Final Project Report

Dev Team Xi
Web1: RTS

James Pool, Brandon Gipson, Thomas Dale

http://web.engr.oregonstate.edu/~gipsonb/Xi/AG-start.html

Introduction:

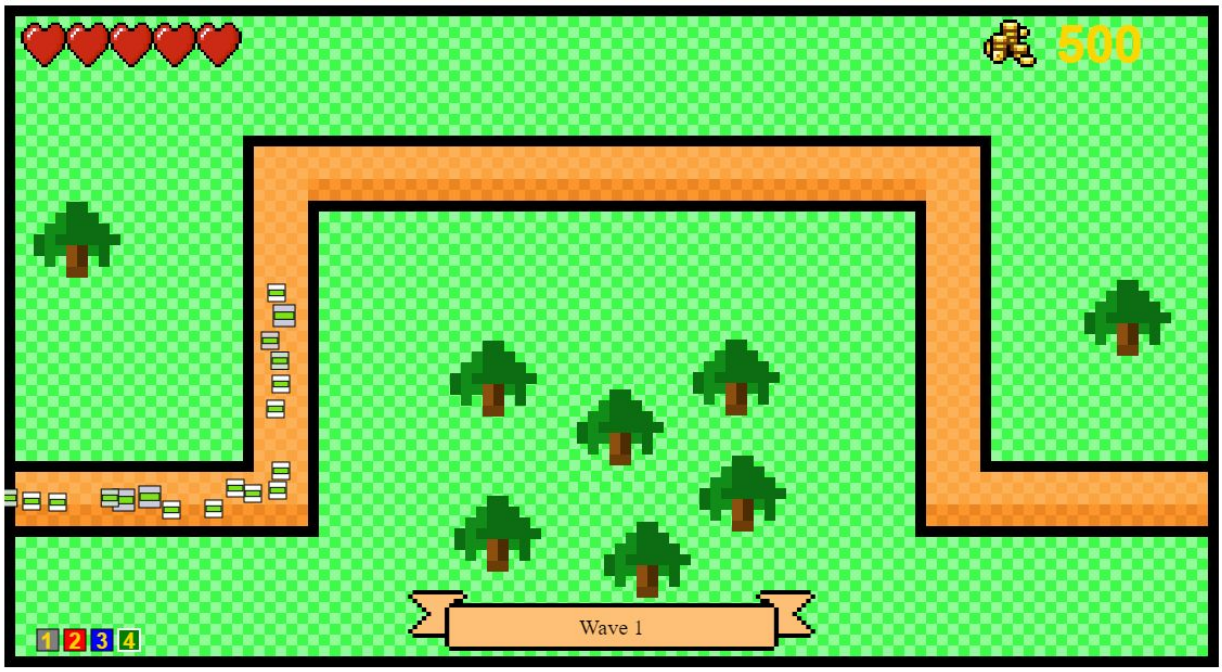
This is the final project report for Development Team Xi. Included in this report is an overall description of the project from a user's perspective, a guide on how to play the game, details of the software and systems used, and a record of what each project member accomplished.

Description:

