HTML5 Tower Defense Game:

Final Report

Oregon State University
CS467 Capstone Project
Fall Term 2021

Team Members:
Abraham Almahmoud
Casey Cheek
Nam Nguyen
Cliff Webb

Introduction

Over the course of the term, the team developed an HTML5 Tower Defense game utilizing the Phaser 3 game framework along with other tools. We were able to not only meet project requirements, but develop a game that compelled us to understand an unfamiliar framework and embrace different coding conventions and styles. Working in a team effectively simulated a software engineering environment, where each member learned, collaborated, and ultimately benefited one another. The following will discuss a brief overview of project parameters, deviations and similarities from the initial project plan, as well as development tools/ libraries and systems used.

User Perspective

The game allows the end user to choose the level of challenge they want at the very start. This is separated by three different map selections that are set to either easy, medium, or hard. Basic game play revolves around placing stationary, auto-firing turrets on the map to thwart the advancement of enemy waves. Placing turrets consumes a type of currency, while defeating foes increases the said currency. The player also controls their own turret which is the central nexus (the base or “tower”). All enemies travel towards this tower in order to damage the player. There are different kinds of auto-firing turrets and base turret ammunition that complement how the player wants to interact with the game, creating a unique experience for all users. All in all, the basic premise of a tower defense game stays pretty conventional, without too many straying features. The appearance is classical, using a nice semi-retro sprite base aesthetic that compliments the games of that time.

Compare/Contrast to Plan

The development of our project followed the project plan to a large degree; however, as expected, there were a few deviations from our initial draft. By the midpoint (week 6), the group was able to program a functional, albeit, basic game level with our game elements. The plan of implementing tower, player turret, map, and enemy components, provided the foundation for level expansion and additional game features. In retrospect, the contributions were aligned with the expectations of our project plan

and group meetings. The subsequent weeks were also similar; during which, we slowly piecemealed the other game features such as transitions, animations, projectiles, sounds, and game logic. Post midpoint, we did end up implementing some of these additional features into our game - as planned. The few deviations that occurred happened in the last few weeks (weeks 9 and 10) of development.

Notably, we underestimated the time allotted for testing game balance, refactoring, and addressing smaller, non-game breaking bugs. While we did test and fix the larger bugs as we developed, the group had an entire week dedicated to bug, code optimization, and game balance issues - which was limited due to time constraints and further development. Upon further reflection, it would have been prudent to set constrained deadlines in closing the development aspect of the project and dedicate the rest of the time towards testing and refactoring. Particularly, we failed - again, time related, to “recycle” assets (enemy/enemy wave) and construct a better performant game process. Another deviation was developing additional (optional) features into gameplay. We initially planned to have enemy bosses, and game play objects such as traps. However, in fairness, a lot of these components found other forms, such as a slow aoe bullet and tougher enemy and enemy distributions. In summary, while there were certainly deviations from our plan, the plan served as a sufficient and accurate template in developing our project.

Development Tools, Libraries and Other

Libraries:

Phaser 3, Tiled, Node.js, Express

Languages

Javascript, HTML, CSS

Development Tools

GitHub, Visual Studio

Servers

OSU engineering server

Other

Most art assets were selected from https://opengameart.org/, while some were created by the team using https://www.piskelapp.com/, a free online sprite editor. Text files that attribute each art asset to their respective artist are included in the source files.

Conclusion

Over the course of this term, our team has worked together to create a HTML5 tower defense game that is fun, easy to play, and challenging to win. We are confident that the experience we've gained here has helped prepare us for more complex projects, including those in a professional context. There were a lot of challenges that presented themselves to us when we started to initially push our code together in GitHub. We had to quickly learn from each other's code, and adapt to accomplish the goals we set. This teamwork is the core fundamental needed to propel ourselves forward in a software career. It is unrealistic to expect that we will ever be in a situation where we would be the sole programmer in a project. Therefore, it is a necessary skill to be able to not only read and quickly understand the scripts of others, but to contribute in a unified manner. In hindsight, this goes to show how important preplanning is when it comes to a project of scale.