

# CSC 476

## Lab 3 - Shaders

Due: Thursday, March 10th

---

### Overview

This assignment is intended to introduce you to writing shaders in GLSL. You will be augmenting the diffuse shader provided for you to include specular lighting. To make specular lighting look good, you really need to use a fragment shader, thus we will modify our program to compute the specular lighting per pixel.

If you wish to start slow, you can start by just adding specular computations to the vertex shader by following steps in (1) or jump in and start at (2)

- (1) To get started:

Please download `csc473ShaderLab1.tar`

Note that this directory includes a complete `opengl/glut` application and `glsl` shader to draw a cube shaded with diffuse shading.

Compile and run the provided code (note you may need to make tweaks to headers etc depending on your operating system).

Now add specular shading to the color computation. This means that you need to define specular material properties, get the 'eye' position of the camera and compute appropriate lighting. Note that there is a helpful 'reflect' function in GLSL. Please use  $V \cdot R$  to compute the specular value, not the half angle approximation.

As this is just a vertex shader, you won't see a substantial change in lighting

- (2) To get started:

Please download `csc473ShaderLab2.tar`

Note that this directory includes only the `main.cpp`, you need to use the previous helper function files in conjunction with it and `glsl` vertex shader and fragment shader to draw a cube shaded with diffuse shading.

Compile and run the provided code (note you may need to make tweaks to headers etc depending on your operating system).

Notice the difference in the vertex shader – why is it different.

Notice what occurs in the fragment shader....

Now add specular shading to the color computation in the fragment shader!. This means that you need to define specular material properties, get the 'eye' position of the camera and compute appropriate lighting. You may use the half angle approximation. Play with the lighting values to make sure that you can render a specular highlight in the middle of one of the cube faces!