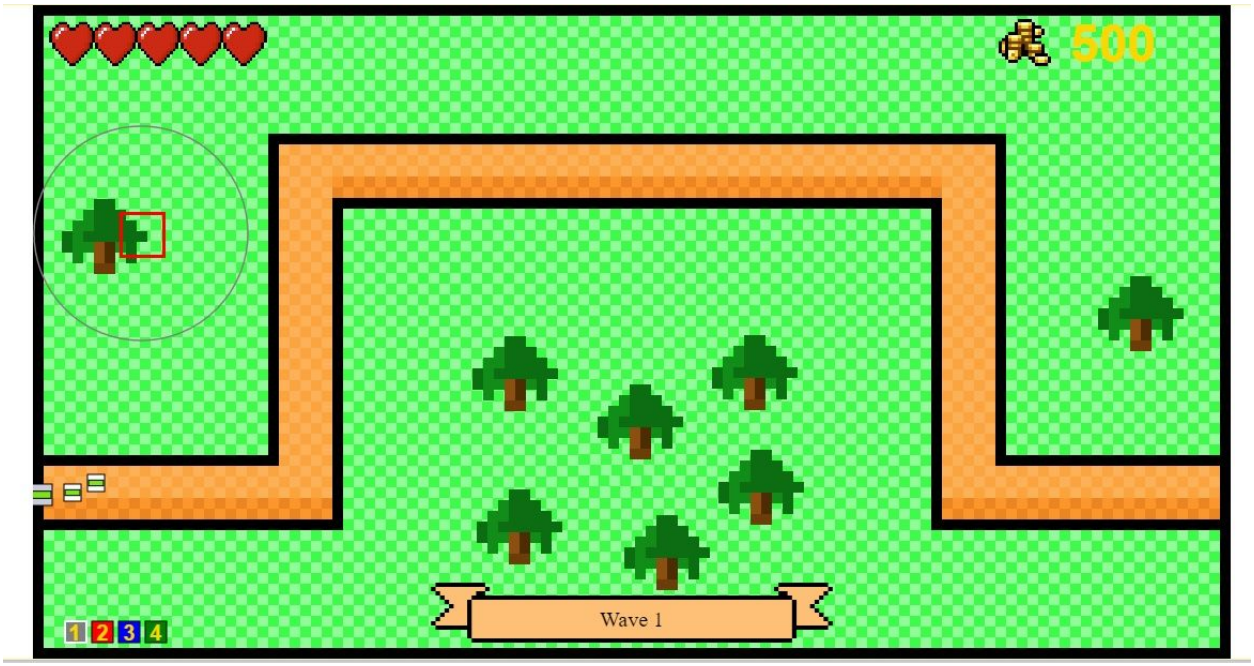
Xi Tower Defense is your classic tower defense game. Users place towers to combat waves of enemy unit spawns over the course of the game with the goal of not letting them reach the end of the path. If any unit reaches the path end the user will lose a heart. Once the user is out of hearts its game over. To aid the user in the defense of the path we have provided powerful gem upgrades. These gems augment the towers offensive abilities. Red increases the tower's damage, blue allows the tower to hit an additional target, and green increases the tower's range. Each tower can hold up to three gems but beware, each additional gem in a tower increases the cost of the next gem. If you run out of gold just kill more enemy units as each unit killed increases your gold. Good luck and have fun.

How to Play:

After clicking start the timer will begin to count down to the first wave. To place a tower all you have to do is press 1 on your keyboard or click the '1' button on the map. After that a reticle will show you where you are allowed to place the tower. Click to drop the tower in the reticled area. To place a gem you can click on the matching colored button on the map and then on the tower you want to place it in. You can also press '2' for red, '3' for blue, or '4' for green to enable the ability to place a gem. After that just click on the tower you want it to be in. Once the game is over, hopefully after you win, just click the "Back to Start" button to go back to the start menu or refresh the page to play again.



