Assignment #3 – Lighting, Materials, and Skyboxes
Due Date: Friday April 14th [4 weeks]

Create a world containing one or more movable lights, a skybox, a user-controllable camera, and at least three objects made of different materials and textures,. At least two of the objects must be moving and be organized in a matrix stack as a hierarchical system.

In addition to moving the camera around, you will add controls for moving a *light* around. You can reuse some of your code from A2, such as key bindings, camera controls, XYZ axes. Your scene can be similar to the one you made for A2, or it can be different, but in either case there must be some substantial change. Try to be creative!

Program Requirements

➢ **World Objects**

At least one object must be imported from an OBJ file not distributed in class. Use the OBJ loader from the book, and an OBJ model from the web or another source, or you can build one using Maya or Blender. Include XYZ axes that can be toggled on/off as in A2.

Use at least two different textures. Textures must be rendered so that they also show the effects of lighting, blended as described in Chapter 7.6 (choose which method works best for your scene). Access texture image files and OBJ files using only *relative paths*.

➢ **Matrix Stack**

At least two of the objects must be organized using a matrix stack, in which the parent object is moving. This should cause the child object to follow, or be attached to, the parent. Use the JOML `Matrix4fStack` class to manage the MV matrix used for the moving object and the child object. This should require only one `Matrix4fStack` instance.

➢ **Lighting**

The world is to be lit by at least: (1) global ambient light that is always on, and (2) a positional or spot light that can be toggled on/off by a button or key. The positional light must be moveable via the mouse, with controls for left, right, forward, backward, up, and down. Use either dragging, or the mouse wheel. (document this in your readme file)

Your shaders must implement Phong or Blinn-Phong reflection on all displayed objects (when lighting is enabled), except for the skybox. Distance attenuation isn't required.

Draw a yellow dot (or something similar) at the location of the movable light, unless there is already an object for the light. You can add a key that removes the yellow dot if you like.

➢ **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 book. You may also use one of the statically-defined materials (gold, silver, bronze) in the `Utils` class on the textbook's CD. At least one of your material definitions should be from some source other than the textbook/CD.

> **Skybox**

Your scene should include a skybox. It may be from the book, or from the web, or one that you create. Note that it must be a skybox, <u>not</u> a roombox. It should be implemented using an OpenGL cube map. You may use the code in the Utils.java file on the accompanying disk. Do not apply lighting to the skybox. Test your skybox by moving the camera around, to make sure that all six sides are properly positioned relative to each other.

> **Viewing**

The user must be able to manipulate the camera using the same camera controls as in the previous assignment (A2). You may choose to restrict the camera (and light) movement controls to within a range that is logical for your scene. However it must be possible to view all six sides of the skybox.

Additional Notes

- Make appropriate use of mipmapping and anisotropic filtering to avoid texture artifacts.

- Document all user controls (for moving the camera and the light) in your readme file.

- Your program must be contained in a Java package named "`a3`" (lower case). Previous requirements for compiling and running your code remain the same.

Deliverables

- This is an INDIVIDUAL assignment.

- It may be submitted up to 24 hours late, with a 10% penalty.

- Submit to Canvas ONE item:

  > a ZIP folder with your A3 submission, as described below

    (it is <u>not</u> necessary to include an additional TEXT file this time)

- The ZIP folder contains the following:

  > .java, .class, .obj, GLSL, and texture files organized in the proper folder hierarchy

  > a <u>.PDF</u> report file consisting of the following eight <u>numbered</u> items:

    1. a screenshot of your running program, showing as many features as possible

    2. a description of your lighting – the types of light(s) and initial location(s)

    3. a list of controls for moving the camera

    4. a list of controls for moving the light

    5. a description of where in your scene the matrix stack was utilized

    6. a list of which requirements you <u>were NOT</u> able to get fully working

    7. source and licensing information for each texture and model that you used

    8. indicate on which RVR-5029 (remote) machine you tested your program