**Final Project Reflection**

**Aaron McDonald**

**04/14/2024**

For my final project for this course I was recreating a scene that I took of my desk with a few simple items on it. Here is the original picture:

A calendar on a wall

Description automatically generated

The image that I was able to recreate within OpenGL is as follows:

A calendar on a table

Description automatically generated

For this scene I took some creative liberty in how the colors differed as well as how some of the items appeared in the scene. Firstly, I created a red table for the scene instead of the black table because I wanted to contrast against the black default background of the OpenGL world. Furthermore, items such as the cup ended up with a different texture due to the inability to source a proper texture to provide the cup with. Additionally, all of the pens were removed from the cup to provide simplicity in recreating the scene as this would have added many additional steps to get all of the pens at the correct angles and for the sheer number of pens it made more sense to simply have an empty cup.

As for creating the objects, I used all simple shapes to keep the polygon count at a low amount. This makes things much easier to render as the computer is generating far less triangles. If I needed far more detail for any of these items then I would simply provide a regular map or a bump map to add extra lighting effects, such as those I have provided to many of the materials in the scene for realism.

The four highlighted objects in the scene are each complex objects, comprised of several simpler objects. Starting with the cup, it is comprised of a cylinder, half torus, and half sphere. Each of these builds the cup to represent the realistic version more closely without creating too many polygons. The cup has a rough texture to match real life more closely while also having a material that is very diffuse and not very reflective. The calendar is comprised of two planes with two separate image textures to create the look of the calendar from my photo. In this case the material is a shinier material to reflect light more similarly like the real version.

The notebook is comprised of many boxes all of different scales and rotations to represent the pages and the binding of the book. The book in the original scene was open so I wanted to copy that here with the left side being angled downward onto the table. The textures used on the book allow the sides to look like stacked paper and the top looks like the lines of a notebook. I was able to do this by choosing the render each side separately, thus allowing me to choose the needed texture for the appropriate side of the book.

Finally, the pen is comprised of a cone and several cylinders, all of a smaller and narrow scale to represent the actual size of the original pen. There were no textures applied to the pen as it did not seem necessary to represent the pen, so instead I opted to use more intense materials with a shader color. The plastic and metal on the pen each reflect light differently to closely resemble the real-life version. The metal in particular is highly reflective, making it shine just like the metal on the real-life pen.

For scene navigation a user may use the WASD keys to pan around the scene as well as the mouse scroll wheel to increase the camera’s speed. Furthermore, to move the camera up and down the user may use the Q and E key. Finally, the user may use the O and P keys to change the camera into orthographic or perspective projection respectively.

Finally, the design of my software allows for reusability because each object that gets rendered is placed in it’s own function. This ensures that if the object needs to be rendered again the function can simply be called to render the object again. Furthermore, many of the meshes in the scene are pre-generated into the code. This means that to draw a new version of this object you simply have to call the appropriate draw function for that mesh, while also being able to modify the mesh dramatically by changing the location, scale, and rotation of the mesh when it is drawn. Each of the rendering functions for the different aspects of the rendering pipeline are also modular and placed in their own function, ensuring that similar to the object rendering functions these may also be called for reusability where necessary.