Bison Invaders - User Manual

Joe Elvin, Yuxuan Huang, Jon Li

Background

The project which we were assigned to work on is the most open-ended problem we have been assigned to date; that is, we were instructed to develop a piece of software while adhering to good principles of object-oriented design and also following a team-based agile development methodology. The project we decided on was a videogame; the way we saw it, this was the easiest way to incorporate all the goals we were given to accomplish. Our game is a twist on a classic wave-based invaders game, where you are able to control a single Bison-themed character and attack hordes of different types of enemies. If the enemies make it to the castle, it gets slightly damaged; if they attack your character, the game ends. There are two game modes available: story mode and endless mode. In story mode, you play until you have beaten 10 waves of enemies; if your castle is still standing at the end of the waves, you win. In continuous mode, you cannot win; you can only do your best, accumulating as many points as possible until the increasingly-frequent enemies overrun your castle.

Motivation

There were several goals we wished to accomplish when we set out to make this game. The first was the overall “point” of the project; we wanted to develop a small-scale software while adhering to the basic principles of professional software development. This includes developing the software as a team as prescribed by the scrum methodology; one of us was prescribed to be the scrum master, and another was prescribed to be the project owner. Part of following the scrum methodology includes developing the software using an agile method. Making this game is the perfect way to apply these principles, as it is reminiscent of a software which a professional development firm may make.

Another goal of ours was to tackle the project from an object-oriented approach. This means organizing our class files in such a way that interdependence between modules is low, while keeping our files highly cohesive. For a video game like ours, it is especially important that we leave it modular and easily editable; there are many improvements which could be made, and we wish to make it as easy as possible for potential future developers to present their own unique twist to the simple software which we have developed. We also wished to minimize the tasks and roles for each method and class, as this helps with keeping the game simple, readable, and easily upgradable.

Aside from the given goals of the project as a whole, developing a video game requires additional goals, the first of which being game engine development and following a model-view-control methodology, given that a successful video game relies almost entirely on user interaction. A big part of this is handling user input based on key presses and monitoring several types of input at a single time, and threading so that several processes can occur at a single time.

Another important aspect of video game development is making a comprehensive GUI with custom graphics, so another goal of ours was to learn how to make our own graphics and upload them to our GUI so that we can tie everything we've made together into a single user-friendly game.

Instructions

Upon running the game, the user is presented with two options for which game mode to play. The first is story mode, in which the user fights 10 increasingly difficult waves of enemies in order to defend their castle. After defeating the 10 waves, a final boss is fought, and upon defeating him, the game is complete. If the user's castle runs out of health at any point along the way, the game is over. Endless mode works in a similar way; there are no

waves, but the hoard of enemies increases in difficulty over time, and the goal is to accumulate as many points as possible until your inevitable loss.

The controls are the same for both game types. Using the arrow keys, the user is able to move the Bison-themed main character around the map. Pressing <Space> allows the user to shoot their currently equipped weapon, given that they have enough ammo in their inventory. Ammo is dropped at random by enemies. The player starts out with an infinite supply of tomatoes, the default weapon (and also the weakest), which allows the user to always have a means of fighting. Pressing the escape key pauses the game, and hitting <Enter> resumes the game at the same point at which you paused.