Module 7: Final Project

Scot C. Batton

CS-330 Comp Graphic and Visualization

Brian Battersby

February 25, 2024

The project that I have been working on with Triangle & Cube Studios has been to recreate a 2D model of a pineapple, apple, and a box of pancake mix sitting on a countertop in a 3D space. This scene was created while cooking breakfast one morning and brought life through shapes and textures. This 3D rendering of the 2D image will be used in a 3D printing later in the business. The scene was created using multiple pre-made meshes that recreate objects in 3D space. The countertop was the easiest of decisions because the plane is the best mesh to create a simple plane. The pancake mix was created using a box mesh and texture was added to recreate the packaging of the original box. The apple was recreated within the rendering using an elongated sphere and rotated to lay on its side. The pineapple was a more complex object to create. The pineapple is made up of an elongated sphere and multiple prisms that help to create the stalk on top. All the objects within the rendering have their own specific texture to ensure realism within the scene. The objects are then lighted individually with two separate light sources and a light source that shines throughout the entire scene. The scene was created using a pre-made meshes.h and meshes.cpp file.

Each of the objects within the scene is created in 3D and has a three hundred sixty-degree view that needs to be recreated and explored. The scene has been inserted into a computer system and was designed to be navigated with a computer mouse and keyboard. The keyboard functions allow for the movement up, down, left, right, forward, and backward within the scene using W, A, S, D keys. There are also O and P key functions that allow for the switching between orthographic and perspective project matrices. The keyboard functions allow for a 3D look of the scene but can be combined with mouse controls that will make an even more immersive experience. The mouse for the computer controls the panning of the camera as if it were the head movement of someone within the scene. Moving the mouse in any direction will allow the user to experience a three hundred-sixty-degree view of the scene. This movement can be combined with the keyboard functions to show all aspects of the objects being shown. The scroll function of the mouse has been added to control the speed at which the camera moves within the scene. When the camera button is rolled forward the camera will speed up and the camera will slow down if rolled backwards. The process input function in the code allows for the inputs of the keyboard to be included and functional. The camera.h file allows for these functions to be implemented. The mouse input, scroll, and cursor are all created through different functions to hide the cursor, control mouse movement, and scroll movement. All these inputs allow the user the best look at the scene as they can get.

There are many functions that are within the code that make the code more modular and organized. The vertex and fragment shaders are functions that allow for the transformation of 3D coordinates of vertices from object to screen space. The vertex shader allows for transformation, scaling, rotation, and translation. The fragment shader can handle textures and lighting. The shader.h and shader.cpp files are used to implement the shaders within the code. The function that is used reusable and allows for the code to be modular and used for multiple textures within the code, I have used five in my code. The creation, scale, rotation, and translation of the meshes is reusable along with the light sources for each individual object. These portions of the code are able to be repeated, and are in my code, and allow for a quicker completion of the code and the reusability to allow the same meshes to be used over and over again.

The project shows a lifelike rendering of the 2D scene that we captured prior to starting this project and will allow for 3D models to be created after. The code is created with modularity and usability in mind and provides a source code that can be used in the future for different objects or to obtain only one objective.

*Resources*

Chapman, S. (2020). *Snhu-CS/CS-330*. GitHub. https://github.com/SNHU-CS/CS-330

*OpenGL ES  :  Android developers*. Android Developers. (n.d.). https://developer.android.com/develop/ui/views/graphics/opengl/about-opengl

Vries, J. de. (2014, June). *Introduction*. LearnOpenGL. https://learnopengl.com