**Design Decisions and Reflection Paper**

The 3d scene I created represents 6 different objects in the picture. The first object is a plane that represents the floor, or the surface that the other objects rest upon. The second object I have is a sphere that represents the fake lemon in my picture. The third object in my scene is a cube that has been elongated to represent a blue notebook that three other objects rest on. The fourth object is an icosahedron that represents a D20 dice. The fifth object in my scene is a pyramid that represents a D4 dice. Finally the last object is made up of two pyramids and represents my complex object. The complex object is a D8 dice and also has two textures applied. Every object in the scene has a texture applied, but the complex object has two textures applied to represent a D8. I created all of the textures used in Affinity Photo, with the exception of the marble texture on the D8. I gave credit in the code and the texture is under the creative commons license. All of the shapes I chose are a good approximation of the real-life objects.

To program the shapes I specified the vertices for all of the objects, except the sphere. I then passed the vertices to the objects to create the mesh. After the mesh is created I pass the objects to a render function to draw the object on the screen. When the object is rendered a texture is given to apply to it. The complex object takes two textures as an argument for both of the pyramids. The sphere is different and is able to calculate its own coordinates from a precision level that is passed to it. The sphere is drawn in bands across the X-axis that move up along the Y-axis until the full circle is drawn. After the calculations are done I then combine the vertices, texture coordinates, and normals into one vector. The vector is stepped through with a specified stride length and each item is passed to the shader to draw the sphere.

The 3d scene starts in a similar position as the picture I chose, but the user is also able to navigate the 3d space. The W,A,S, D, Q, and E keys are used to translate the user across 3d space. The W and D keys are used to move on the Z-axis. The A and D keys are used to move on the X-axis. The Q and E keys are used to navigate on the Y-axis. The user is also able to move their viewpoint with the mouse. The mouse will control the pitch and yaw of the view. The speed at which the user navigates the scene may be controlled with the mouse wheel. The last item to note is the P key, which will change the view from perspective to orthographic and back again.

Each of the devices are polled in a callback function and updated with each frame. The frames are also calculated in delta time so that the speed does not change and remains consistent. The perspective is calculated and passed to the shader so that it may properly render the scene for the viewer's perspective. I also have added future functionality to properly handle mouse clicks. The mouse clicks do nothing but output text to the console at this time, but their inclusion means that it is possible to add functionality to them easily.

The final item I want to discuss in this reflection is how I organized my code. It would have been possible to code all of this in one file and run it, however it would be very messy and hard to navigate. I started to modularize my code fairly early in the process. I wanted to make it easier to update and add functionality. I moved the shader and rendering to one file. I made all 3d objects and the camera objects so it is easier to create and add more. There are files and objects for all of my 3d shapes. Each of my functions and objects are able to be used by multiple items. An example of this is my overloaded render function that is able to take in any of my objects and render them. It also takes in the pyramid object with one or two textures passed to it.

I want to continue to work on this assignment in the future. I have left notes and comments for myself for that reason. I had a lot of fun making this entire project. There were times where it was difficult, but ultimately I had a lot of fun and plan on returning to this to work more on openGL.