CS 330 8-1 Project

Reflection

Matthew Wilt

Justify Development choices for my scene

I chose a scene in week 2 with a lot of black. Knowing what I know now I would have made a scene that had more color. The objects I picked for the scene were relatively primitive objects. I used a plane for the desktop layer, which was flat and seemed easy. I had to change my texture because of time to go back and reshoot my chosen scene with no objects to get the perfect texture wasn’t feasible. I also used a plane for the Wacom tablet. I knew a cube would be used for the hard drive. I used cylinders for the tape, chap stick, and Wacom pen. I used the ShapeGenerator functions for the planes and sphere. The objects I chose for the scene I wanted to keep as simple as possible.

Explain how a user can navigate your scene

Users can use the keyboard and the mouse to navigate my scene. The keyboard and mouse settings were setup and configured within the process input function. I was able to incorporate WASDQE for basic camera movements. The mouse was setup in the callback functions whenever a mouse enters the screen. The user can just move mouse or scroll to navigate the scene. By pressing the P key, the user can switch to an ortho view of the scene. This was a tricky one to figure out, but I got it to work.

Explain custom functions

In was able to call a cylinder because of the cylinder links that were provided in the course. This is also true for the plane and sphere. As I setup the arrays and buffers for these I was able to incorporate the includes to just call the objects to draw. When setting up the vertices, I learned that I could just pull that data into whatever I was drawing, that wasn’t linked or included. I had to restart this project several times during this term. Spending time researching and going through the modules numerous times, I finally picked up on how the code could and should be organized. In my program, I start with all the necessary includes, then I declare all the variables. Then use the provided window and context window code. Once the vertices are created below that, I started creating the arrays and buffers one by one. Below that is the texture creation and linking. Then into the while look is where I called the functions needed, as well as binded the arrays and settings for each object. Finally drawing the objects. Below that is all the functions created and needed.

Through this course I tried many different ways to get this to work. As you will see in my program, I left in all the commented out code that I didn’t end up using.

Read me questions

You will also reflect on the work that you have done in this project. Reflecting is a valuable skill to cement your learning. It will also help add context to refresh your memory when you use your portfolio in the future. **Update the README file** in your repository and include your answers to each of the questions below, using the sub-bullets to help guide your thoughts. You could include the questions and write a few sentences in response to each one, or you could write a paragraph or two weaving together all of your answers.

1. How do I approach designing software?
   * What new design skills has your work on the project helped you to craft?
   * What design process did you follow for your project work?
   * How could tactics from your design approach be applied in future work?
2. How do I approach developing programs?
   * What new development strategies did you use while working on your 3D scene?
   * How did iteration factor into your development?
   * How has your approach to developing code evolved throughout the milestones, which led you to the project’s completion?
3. How can computer science help me in reaching my goals?
   * How do computational graphics and visualizations give you new knowledge and skills that can be applied in your future educational pathway?
   * How do computational graphics and visualizations give you new knowledge and skills that can be applied in your future professional pathway?