Zong Lee

CSE 165

Team 39

December 12, 2021

Project Description

In this project, understudies were entrusted to implement an OpenGL game utilizing the SOIL library. We were allowed to do whatever we were keen on but one object on the scene is to be controlled by the user as the other moves independently, for example, Super Mario and Space Invaders. OpenGL is the name for the specification that portrays the conduct of a rasterization-based rendering system. It characterizes the API through which a client application can handle this framework. The OpenGL rendering system is painstakingly determined to make hardware executions admissible. The OpenGL API only deals with delivering illustrations. OpenGL doesn't give functions to animations, timing, file IO, etc. OpenGL is uniquely concerned about rendering.

I personally wanted to make a platformer/side scroller game where the player needs to avoid move moving objects or enemy ships that move autonomously at a rate. Like an intersection or vehicle avoiding game. The player should move in a proper position in different levels in these games yet I needed my player to move uninhibitedly around the screen. I likewise desired my game to have a space theme that makes it seem as though a Galaga game without the shooting. My inspiration for this was expecting the game wouldn't be excessively huge for a person to do, it seemed, by all accounts, to be captivating and provoking enough to show my insight into the course material. It permitted me to convey myself more than drawing a picture.

My technique for finishing this task was to comprehend the prerequisites that should be carried out into the game to fulfill the project proposal. Develop a plan to meet the prerequisites and design specifics such as a layout or other materials. Start hammering out finer details on design documents. Construct code to and start to build areas in the game. Architecture for the game is created, tools are being worked on, and aesthetics begin to shape into art designs. Test written code checking for bugs and mistakes. As production moves along, the game is tested and difficulty is balanced. Begin to polish all assets in the game and add audio where it's needed. Deploy the game while maintaining and tweaking areas in need. My main objectives were to get the collision detection, scoring framework, textures utilizing SOIL, and autonomous movements done and working.

There were many difficulties like setting up the GLUT libraries. I couldn't run my game without mistakes with the libraries. This took me many endeavors to investigate and burned through a great deal of time. Learning and becoming acquainted with OpenGL, GLUT, SOIL libraries likewise took a ton of time as it wasn't something instructed in lectures. My first few mishaps were issues with creating enemies as I was experiencing difficulty getting enemies to produce randomly in the y-axis. I was having segmentation faults while exiting, however this was fixed as it was simply deleting non-existing objects. Eventually, I got everything to run with no difficulty.

To conclude this project, I effectively carried out an OpenGL game utilizing the SOIL library. I chose something I was interested in which was to assemble a side scroller game. One object on the scene was successfully implemented to be controlled by the user and many objects were made to move independently. OpenGL was utilized to depict the behavior of a rasterization-based rendering system of a game. It defined the API through which the client application controlled the framework. In the end, the OpenGL API managed its occupation of delivering designs.