Computer Graphics Labs

Abel J. P. Gomes

LAB. 6

Department of Computer Science and Engineering
University of Beira Interior
Portugal
2020

LAB. 6

ILLUMINATION AND SHADING OF 3D SCENES

1. Learning goals
2. Direct illumination in OpenGL (revisited)
3. Example: Illumination of a scene with two cubes
4. Programming exercises

Lab. 6

ILLUMINATION AND SHADING OF 3D SCENES

In this lab, we intend to learn how create illuminated 3D scenes.

1. Learning Goals

At the end of this chapter **you should be able to**:

1. To build up illuminated 3D scenes.
2. To use geometry normals as a fundamental step to illuminate and shade 3D objects and scenes.
3. To use the flat shading, Gouraud shading, and Phong shading as well as to distinguish between them.
4. To handle vertex and fragment shaders in GLSL.

2. Illumination in OpenGL (revisited)

Before proceeding any further, have a look at the following web links to be aware of the illumination model in OpenGL:

https://learnopengl.com/Lighting/Basic-Lighting

3. Example: Illumination of a scene with 2 cubes

An example program illustrating how direct illumination works is available at the web page of the course at:

`http://www.di.ubi.pt/~agomes/cg/praticas/lighting.zip`

In this program, the scene consists of two cubes. One of the cubes works as the light that illuminates the scene. However, only a cube (vertex positions and normals) is transferred to GPU.

Questions:

(1) Which are the objects of the scene?
(2) Which is the location of the viewer?
(3) Where is the projection plane?

(4) How both cubes are generated if only a single cube is transferred to GPU?

Remarks:

The lighting.zip code was adapted from the one available at:

https://learnopengl.com/Lighting/Basic-Lighting

so, the student is referred to carefully read this webpage to better understand how lighting interacts with objects within a scene.

Also, if the GLAD files that are wrapped in lighting.zip do not work, and as explained in the web page above, you need to use GLAD web service (https://glad.dav1d.de/) to generate such GLAD files to interface OpenGL with your graphics card driver. These files need to be added to any graphics application onwards. But, before generating the GLAD files, you first need to check which OpenGL version is supported by your computer, using GLview (http://www.realtech-vr.com/home/glview) or similar package.

4. Programming Exercises

Change the previous program `lighting.zip` to solve the following problems:

1. To get a better understanding of Phong's lighting model, move the light source around the scene over time. This movement can be accomplished using either sin or cos.
2. Additionally, experiment with different ambient, diffuse and specular light factors (or strengths), trying to grasp how they impact the result. Also play around with distinct shininess factors.
3. Change Phong shading from world space to view space.
4. Replace Phong shading by Gouraud shading. Doing that correctly results in a bit off lighting (particularly the specular highlights) on the cube object.
5. Change the properties of the object's color over time. This is equivalent to change the values RGB of the object in the scene. Note that reds, green, and blues are in the interval [0,1].
6. Now change the properties of the light's color over time. This is equivalent to change the values RGB of the emitted light of the light source in the scene. Once again, reds, green, and blues vary in the interval [0,1].
7. Add a new object to the scene (e.g., a new cube). Now the scene has tree cubes, one of which is a light source. The first cube's material is copper, while the second cube's material is jade. Loot at http://devernay.free.fr/cours/opengl/materials.html to know the RGB values for each material.
8. Now, add a new light source that emits red light. What happens?
9. Finally, add a gold sphere to the scene, using an .obj loader. There are various .obj loaders available over the web, namely:

http://www.di.ubi.pt/~agomes/cg/praticas/cap2-cookbook.zip
https://github.com/tinyobjloader/tinyobjloader