7-1 Final Project

Joel Garcia

SNHU CS-330

**Introduction**

I want to start by saying that one of the biggest challenges in the overall project was that I was learning bit by bit, adding to the project code bit by bit over time. Because of this the code is a lot messier than I would like, with pieces stacked on top of and in between other pieces of code each week. Were I to start over with the knowledge I have now I think I could do a much better job with much cleaner code, but I don’t really have the time to rewrite everything so Frankenstein code it shall remain. The code is still perfectly legible and follows best practices etc., but I though it best to preface this way because it informs a lot of my design decisions.

**User Controls Documentation and Development Choices**

In this section I will briefly discuss the user controls and implementation.

* Camera – Mouse Movement = look around

This was pretty straightforward as I used the prebuilt camera.h file. The various pitch and yaw movements that go into this are a bit difficult, so I thought it best to just implement what was already provided.

* Movement – WASD = forward/backward/strafe left/strafe right, QE = up/down

While this is pretty easy to code on it’s own, since I was already using camera.h I just used the prebuilt version for this too only modified to add the up/down movement.

* Movement Speed – Mouse Scroll = increase or decrease movement speed

This was easy enough to incorporate into camera.h as it already has a mouse scroll function that changes zoom. I just changed the code to modify movement speed instead of zoom. I set a lower limit of 0.0 to stop movement completely and an upper limit of 15.0, as anything above 15 made it really easy to lose sight of the scene.

* View – P Key = toggle between perspective and orthographic

This was the hardest part for me, partly because I didn’t understand what orthographic is and partly because I couldn’t figure out how to toggle. The regular key input will continue applying the key press as long as it is held which makes the view change repeatedly when P is pressed. I ended up using a key\_callback function that reads only one key press and changes a Boolean value each time. Depending on the current value of said Boolean, either perspective or orthographic will be displayed. While I figured out the toggle function a couple weeks ago, because I had trouble understanding orthographic parameters I didn’t fully implement it until the final project. This is one of those things that if I had understood it from the start I could have implemented it with less lines of code, but because I already had nearly everything else built and couldn’t easily move things around without drastic changes, I ended up simply creating two URender functions; one that renders in perspective and one in orthographic. In the main method that calls the render function I have a pair of “if” statements that calls either perspective or ortho depending on the Boolean I set up.

**3D Scene, Objects, and Development Choices**

A picture containing tableware, dish, food, meal

Description automatically generated

* Scene

My reference image was sourced from a Ukrainian friend of mine and depicts a borscht meal setting. I tried to default to something close to the original point of view and made as few changes as I could, while also taking certain liberties to fill out the scene.

* Objects

While I can create my own meshes (I have in previous assignments and even do in the final project), I used Professor Brian’s meshes code for almost all of my objects as it does a much better job in a much cleaner way than I could, and because it was stated early on that creating custom meshes isn’t necessary and is outside the scope of the course. With that in mind, I started with a plane for the table surface and the dish holding bread and cheese. The change from garnishes shown on the plate to cheese was the first change I made, and I did so to simplify the learning process, partly because the garnish is a complicated mass of objects and partly because even with the prism shape I couldn’t get and garnish textures to display correctly, thus, cheese. Once I got to actually filling out the final project, the front bowl containing borscht came next and all the other objects were pretty easy to add from there.

* Textures

The hardest part with textures was finding images with the proper usage rights. Even when conducting an advanced search for common license images there are so many that end up giving an access violation, and I ended up having to use a lot of trial and error to find ones that work. In a lot of cases the textures used weren’t even my first choice, they were just the best I could find that worked. The most difficult textures to implement were for the sour cream tub and condiment packs, all of which are textures pulled directly from product images. The sour cream or “сметана” tub is a bit blurred out so I found the closest approximation I could, and I had to double the image side by side to get it to wrap properly. The mustard (“гірчиця”) and horseradish (“хрін”) are the exact product images and they have both been rotated 90 degrees and mirrored in order to display correctly.

* Lighting

Similar to other areas of the project, rather than try to create a custom phong lighting model I just ripped it from a previous assignment and adapted it here. The primary source of light in my reference image seems to be a window that is behind the viewer, so I implemented a plane object as a light source that is just behind the default camera view. This is actually the only object that uses my own custom mesh since I figured it would be easier to quickly write my own mesh for a vertical plane than to use the prebuilt mesh for the horizontal plane and rotate it. I also have a second light source in the form of a small sphere (mean to represent a ceiling lamp) above the scene, just out of default view. The original image doesn’t seem to have a secondary light source but the rubric requirements ask for one.

**Conclusion**

The final part of this writing prompt asks for an explanation of any custom functions you used, but I didn’t really use any. Even my “custom” functions like the light plane mesh or ortho view toggle aren’t truly custom, but rather are adapted from other functions either found in other assignments (UCreateMesh) or found through a google search for implementing a toggle (key\_callback). This lack of fully custom functions serves my conclusion: It is not my duty to reinvent the wheel, but rather to adapt and implement the wheel as the job requires. This is supported partly by the course material that has repeatedly stated that certain code is outside the scope of the class, and by my own general understanding of team-based software development that does not necessarily want you to be creating your own custom code no one understands. With this in mind, I did the best I could with the tools already available, and by doing so I believe that I was able to create a richer, more complex scene and overall better finished product.