**Final Project Reflection and Design Decisions**

My 3D scene is the representation of the 2D image of a few items on the kitchen counter. Except for the spoon which requires two primitive shapes to re-create this complex object, the rest of the objects present in the picture (scene) are basic primitive shapes. Since all the objects on the scene were on a table, I used a plane to re-create the table which is simply a square. You can think of it as only one side of a cube. It is that simple. To re-create the 3D scene, I decided to use spheres to represent eggs in the scene. Although the sphere is circular and not an oval shape, it is the best shape to use for creating an egg. I use a rectangular cube to re-create the butter which I believe after the plane was the simplest shape in the scene. Creating the bowl was a little more challenging. It is half of a sphere, so I had to find out a way to cut the sphere in half. To accomplish that I change the formula to draw the sphere to only draw a sphere between the range of zero and half the PI. The most challenging object in the scene was the spoon because, in addition to using two primitive shapes to re-create, these two shapes must place in a way relative to each other to represent the full shape of a spoon. Although I could use either an elongated cube or an elongated cylinder for the spoon’s handle, I decided that an elongated cylinder would be much more pleasing in the scene. I draw a half sphere for the head of the spoon and an elongated cylinder for the handle of the spoon. Then I used appropriate transformation matrices to place these two shapes in the scene correctly to shape the full spoon.

The program is fully capable of receiving the keyboard, mouse, and combination of both input commands as well to provide an immersive scene for the user that they can navigate around it. On the keyboard, the user can use the keys WASDQE to move close, left, far, right, up, and down respectively. With the mouse, the scroll can be used to zoom in and zoom out as well as the directional mouse movements to move around the scene.

Three specific header files contain formulas and functions that help to draw some of the primitive shapes in the scene. These header files are cylinder.h, Sphere.h, and HalfSphere.h. In addition to those, four major custom functions handle different functionalities of the program. Since these custom functions have their entire code outside and separate from the main body of the program, they can be modified easily and add extra functionality to them. These custom functions are as follows:

* framebuffer\_size\_callback(): this function set the correct size of the window for the user (viewport window). Any time that the user changes the size of the window, the program calls this function and provides it with the current parameters to set the appropriate size of the window for the viewport window.
* mouse\_callback(): this function handles the mouse movements in the scene. This function reads and gathers vertical and horizontal positions of the mouse constantly and adjusts the camera position in the scene. It also ensures that the pitch stays within the program’s acceptable range of -89.0 to 89.0.
* scroll\_callback(): this function handles the mouse’s scroll wheel movement. The mouse’s scroll wheel can be used to increase or decrease the field of view in the window. It ensures that the field of view stays within the program’s acceptable range of 1.0 to 45.0.
* processInput(): this function handles the keyboard’s input. There are certain keys on the keyboard that the user can use to interact with the window. Each time one of those keys on the keyboard is pressed, the function adjusts the position of the camera based on the position that has been defined for the specific key.