are welcome to alter this to produce a different effect. A small amount of extra credit may be given to those who demonstrate creativity, or those who create multiple shaders that we can select from at runtime (e.g. changing which shader is active by hitting a key).

Finding Other Resources
Processing ships with several example projects that use shaders. To access these examples, go to File -> Examples… and open the Shaders folder. In Processing, all shaders are stored in the "data" directory for the sketch.

There are many resources available online for using shaders in GLSL. Gamedev.net has a tutorial introducing GLSL here:

http://nehe.gamedev.net/article/glsl_an_introduction/25007/

The official OpenGL shading language reference is here:

http://www.opengl.org/documentation/glsl/ and html version here:
http://www.opengl.org/sdk/docs/manglsl/

And a little 2 page quick reference is here:

http://mew.cx/glsl_quickref.pdf

As you find other resources, especially pertaining to the shader support in Processing, please post them to Piazza so that everyone can see!

Submission Instructions
A zip file containing the following must be uploaded to Blackboard:

- Your well-documented code
- A Windows or Linux executable
- **Either a video or at least five screenshots showing your program in action**
- A README listing the number of late days you wish to use, any instructions required for running your code, and a brief description of the effect you were aiming to reproduce and the features you implemented in order to do so

Emailed assignments will **not** be accepted.