This is what the wave system looks like



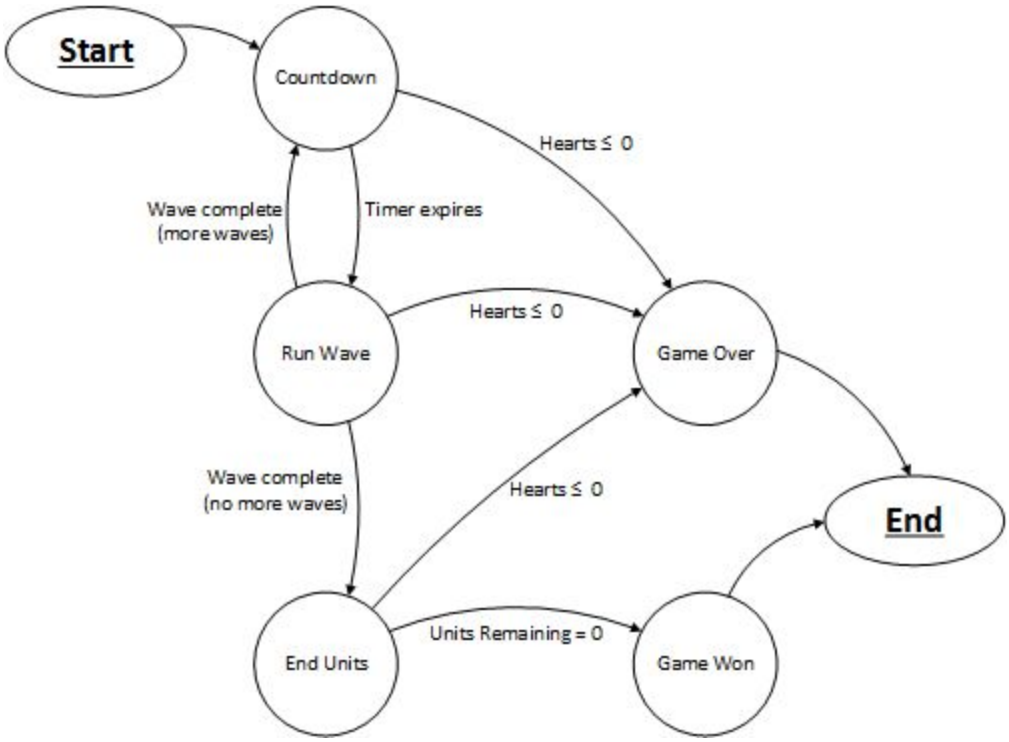
This is an example of what tower placement is like (The square is red for illegal placement)

Software and Systems:

To make collaboration easier we used an online IDE called Cloud9 to create this project. Essentially it's a virtual IDE ran on a server similar to AWS or Microsoft Azure. To host the finished project we just used Oregon State's student web hosting. Other than that it's not too complicated under the hood. We used HTML5's canvas and CSS for all of our graphics and

JavaScript for our logic and basic web page manipulation. We did separate each game function into a different JavaScript file to keep everything organized. They are each named for what they focus on.

Below is a high level state-transition diagram that demonstrates the software logic flow:



Sharing the Load:

Mostly we all worked together on many different aspects of this project expanding, editing, and helping each other out in all aspects of the code. We did try to specialize some into the three aspects recommended in the original project proposal: graphics, AI/logic, and unit balance/creation. Here's a general breakdown of what each team member accomplished:

Brandon Gipson -

- Mainly focused on the creation of game graphics (background, coins hearts, etc) and the logic that surrounded it (ability to increase/decrease hearts and coins)
- Created tower boundaries
- Created tower and gem placing ability (animations, keybind, etc) including graphics for costs and what happens when you don't have enough coins
- Created tower menu system and logic surrounding it
- Added sounds and music into the game (all of which are in the creative commons)

- Created start menu webpage and the navigation tools between it and the main page

James Pool -

- Game Logic & Rendering Loop
- Game State design
- Unit coloring algorithm
- Unit Pathing algorithm
- Game unit & Tower class structure and base functionality (Expanded by others)
- Wave indicator banner & countdown timer

Tom Dale -

- General design of tower and unit interactions including gems
- Tower targeting and attack functionality
- Interactions with gems for towers and units
- Unit wave implementation, as well as what spawns in each wave
- Balancing the towers damage, modifiers, and costs
- Unit Health, modifiers, and values

Conclusion:

Thus ends the final project report for Development Team Xi. We are all really proud of what we have accomplished with this project and have so many other ideas we wanted to implement. The experience we've gained working as a team on a project of this size will be absolutely invaluable in any job we seek after graduation.