When brainstorming the idea for our project, we decided to develop something that was heavily reusable, and utilized all concepts learned in class. We came upon the topic of writing an engine that could quickly and easily create 3D scenes. We decided upon a design that we realized could recreate every assignment we had done in class. But we needed something new to do for this project.

The purpose of our project was to master OpenGL concepts and develop an environment to quickly and easily display 3D scenes. In order to test this environment, we designed a simple space shooter game. The object of this game was to control a ship that could shoot projectiles at oncoming ships. When a projectile collided with a non-player ship, the non-player ship would disappear from screen. We also decided to add a ‘final boss’ character. We decided the most entertaining character we could use was a mechanized Dr. Yerion, and use her trademark ‘BeBop!’ as well.

The game features a player ship, controllable by the keyboard. It can shoot a projectile by holding down the space bar. The enemies flying toward the player provide an exciting environment full of action and thrill. The game is controlled by an OpenGL based rendering engine. The engine utilizes a node based hierarchy where each node’s position, rotation, and scale is controlled by its parent, and the top node is controlled by the programmer. The engine features a simple collision detection system and object pooling. There are multiple ‘factories’ that can create different types of objects to display on screen. The two available factories can produce textured objects or solid color objects. There is support for lighting as well, as each solid color object can have its light material properties changed at any time.

In order to move the objects around, the model view matrix would have to be updated. With the number of objects on the screen, care would have to be taken so that each object is displayed where it was meant to be displayed. Several OpenGL functions were utilized to make the game run independent of frame rate, which required knowledge of how to normalize translations, rotations, and scalings. In order to support both of the factories, we needed to understand how OpenGL utilizes shader programs, and be able to switch between them depending on the current object being drawn. Our code is also completely modularized with a huge emphasis on reusability. Thus, our project must be comprehensive.

For these reasons, our project deserves an A. Both of us feel strongly that we can develop many different projects using OpenGL and this engine because of all the effort that we put into it.