# CS 330 Comp Graphic & Visualization

**7 - 1 Final Project: 3D Scene.**

From creating the last 3D scene in this course, I was able to have a better idea of what I wanted to build within the OpenGL environment. The scene I was looking to create was simple, yet containing just enough challenge to satisfy the criteria with multiple shapes, textures, and lights within the model. Hence, my initial choice for a 3D scene was a chair, which I found to be simple and looked easy to plot points in a 3D space, but that quickly became very difficult to draw, so I decided draw a table or desk with three objects (Desktop Computer Monitor, Mouse, Keyboard, and a Mug) on top of it. The desk is a rectangle shape with legs supporting it underneath. I also picked the table scene due to the little complexity involved with recreating the scene (triangles to draw). With less triangles to draw, coding the required functions per the provided guideline was easier. It was understood that these shapes would be built using vertices that the program could link together to draw a surface. Each of these vertices needed to not only have x, y, and z positional coordinates, but also coordinates detailing how textures are applied. Having this in mind helped me to keep the functionality of the code organized within their own functions and enabled failing code blocks to be fixed so that the desk scene to be rendered when the program was ran.

I made sure users with different input or rather limited input devices would be able to navigate the (desk) 3D scene and ultimately have the same experience by way of the "W, S, A, D" keys on the keyboard. To able to zoom in and out of the object for example a user without a mouse will use "W" and "S" keys respectively. The "A" and "D" key would be used to move the 3D object to the right and/or left. There is also the ability to pan the object around the center pivot point by holding the "Alt" key on the keyboard and left clicking on the mouse. Using the functions of UKeyboard, UKeyReleased, UMouseMove, and UMousePR allowed me to code which specific key to press in order to perform the zoom in and out and moving left to right tasks. The virtual camera was used such that at a certain point while it is locked it will be able to move with the keys as they are pressed on the keyboard.

The process of drawing the objects is the one area where I made sure some customization was done to ensure each piece/part of the entire scene is clearly explained and modularized for anyone looking at it in the future would be able to easily understand the thought process behind the creation of each piece. Inline comments also could help in that regards. Most of the heavy lifting was done in prior modules which were well commented, so those helped me to put together the code to be able to perform all the tasks that is asked of us to do. I set up all of the function prototypes in the beginning of the code and then added code to each of the functions to make the program work all together. Each handles a certain part of the code, such as creating the window and rendering the object. There are other functions that have their respective role to play in the code and are all organized in the same layout all through. Being able to modularize and allowing for additional input of points in the code makes it quite reusable since you can do so with the supporting code for it to render the object properly.

Kind regards,

Denis Dzenyuy