7-1 Final Project

Brandon Lombard

Southern New Hampshire University

10/17/2021

My 3D scene depicts a toaster in a kitchen setting. The toaster base itself is the plane while the lever involves cubes, and the knob is an attempted cylinder, while the button is a torus. The toaster base is a tall trapezoid in shape because I had issues making the edges rounded. I also made it tall instead of a complete cube because it is more rectangular in the original image. The knob is set to be on the left while the lever is one the right, as seen in my original project proposal. I chose to create a dark bar behind the lever to make it look like a gap and to be more realistic. For this, I just made a thin cube behind the lever. I used the same method for the top of the toaster where it looks like a slit meant for cooking items. The knob and toaster plane are textured with a silver stainless steel look, while the lever is mainly colored grey to make it stand out. While my object is blockier, I placed the elements to the best of my ability to be organized. The lighting in this scene is mainly from the front position with a bright white color, to be similar to the kitchen lighting and make it shine bright as in my selected scene image. The front light is mainly specular, while I chose a second light coming from the left of the toaster being mostly an ambient yellow. This is because I wanted to recreate a yellow light coming from the other room. I chose to make a toaster scene because I thought it would be a bit of a challenge due to the curved edges and silver texture. I proved myself right as I struggled throughout the assignment.

This 3D scene allows the user to scroll, use the WASD keys to move around, Q and E to go up and down, and use the mouse to look around. I used multiple camera movement elements to allow the user more control over the scene. WASD allows forward, back, and side to side movement and was written in the camera header file. In order to create the up and down movement, I defined the “UP” and “DOWN” commands in Camera\_Movement and specified “Up” in the camera attributes. When a user presses Q on their keyboard, the camera moves up, and when a user presses E, they go down. I was able to do this by setting the position plus and equals to Up times the velocity for Q, and the position minus and equals to up when E is pressed. On the mouse scroll, I made it zoom based on the mouse scrolling direction. Forward scrolling means the user moves forward while backward scrolling makes the camera go backwards, similar to using W and S keys. Finally, by processing mouse movement depending on the x (yaw) and y (pitch) offsets and mouse sensitivity, I was able to allow the user to simply move the mouse in any given direction and the camera follows, therefore giving further control to the user. The only method I struggled with and could not complete were changing perspective views to ortho views by pressing “P.” I did attempt it and the code is still in the project starting at line 465 in Final Project.cpp.

Some of the custom functions I used in this program includes texture changing functions. By pressing one through three, the user can change the style of the texture from regular to more cartoon-like. Additionally, the user can use the brackets buttons “{“ and “}” to scale the texture image to fit better such as by sizing down the texture by pressing the left bracket, or enlarging it with the right bracket. This allows the user to adjust the texture if needed, such as if not rendered properly, until the user finds the best look for the 3D model. Additionally, I added some keys to the camera movement, such as the plus and minus symbols on the user’s keyboard also move the camera forward and backward. I thought that this would increase the user’s ability to move around the program due to the increased level of camera methods. Overall, I added a few elements to allow the user another level of customizability.