# EECS 672
# Project 3 Report

Noah Benham

November 24, 2015

# 1 Project Inspiration

I decided to continue building upon my existing models, as I felt my existing scene would benefit greatly from an improved lighting model approximation. As stated in my project 2 report, I've always been interested in the outdoors and nature, but wanted to create a scene that merged the outdoors with indoor geometry. I originally planned on creating a picnic on a grassy lawn with a blanket and related geometry, but eventually decided upon a picnic table with an umbrella.

This picnic table scene loosely mimics my family's backyard. My family frequently eats dinners outside, which has a setup very similar to my project. These dinner also almost always include our two dogs, so I've included an obligatory dog toy object as well. While not an exact replica, I've intentionally setup my scene to allow generous expansion in future projects, such as a possible project 4. This could potentially include objects such as trees, bushes, and potentially the backside of the home, or other aspects such as transparency.

# 2 Model Generation

My model is broken down into subclasses, and further subclasses from there. My Project 3 code has only a couple of additional models, as most of my complexity comes from the lighting model implementation and creation of new objects such as a sphere, from which I generate a bowl object. My project2.c++ class includes several models such as a Floor, Table, Chairs, Glasses and an Umbrella. Each of these classes in turn use either a block, cylinder or combinations of both to create their geometries. Project 3 introduced a new bowl object, with a newly created sphere as its base object.

Each of my models have constructors that provide functionality to modify their values according to their design. For example, the Chair class accepts xMin, yMin and zMin parameters, as well as xLength, yLength, and zLength. Contrasting this model, the Glass model only accepts staring x, y and z positions, as well as a radius. For models that accept lengths, it's important to note that I've taken liberty to ensure they expand correctly. For example, if a chair is resized, the legs, base, and all back components will all resize accordingly.

# 3 Project Specifications

My main source of documentation was `http://people.eecs.ku.edu/~miller/`. Specifically, I found searching the class website to be particularly useful in finding relevant review material and code snippets that inspired my Project's design and allowed me to meet specifications. I believe my lighting model would not be entirely accurate without the office hours of which I took advantage.

Since all of the specifications were posted on our project assignment page, meeting requirements was not difficult. I started project 3 earlier than previous projects, which I found to help greatly, as project 3 had extensive nuances, specifically with regard to the lighting model implementation, at least in my case. Every time I complete a Graphics project, I feel my understanding of the code grows substantially compared to my understanding gained from attending class, since actively participating in the coding process is immensely helpful to the learning process.

# 4 Interactive Controls

Project 3 support scrolling in and out with the mouse scroll wheel. Scrolling up will zoom in upon the scene, while scrolling down will zoom out. Additionally, pressing the left mouse button down and moving the mouse around the scene will move the viewer around a fixed center point in the scene.

The 'P' (perspective), 'O' (orthogonal) and 'Q' (oblique) keys will change the perspective, respectively. The view may need to be re-oriented very slightly after striking one of these keys to adjust to the new perspective, such as scrolling in/out or moving the viewer around the fixed center point.

I also created a debugging function that allows the user to strike the 'A' 'S' 'D' 'G' and 'T' keys to move the eye around the scene. However, I did not adjust the MC region of interest, resulting in the scene being clipped. Thus, I did not include the function in my final submission.

# 5 Difficulties

While rotations were definitely one of the most difficult parts of project 2 for me, this project posed different challenges. Many of my previous difficulties were overcome with the cryph utilities once I learned how to use them. With project 3, most of my difficulties came in getting my lighting model implemented successfully. While the Phong Lighting model equation looks fairly straightforward, there are many variables and computations that need to be checked with if statements, and in some cases modified in order to represent the scene accurately.

I also had some trouble getting the handleCommand method implemented, as my initial definition of the method wasn't compiling correctly. Even after correcting it, I had to manually remove the existing .o files generated for the mvcutil files before my project3 would compile. Thankfully, I had already created a makefile for project3 which made this process easy to replicate for the mvcutil files.

I found the scene manipulation/interactive viewing relatively easy to implement. The methods were not complex, compared to others such as the lighting model. However, I did decide to flip a couple of controls such as the mouse scroll wheel and the x-y controls when pressing the mouse button for more intuitive behavior.

# 6 Unique things

One of my goals in creating this picnic scene was to display objects that would show the lighting in a unique way. My inclusion of an umbrella that displays both the top and bottom of the object highlights the difference in lighting between the two different object sides. Similarly, the bowl objects on the table show specular lighting well. In order to improve their appearance, I moved a great majority of my code from the vertex shader to the fragment shader - most importantly, the lighting model code. Ideally, I would've liked to be able to cast a shadow from the object, but unfortunately our capabilities are slightly limited in that regard.

One of the things I enjoyed was searching for RGB colors that more accurately described what the objects truly were. I've even included some of the more "colorful" names in my code. As noted previously, for models that accept lengths, I've taken liberty to ensure they expand correctly. For example, if a Chair is resized, the legs, base, and all back components will all resize accordingly. This required significant computation and testing to ensure correct resizing.

Another important thing to note is the way the sphere object is drawn. This class serves as the base object for my bowl object. Since I'm drawing it in triangle strips from the bottom of the sphere up, I'm able to very easily draw only half of the triangle strips and have an even top to the shape. I believe inclusion of this object increases my project's complexity greatly. While I didn't include a great number of new objects, I took great care in insuring my lighting model was accurate and the object's coloring represented them well.

While commented out in my final submission, I found moving the center point to be very helpful in navigating the scene. If interested, one could easily re-enable it to include the functionality after adjusting the MC region of interest issue as discussed previously.