**Justify development choices for your 3D scene.**

A computer monitor and keyboard on a desk

Description automatically generated with medium confidence

I used the above 2D reference photograph to build my 3D scene. I chose to build the computer monitor first. This was an important component of my scene because it is the center piece of the image. It is also the main component of my desk environment. This object was built up of two different shapes. The major potion of the monitor is a rectangular cube and the base is built of a pyramid. In the design process I added the flat portion of the pyramid in the front because it gave a better visual flow. When coding this portion, I used two different build objects that were built into separate functions. I used a function to create the monitor itself, then a function to create the screen that would have a desktop visual.

Next, I created the speakers. Initially, I considered making two large black blocks but decided I would make them a little more complex. I split the creation of the speakers into three functions. I created the base of the speaker with a modified cube code and then created the sound producing portion of the speaker with a plane. This allowed me to use two separate textures to create a more realistic object. I also added a dial to the right speaker to be uniform with the image.

I also created a keyboard. This was another design element that was important to me. It was difficult to identify the computer as such without this element. I felt it looked like it could be a television, so I decided to add the keyboard at the end. This is made of a modified cube code, which I set in its own function, as well as a plane that allowed me to texture the keys atop the keyboard.

I added two objects that would serve a continuity purpose as opposed to filling requirements. I added a lamp when I learned we would be adding specialized lighting to the scene. This is made up of three cylinders that are separately textured. I also added a desk with a wood grain texture to my scene. I originally planned to create it with a plane shape but decided to give it more girth by creating a modified cube. This made it feel like a component of the screen and not just an afterthought.

**Explain how a user can navigate your 3D scene.**

The implementation of a camera allows my user to have a first person view of the 3D environment. With WASD, they are able to move forward, left, back, and right, respectively. I then added movement up and down using the Q and E keys. With the mouse, they are able to change their viewing reference. This allows for a swiveling of the head. With the scroll of the mouse, a user can better control the movement speed.

This is all if the user maneuvers the scene with a keyboard and mouse. If we were to use a controller, it would be an interesting transition. I would map the WASD to the left thumb stick and directional pad on the controller. This will allow for the user to use it to move around the screen. I would then map the up and down function to the A and B buttons, respectively. The left thumb stick would be used to control the frame of view for the user. I would prefer to use this method to maneuver the scene as it would allow a more immersive experience for the user in my opinion.

**Explain the custom functions in your program that you are using to make your code more modular and organized.**

I prefer to write all of my functions into the source code. This is a personal preference for me as I have never been fond of writing full files for functions. I also used simple modified cubes, so it was simpler to keep it all in the same file. The only function I used that was written to a separate header and code file was the cylinder.

The custom functions I created were my:

unsigned int loadTexture(const char\* path);

void CreateDesk();

void CreateMonitor();

void CreateScreen();

void CreateSpeakerBase();

void CreateSpeaker();

void CreateKeyboard();

void CreateKeys();

I chose to write these functions to not only simplify and clean up my code, but to better identify the creation of each object in the code, and its corresponding texture. I used the functions to create multiples of duplicate objects. This kept from repetitive coding and allowed me to have nearly identical objects in different places in my scene. The loadTexture function was incredibly helpful in cleaning up the implementation of 9 separate textures. The use of this function was the most pivotal in my code. It decluttered the texturing process.