**Final Project**

Michael Kinful

Southern New Hampshire University

CS-330-11427-M01

Kurt Deisch

June 22nd, 2024

**Final Project**

To complete the task, the use of OpenGL and GLSL in the project would be the first task. There will be four primary objects in the main scene: a speaker, purple glitter ball, perfume bottle, and stapler.

“maincode.cpp” is responsible for creating a glfw window and also the main application algorithms that deal with the opening event, the looping process and virtually the closing routines of the application. It is duplicated with the OpenGL context, which is setup with screator, window, render, poll, and releases resources which when used are not supposed to leak memory. As a result of this use of separation and in each case is a different one, by giving the scene manager its own class and the rendering inclusion into another class, the maintainability and the independence of the algorithm codes are made.

“scenemanager.cpp” deals with dealing with modeling and object creation, which means it creates parts of the scene like speaker, glitterball, bottle, and stapler. It is the shader initialization which does all the work from the compilation, then the fragments and vertex draw, and finally the linking. Textures are to be applied to the objects for better simulation and some unallocated resources are defined and returned which is carried by the library file.

“viewmanager.cpp” manages user input and graphics. It initializes the GLFW window and context, and it sets the callback functions for the window resizing, mouse motions, and the camera zoom, never forgets that any of the formed shapes is a script and inserts functionality into the application that manipulates an objects' positions or their grasping method by a user.

Objects' proportioning accuracy was a priority in image interpretation based on the sample given. The speaker is a tall, thin cylinder, the glitter ball is a small sphere, the perfume bottle is a shorter, wider cylinder, and the stapler is a rectangular box. The objects are correctly placed to depict real-time physical objects.

The plan employs shaders to manipulate light, map objects, and make normal transformations so that the scene can accommodate different lights and textures at the same time. Specific shaders amplify the visibility of light-colored things. The lighting arrangement incorporates such things as a spotlight and a key light, thus imitating the real world by creating the ambience, diffuseness, and the specular parts necessary for depth and realism. A dynamic spotlight flies over the scene, which gives a demonstration of light's behavior with the objects it interacts with.

The design decisions center on clearness, maintainability, and visual physics. The main code, which should be modular, such as maincode.cpp, scenemanager.cpp, and viewmanager.cpp, renders fault localization to each part, thus allowing easy fixing independently. The assigned functions help to make the base code stronger and more adaptive for the next generation. They allow for such things as bigger items and interactions which bring about not only better employment of the content but also make it more reader-friendly and easier to follow.