* Through the course, I gained knowledge on multiple techniques for producing shapes, implementing textures, and displaying lighting effects. Using vertices and coordinates, I can now create intricate shapes using OpenGL. As a result of my newly acquired skills, I was able to design a pen for writing, an apple, a wristwatch, and a wooden table to hold these objects.
* To ensure clarity and recognition of the objects, I applied different textures to each of them. For instance, I utilized an actual die texture with minimal light changes to make the die look natural when repeated. Similarly, the pencil had a pencil texture which, when replicated, looked a little odd, but still looked like a pencil. To contrast the brown leather strap, I used a silver metal body for the watch dial, I kept it simple by applying a metallic texture, whereas the watch face required some image manipulation to create a repeating pattern of triangles that came together into a circle. Finally, the table was given a non-repeating wood texture, which closely resembled the original picture. After creating the basic shapes of the objects in their respective colors, I made them modular and arranged them in the scene before adding textures to achieve some level of recognition and coherence.
* To ensure proper rendering, the code for this project is divided into multiple sections. The various shader functions are applied in a specific order, while textures are loaded and unloaded in their own functions, input processing is done separately, and mesh creation and rendering have their own designated functions. Furthermore, within the rendering and mesh creation section, objects are handled in a modular way through the use of buffers, vb's, and vat's, making it easier to add new objects without making unnecessary changes to the code. The virtual camera used in this project is a unique, as it is the scene that revolves around the camera and not the other way around. The camera is treated as an object that can move around scenes similar to a real-world camera. Movement of the camera is achieved through keyboard inputs and mouse or trackpad movements. It kind of reminds me of a video game, since the user is in control of the camera and has shortcuts for specific actions. Pressing “W” moves the camera forward (zoom in), “S” moves the camera backward (zoom out), “A” moves the camera to the left, “D” moves the camera to the right, “Q” moves the camera up, and “E” moves the camera down. With these keys and mouse or trackpad movements, all camera movements, except roll, are implemented. Since these functions are custom made, a lot of it can be reused for other purposes or within other projects such as the UCreateTexture function, which can be implemented in another similar project, granted that the same variables are used. Additionally, files such as the mesh and lighting file can be reused since it serves as a template for how lighting and mesh should work and look.