

# COSC4370 Fall 2020 HW3 - 3D Viewing and Shaders

Oct 13, 2020  
Due: Oct 23, 11:59 PM, 2020

## 1 Introduction

In this assignment, we will practice 3D viewing and dive a little more deeply into OpenGL by implementing the Phong shader model.

## 2 Setup

Since this homework assignment is more complicated than the last, you will need a few libraries installed on your system: GLUT (which you should have already installed for HW2), GLEW, GLFW, and GLM.

On Ubuntu/Debian, first run `sudo add-apt-repository ppa:keithw/glfw3` in order to add a repository containing the GLFW3 library.

On Ubuntu and other Linux variants, these libraries can be installed with a one-liner at the terminal: `sudo apt-get install libglew-dev libglfw3-dev libglm-dev` . (Note: on some Linux variants, the package names might end in `devel` rather than `dev`; check your distribution's package database to find the correct package.)

On OS X: GLEW: If you have Homebrew installed, you can run `brew install glew`, or if you have Macports, you can run `sudo port install glew +universal` and `sudo port install libsdl +universal` . GLFW: If you have Homebrew installed, you can run `brew install glfw3` . With Macports, you can run `sudo port install glfw` . GLM: If you have Homebrew installed, you can run `brew install glm` . With Macports, you can run `sudo port install glm` .

On Windows: You can download GLEW from <https://sourceforge.net/projects/glew/files/glew/1.13.0/> (select the download that ends in `win32.zip`) and GLFW from <http://www.glfw.org/download.html> (you will most likely want the 64-bit Windows binaries). The header-only (no compilation necessary) GLM library can be downloaded from <https://github.com/g-truc/glm/releases>. In your Visual Studio project, you will need to add the appropriate include directories and library directories for each of these libraries. Some help with this can be found at <http://www.41post.com/5178/programming/opengl-configuring-glfw-and-glew-in-visual-cplusplus-express>.

## 3 Compiling and Running the Code

For Linux and OS X, we have included a Makefile that will automatically compile the homework, assuming you have the correct libraries installed. Just run `make` in a terminal. The program that is generated is named `hw3`.

On Windows, you can use Visual Studio in the usual way to compile and run your program.

Note that the files needed for compilation include `main.cpp`, `Camera.h`, and `Shader.h`. Your vertex and fragment shader files are loaded by OpenGL at runtime; you do not need to compile them with the other

files.

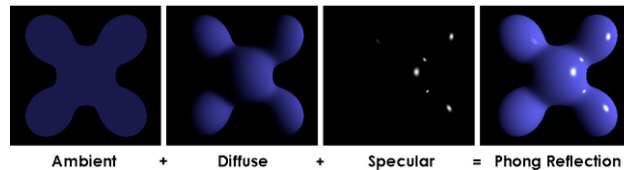
Note that the program takes no command line arguments etc. - you can just compile and run.

## 4 The Main Assignment

The goal of this assignment is to implement the 3D viewing, Phong shading model and shaders. You will write the vertex and fragment shaders for the Phong model to shade a simple cube, whose geometry is constructed in main.cpp. We also offer two facilitating files: camera.h and shader.h. You need to produce similar images in the following figure:

- image which is only rendered by using ambient light
- image which is only rendered by using diffuse light
- image which is only rendered by using specular light
- image which is rendered by a complete phong shading model

Also, you need to finish code in Camera.h and main.cpp. In main.cpp, you need to generate all uniform parameters (you need to find those parameters out, such as cube color, light color and etc. ) for your shader and pass them into shaders, please review materials of lecture "Opengl and Github". Also, similar to homework 2, please complete MVP (model view projection) matrix and pass them into your shaders (rotate your cube by using Model matrix).



## 5 Assignment Requirement

Submit all deliverables to your Github repository.

- Code for your shaders (phong.vs and phong.frag)
- Camera.h and main.cpp
- 4 images.
- details report. Please explain as detail as possible.

## 6 Academic Integrity

We have detected some of you submitted very similar homeworks. Please finish your homework independently. A zero point will be assigned if we find you copy each other. If you refer from a source, you need to attach the link in the report.