Final Project – Kitchen Table

By: Brandon Rickman

CS330 – Computer Visualization and Graphics

Dr. Loay Alnaji

Southern New Hampshire University

August 19, 2019

**Description of Project**

The kitchen table project was a functioning C++ computer program that utiziled the OpenGL API, GL math models, and SOIL2 image libraries. This project started out as a mug and turned into a kitchen table.

**Development Challenges**

At the beginning of this course we chose an object around the house that we could then model in our computer program. I chose to make a metal mug. As I started working towards created the primitive shape (the polygons that make up the faces of the mug) I was met with many challenges. Most, if not all, the code examples I was able to reference were based on Legacy OpenGL. This presented the challenge of how to convert the code into Modern OpenGL. Crunch for time and almost two weeks late I settle to change the object to a kitchen table. This was solely based on the idea that I know how to draw triangles and squares but not cylinders and circles.

The other area I struggled was with SOIL2. After ensuring I was able to complete the practice tutorials and render graphics onto a cube, I set out to do it again on a triangle. However, it was not working. I continued researching solutions to my problems, watching videos, and trying to determine where my problems existed, but was unable to come up with a solution, until I tried creating a completely new project in Visual Studio 2017. I set up the SOIL2 packages again (this time based on another setup procedure), and copied the code into a new source file. That’s when it all started working. When I moved over to my final project, it ran the first time! This meant that most of my previous errors were caused by the incorrect setup of SOIL2.

Through all the debugging, changing, training videos, books, online code snippets, and tutorials I researched, I can proudly say that I understand 90% of my code and what it is doing or where it came from. This course has been difficult; however, I fell I have learned so much from this course, not just with OpenGL, but with C++, Object Oriented Programming, development phases, set up of environments, training and reviewing others code, and even problem solving. There were so many times I wanted to pull my hair out or break my computer, because I could not understand why it wasn’t working. These were the times I would step away and take a break.

**Scene Navigation**

* Holding the ALT key and LEFT mouse button the user can move the mouse to rotate the camera around the object.
* Holding the ALT key and RIGHT mouse button the user can move the mouse UP and DOWN to zoom in or out on the object.
* Pressing the SHIFT key will change the view perspective.

**Custom Function**

Due to several programming issues I did not stray too far from the tutorial defined functions. However, I did change the code block inside functions to suit my needs. For example, the function UMouseMove. I edited this function to determine where the mouse pointer was once called. Then using some math calculations determined x and y values as necessary to track the mouse. I separated these calculations into IF block statements for determining the direction the mouse was moving and set the view calculations of how I wanted the model/camera to react to these button presses. This function could easily be transferred from one project to the next, as I did when moving from the practice problems to the final project.