**CS-330 Comp Graphic and Visualization**

**Ali ARSLAN**

**7-1 Final Project Reflection**

In the scene I chose, there were many objects, and I tried to program them all as much as possible. I specifically created this scene because it contains various materials (liquid, transparent glass, ice, wood, etc.), allowing me to work on all of them within a single scene. While working on this scene, I had the opportunity to practice with many different objects in terms of shape and material, and I gained certain knowledge about each of them. However, I didn’t expect that working with such different materials would be this challenging. Additionally, I wanted to work on relative positioning because objects like glasses, bottles, and laptops are complex.

I tried to discover the textures of nearly all the materials through trial and error. Due to my inexperience in this area, it took a lot of time, but while working on relative positioning, I used various methods such as transformation matrices and scaling according to reference objects to complete them. I think I did quite well in this regard. In fact, I created a few better models, but since these changes required modifications to the libraries provided for us, I decided not to use them. Instead, I tried to create them by modifying our own code. For example, to create a hollow cylinder, I used two cylinders and a large number of planes to fill the gap between them, thus solving the situation without altering the resources we needed to use.

Another change I made is that when the TAB key is pressed, the camera type switches to Orthographic. Since the key listener loop is running, it would interpret the key press as if it were pressed multiple times due to the duration the key is held down. I solved this problem by adding a property to the view manager that controls this property value when the key is pressed, preventing it from being interpreted as pressed multiple times as long as the key is held down. This increased the usability of the code, and now the transition between view types works smoothly when the TAB key is pressed.

The program uses two different input devices: the keyboard and the mouse. The camera can be moved forward, backward, up, and down using the WASD and QE keys on the keyboard.

* W -> Forward
* S -> Backward
* A -> Left
* D -> Right
* Q -> Up
* E -> Down
* TAB, P, O -> View changes

With the mouse, the camera orientation changes, and we can slow down or speed up the camera using the scroll wheel.

All objects are treated as functions, and simple geometric shapes that make up the objects are added relative to a reference point. This way, when the rotation values of an object change, or its dimensions and position change, all components adjust automatically without needing to manipulate each one individually. I believe this greatly contributes to modularity, allowing these objects to be reused without any changes when the scene layout changes or when creating other scenes, simply by altering dimensions, rotations, and position information. All these features made the program that I developed , more organized and significantly increased its reusability.