CSUS COLLEGE OF ENGINEERING AND COMPUTER SCIENCE
Department of Computer Science

CSc 155 – Advanced Computer Graphics
Fall 2018
Dr. Muyan

Assignment #3 – Lighting, Materials, and Shadows

Due Date: Monday, November 12th (11:59 PM)

You are to create a world containing at least three different objects made of different materials and textures (use at least two different materials and two different materials), one global ambient light and one user-controllable positional light, and a user-controllable camera. At least one object must cast a shadow on at least one other object in the scene.

Unlike A2, objects do **not** move. However, in addition to moving the camera around, you will add controls for moving the positional light around. You can reuse some of your code from A2, such as key bindings, camera controls, XYZ axes. You can make a scene similar to that of A2, or an entirely different scene. Be creative!

Program Requirements

- **World Objects**

The world should contain at least three different objects. At least one object must be imported from an external model. You can use the posted obj loader and one of the posted obj models, or a model from another source, or you can build one using Maya or Blender. Include XYZ axes that can be toggled on/off by SPACE bar as in A2.

Use at least two different textures. You can use textures created by you and/or outside source.

- **Lighting**

The world is to be lit by: (1) a global ambient light that is always on, and (2) a positional light that can be toggled on/off by the “t” key. The positional light must be moveable with a small amount via the following keys:

o – move the light forward

k – move the light backward

j – move the light left

l – move the light right

p – move the light down

i – move the light up

Your shaders must implement either Phong or Blinn-Phong reflection on all displayed objects for global ambient light and the positional light (when enabled). It isn’t required to account for distance attenuation, although you may add that if you wish.

The shader must also blend lighting and texture colors for objects so that the textured objects also show the effects of lights.

Draw a dark yellow point (or something similar) at the location of the positional light.

- **Materials**

The program must use at least two different *materials* for rendering world objects. You may obtain your material definitions from any source (such as the table in the lecture note slides, although you are encouraged to find material definitions from other sources as well). You may use the **`graphicslib3D.Material`** class to define your materials, or you may implement an analogous class of your own. The **`Material`** class already has three pre-defined static materials in it, and you may use those for *one* of your two materials.

- **Shadow-Mapping**

Use the standard shadow-mapping technique such that at least one object casts a shadow on at least one other object. The shadow should move appropriately as the user moves the positional light around. Make a reasonable attempt to reduce shadow-acne.

- **Viewing**

The user must be able to manipulate the camera using the same key bindings used in A2 (i.e., w, s, a, d, e, q, left and right arrow, up and down arrow).

Additional Notes

- Make appropriate use of mipmapping and anistropic filtering to avoid artifacts.
- Your code must be contained in a Java package whose name is exactly "**`a3`**" (lower case). As before, the "main" class in your program be named exactly "**`Starter`**". It should be possible to run the program from a command prompt in the parent directory by typing the command:

```
java –Dsun.java2d.d3d=false a3.Starter
```

Deliverables

Submit to Canvas **TWO files (zip file and txt file) SEPERATELY** (i.e., do NOT place the txt file inside the zip file). The *ZIP file* should be named as YourLastName-YourFirstName-a#.zip (e.g., Doe-Jane-a3.zip) and should contain:

(1) your Java source files, compiled (.class) files, and GLSL shader files
(2) your texture image files
(3) your model files
(4) a screen capture (.jpg) of your program running

The submitted files must be organized in the proper hierarchy in the ZIP file; that is, the .java and .class files must be contained within a subdirectory named "a3".

The TEXT file (i.e., not a pdf, doc etc.) should be named as readme.txt and should list:

(1) explain which objects in your scene satisfy which requirements
(2) source information for each texture and model you used
(3) additional information you want to share with the grader
(4) the lab and the lab machine you have used to test your program

You will receive the grader comments on your text file when grades are posted.