Title:

Robust Engine

Team Members:

Lucio Franco

Benjamin Snoha

Description of application:

Robust Engine is a game engine that is designed in order to allow a variety of games to be created with it. The engine itself has a state management system to keep track of different scenes. A scene can be a level in a game or a screen that displays information. The scenes keep track of generic game objects and each game object has its own logic implementation. This means that in order to make a game, the end user only needs to create new game objects with custom logic and add them to a particular scene.

In order to demonstrate the capability of Robust Engine we created a small game. The game is from a top-down view and implements a tile system. The player can walk around using the WASD keys and is automatically rotated towards the direction of the cursor. The player can click the left mouse button to shoot bullets. The goal of the game to reach the "Finish" block located near the end of the level without being killed in the process. Throughout the level are turrets that aim at the player and shoot. The turrets can be destroyed by bullets. Once the user navigates to the last level and reaches the last "Finish" block, then the game is over.

Although our game is a top-down shooter, Robust engine can easily be used to develop different applications. For example, a platform game could be implemented in the same fashion as our example game. The only thing that would need to change is the logic of the game objects that are added.

Goals:

We originally intended to write an entirely different application than Robust Engine. However, driven by our passion for game development and data structures, we decided to create a game engine instead. When we set out on this journey our only goal was to develop a working game engine. We primarily wanted something that could produce a finished game. In the end, we accomplished our goal.

Goals not achieved:

Although it sounds arrogant, we achieved all of our goals with ease. Whenever we encountered a problem, we simply put our heads together and worked it out. Our use of group programming and debugging was essential throughout our coding process.

Who did What:

Lucio: Programmed the entire structure of the game engine. He implemented the state management system with scenes, he created the abstract game object, and created the initial game loop.

Ben: Programmed the game that was based on the game engine. He implemented all the game logic the game objects and created the artwork.

Both of us: In addition, we programmed together most of the time and helped each other out a lot throughout the whole process.