**Ashley Hutton**

**KUID: 2764001**

**11/10/17**

# EECS 672

# Programming Project 3

### Summary

The goal of this project was to create and render a 3D scene using the techniques we have learned in class. Specifically, we had to make subclasses of the ModelView class to create various 3D objects, and implement the Phong lighting model.

### Scene Description and Background

My scene is of a standard living room. It contains two couches, two side tables with lamps, a coffee table, two bookcases with books, and a TV. Originally, the scene was going to model my personal living room, but then I realized I didn't have enough unique objects to create at least 3 ModelView subclasses. The idea is not necessarily "something of special interest" to me. However, I do enjoy sitting, and reading, both of which can be done in this living room.

### Model Creation

The model started on a piece of paper. As I drew the elements of my scene, I was able to decide what objects were challenging, but still manageable to create (given my current skill).

The first object I created in OpenGL was a couch. This was a mistake and I will speak to this in the "difficulties" section. I backtracked and restarted by first placing my floor, and then went back to create the couch.

In the main function, I call several helper methods to establish the geometry. First: the walls and floors. Next I rendered the couches, then the tables with their lamps and then I rendered the two bookshelves, which I filled with books.

### Project Specifications

Following the "Project Requirements" section, I met specifications by:

- Implementing the Phong lighting model by adding the instances variables "ka" and "la" to go along with the already existing "kd". From there I adjusted the shader to use the Phong lighting model.
- Created multiple subclasses (and did not create instances of "TEMPLATE_SubClass") to represent different objects. At least three of those objects were not simple geometric shapes:
    - Couch
    - Bookcase
    - Table
    - Lamp
- Completed the ModelView_Additions.c++ file with the methods needed for project 2
- Produced at least two instances of the various concrete ModelView subclasses:
    - Two couches
    - Two bookcases

    - Three tables
    - Two lamps
- Set 3DViewingInformation to specify an eye, center and up that made my line of sight not parallel to a primary axis

*Update for project 3:*

- All ModelView subclasses from project 2 were converted into SceneElement subclasses
- SceneElement method bodies were completed
- Initial line of sight is parallel to a model coordinate axis
- Dynamic rotation, panning and zooming is supported
- The Phong local lighting model is implemented in the CPU and GPU
- Additional geometry has been added for project sophistication:
    - Couch armrests
    - Couch pillows

**Difficulties**

A recurring problem I had throughout the project was trying to establish a standard implementation for my geometry. Since the modeling was new to me, I was unsure at the best approach at the beginning, so I just found brute force solutions. After some practice I was able to recognize good and bad implementations for a particular object.

As I mentioned above, the first object I tried to create was the couch. This was a nightmare because I wasn't thinking "big picture". Instead, I was just trying to figure out how to get two rectangles put together to make a couch. Once I made my first successful couch, I realized I had made it very difficult to replicate or change. To make a new couch, I have to specify the coordinates of each individual vertex on the couch. For future objects, I was careful to make their constructors as independent as possible of the actual placement and direction of the object.

Another difficulty was understanding how to change the eye, center and up coordinates and projection parameters to give me a view that I wanted. I used metaview to help with this problem, but I still need to review the details to have more comprehension.

*Update for project 3: Most of these problems were solved using the BasicShape class. My code is a lot cleaner and more modular now.*

**Uniqueness**

I tried to put detail into this living room to make it realistic. All of the books on the shelves are different sizes and colors, and one was left out on the table as someone might do if they had taken it out to read. The floors also have baseboards and crown moulding. These are small details that do not necessarily reflect anything spectacular in code, but I think attention to detail is important in software projects because it shows thoughtfulness (which is easy to miss out on).

**What is new in Project 3**

*Features*

There are several new features present in project 3. The user of the application can dynamically view the model by rotating (left click), panning (shift+left click), or zooming (mouse scroll).

Additionally, the lighting model has become more sophisticated. Instead of assuming a single directional light source with maximum intensity, a full Phong lighting model implementation has been added. The two lamps in the scene act as positional light sources, and there is a third directional light source to help illuminate the rest of the scene. The lamps have the same light intensity, and the directional source is a bit dimmer to show off the positional light sources. The light from the lamp is coming from inside of the lamp shades, at about the middle of the cylinder.

Along with dynamic viewing, and the updated lighting model implementation, there are new geometries in the scene for project 3. Armrests have been added to the couches, and a few decorative pillows have been placed.

*Difficulties*

The most difficult part of the project 3 implementation was the Phong lighting model. Most of the problems were minor errors that had large repercussions. In my opinion, the shaders are the most difficult part of the application to debug.

*Implementation Changes*

The BasicShape class was very helpful for project 3. In project 2, the most difficult geometry was the couch. This time around, I refactored my Couch class to use BasicShape. I was able to add the armrests easily to make the couch more realistic and detailed.

I created the pillows using the BasicShape class, and I updated the shades on the lamps to be BasicShape cylinders instead of custom ones.

All classes were properly updated to inherit from the SceneElement class. Doing so eliminated all shaderIF instance variables as well as material property instances variables. The material property information is now stored in the PhongMaterial class, which the SceneElement contains an instance of.