SCHOOL of ELECTRICAL ENGINEERING & COMPUTER SCIENCE
FACULTY of ENGINEERING & BUILT ENVIRONMENT
The UNIVERSITY of NEWCASTLE

Comp3320/6370 Computer Graphics

Semester 2, 2018

Lab Week 5

Jake Fountain

Version 1

1 Summary and Goals

This week's lab covers basic lighting and the effects of the different components of the lighting equation. Students are encouraged to ask questions about math exercises from the lectures and resources section. Remember, the **math is assessable** (it is in the lectures)!

2 Lighting Per Vertex

First we will look at a basic example of lighting. The simplest and fastest lighting method is computed per vertex, but achieves relatively poor results.

1. Download "LabW5_Code_2018_PerVertexLighting.zip" and **extract** the contents to a local folder.
2. Open the Visual Studio Project found at "LabW5_Code_2018_PerVertexLighting/GLFW_EXAMPLE/GLFW_EXAMPLE.sln"
3. Build and run the code. You should see a rotating box with a cat texture. Notice the sides which don't directly face the camera are darker than those facing it directly.
4. Open "LabW5_Code_2018_PerVertexLighting/GLFW_EXAMPLE/GLFW_EXAMPLE/main.cpp" and find and complete the two *TODO*s in the render loop.
5. Open the vertex shader at "LabW5_Code_2018_PerVertexLighting/GLFW_EXAMPLE/GLFW_EXAMPLE/shader.vert"
6. Read and understand how the diffuse lighting component is calculated. Complete the *TODO* and read the *NOTE*

3 Lighting Per Fragment

Modern graphics hardware has no trouble calculating lighting in real time for every fragment (read: pixel). In this section you will fill in the gaps in a fully featured per-fragment lighting shader.

1. Download "LabW5_Code_2018_PFLighting.zip" and **extract** the contents to a local folder.
2. To complete this part of the lab, you must complete the *TODO*s in the file "LabW5_Code_2018_PFLighting/GLFW_EXAMPLE/GLFW_EXAMPLE/shader.frag" Other files do **NOT** need modification.
3. Instructions:
   - In this example the light is orbiting above the block, but you won't be able to see it until you have implemented the diffuse component.
   - Complete the TODOs in order starting from TODO1
   - Run the code after you complete each *TODO* to test and observe the effects.
   - **Hint: check the console window for shader compile errors**
   - You will need the following equation for the diffuse component:

     $$i_d \propto \mathbf{n} \cdot \mathbf{l} \tag{1}$$

     where $\mathbf{n}$ is the surface normal and $\mathbf{l}$ is the vector pointing from the surface to the light.
   - For the specular component, you will need the equation:

     $$i_s \propto \mathbf{r} \cdot \mathbf{v} \tag{2}$$

     where $\mathbf{r}$ is the outgoing reflection vector of the light and $\mathbf{v}$ is the vector pointing from the surface point to the viewer.

4 After The Lab

Study the rest of the PFLighting example, including the "main.cpp" file and the "shader.vert" file. In particular note the loading of vertex normals and lighting position. Also understand the process of loading the normals and other data through the vertex shader. Read the comments if you are confused and talk to your tutor next lab if you are still having trouble.