Austin Bailey
EECS672
Project 3 write up
11/10/2017

In order to satisfy the sophistication requirement for the project I implemented rotations into the scheme of the rubiks cube. There are fourteen supported features as follows:

u: rotates top clockwise
U: rotates top counter-clockwise
d: rotates bottom clockwise (from the perspective of the viewer beneath the block
D:rotates bottom counter-clockwise
f: rotates front clockwise
F: rotates front counter-clockwise
b: rotates back clockwise
B: rotates back counter-clockwise (from the perspective of the viewer behind the block
r: rotates right clockwise
R: rotates right counter-clockwise
l: rotates left clockwise
L: rotates left counter-clockwise

s: shuffles block by randomly doing five clockwise rotations. The rotations are printed in the t terminal and holding shift and doing them in reverse will result in a solved cube (proves that the cube maintains its solvable property through all rotations)

S: solves cube

To access a given feature, press the corresponding button while the project runs. The logic implementation of the rotations can be found in the rotate function of RubiksCube.c++ lines 232 to 785, and the directionality function can be found from lines 804 to 969. The code is very dense and repeats itself on occasion, one of my goals for project 4 is to make it more concise and readable.

There are a few mistakes in the lighting model that I couldn't fix- mainly the directional light does not rotate with the scene. The interactive viewing matrix combinations seem to be correct – they were mostly taken from the class notes.

The most difficult part for me was the lighting model. It was very frustrating because I knew all of the logic and equations from class however my fragment shader gave me a gray screen and a bunch of errors the first few times I implemented the computations. A single mistake in the fragment shader causes the entire project to gray out, and because its not compiled there is no indication as to where the error is. My final solution was to start again from scratch, and do each of the parts separately, running the shader after every change. It was very difficult. The second most difficult part was the logic for the rubiks cube, although once I got past the learning curve, it became more time-consuming and tedious than intensive.