

Tower Dream

Project Backlog

Team 15

Yixiong He, Minyi Lu, Kevin Jiang, Weifeng Huang, Zhe Chen

Problem Statement:

The goal of this project is to create a tower defense game in which players are able to not only build defenses to ward off enemies, but also be on the offensive by spending resources to deploy their own troops. Often when we have played tower defense type games in the past, there was only one objective: to build enough to defeat all waves of enemies. However, we plan to twist the game's fundamentals to achieve a more modern style of gameplay.

Background Information

Audience

Our software is targeted towards gamers and tower defense enthusiasts. Our players will be able to experience a new form of tower defense in which they have to balance defense as well as attack to succeed.

Similar Platforms

There are a variety of similar platforms online. First, the classic tower defense games such as bloons tower defense, which focus only on defending and building. Then, there are games such as the Starcraft II Squadron Tower Defense mod, which implements buying units in addition to building defenses. SC2 Squadron TD implements sending units wonderfully, but the game is very unbalanced, and a lot of the time there is no variety, as one strategy works almost all the time. Furthermore, SC2 is now a very old game, and most people are not willing to purchase the full SC2 game just to play the tower defense minigames.

History

All members of our team have played tower defense games before. However, this game mode becomes stale after a while. One set strategy proves itself to work all the time, and

there is no more variety left in the game. By implementing the function to have attacking units, strategies evolve and replay value increases. Players find themselves trying to balance their offense and defense rather than following a set routine every single game.

Environment

Unity - We will use Unity as our developing platform, it is a powerful game engine that will allow us to create games in multiple platforms. Also, the existing assets and active community will be helpful for development and testing.

Unity Collaborate & GitHub for source control - Unity Collaborate is a built-in tool that enables team to work together using cloud build and source control just like GitHub. Furthermore, GitHub also has a unity extension that brings GitHub Workflow to Unity projects. We will be using both tools to develop the project as a team.

Backend and database - As we try to build a game, we need a steady and fast internet connection to load the data from the backend. That's why we chose GameSparks as the backend to store our data and to interact with the local machines.

Requirements

Functional

Bold - Finished

Underlined - In Progress

Italic - Changed

Backlog ID	Functional Requirements	Hours	Status
1	As a user, I would like to create and log in with an account.	8	In Progress: Moved to Sprint 2
2	As a developer, I would like to implement defenses.	10	In Progress:

			Moved to Sprint 2
3	As a developer, I would like to implement offensive units.	7	Sprint 2
4	As a user, I would like to traverse the main menu to choose what I want to do.	10	Completed in Sprint 1
5	As a user, I would like to upgrade defenses.	8	In Progress: Sprint 2
6	<i>As a user, I would like to upgrade offensive units.</i>	7	Sprint 2
7	<i>As a user, I would like to have progression in terms of leveling and experience.</i>	6	Sprint 2
8	As a user, I would like to defend against a variety of enemies.	6	In Progress: Sprint 2
9	As a user, I would like to be able to log out of my account.	6	Incomplete: Moved to Sprint 2
10	As a user, I would like to start with an appropriate amount of currency and to be able to obtain currency by defeating enemies/waves.	10	Completed in Sprint 1
11	As a user, I would like to sell defenses during the game.	8	Completed in Sprint 1
12	As a user, I would like to have my progress saved after I close the game.	6	Sprint 2
13	As a developer, I would like to implement different layouts and levels.	4	In Progress: Sprint 2
14	As a developer, I would like my AI to choose optimal targets in their actions.	9	In Progress: Moved to Sprint 2
15	As a user, I would like to know the time until the next wave spawns.	6	Completed in Sprint 1

16	As a user, I would like to know how much longer I have to defend until I beat the wave.	4	Completed in Sprint 1
17	As a user, I would like to freely move the camera.	10	Completed in Sprint 1
18	As a user, I would like to have a shop where I can purchase towers, units, and upgrades.	7	In Progress: Sprint 2
19	As a developer, I would like to inspect our project, unit inspect our code, and inspect the design of our code.	100	In Progress: Sprint 1 and 2
20	As a user, I would like to see how much currency and resource I have.	6	Completed: Sprint 2
21	As a user, I would like to read the rules of the game anytime.	9	Completed in Sprint 1
22	As a user, I would like to be able to see enemies health.	8	Completed in Sprint 1

Non-Functional

Backlog ID	Functional Requirements	Hours	Status
23	<u>The application should be supported by the two most popular operating systems: Microsoft Windows and MacOS.</u>	20	In Progress : Moved to Sprint 2
24	The application should be attractive and easy to use so users will be able to learn and play efficiently.	8	Sprint 2
25	<u>As a developer, I would like to store information securely on the database</u>	10	In Progress : Moved to

			Sprint 2
26	<u>The database should be able to handle multiple rapid requests at once.</u>	10	In Progress : Moved to Sprint 2
27	<u>User information should be stored securely.</u>	8	In Progress : Moved to Sprint 2
28	As a user, I would like the game to be balanced in terms of difficulty	4	Sprint 2
29	As a developer, I would like to learn how to use the Unity and Unity Collaborate.	4	Completed: Sprint 1 & 2
30	As a developer, I would like to learn how to use the GameSparks backend.	6	Sprint 1 & 2

User Cases:

Case: Account creation

Action:	System Response:
1. Click create an account	2. System asks for desired username and password
3. Enter desired username and password	4. Account is made and saved in database

Case: Building a tower

Action:	System Response:
1. Click on tower you want to build from tower creation menu	
2. Click on map where you would like to build the tower	3. Tower is built on the tile given you have enough currency.

Case: Sending offensive units

Action:	System Response:
1. Click offensive unit you would like to send	2. Unit is sent given that you have enough currency

Case: Read instructions/rules before game starts

Action:	System Response:
1. Click read instructions when on main menu	2. Instructions page is displayed
3. Click close	4. Instructions page is closed and main menu shows

Case: Upgrading units

Action:	System Response:
1. Click on already built unit	2. Unit information shows up
3. Click upgrade	4. Unit gets upgraded

Case: Basic combat and game flow

Action:	System Response:
1. Enemy moves along the path	
2. Defenses shoot at enemies	3. Enemies take damage and enemy hp bar shows. If enemy hp reaches 0, enemy dies
4. Enemy is defeated	5. Currency is rewarded
6. Last enemy is defeated	7. Wave is cleared, next wave countdown started, and wave clear currency is awarded
8. Last wave is cleared	9. Level is beaten, and victory screen shows up.
10. Click back to menu on victory screen	11. Main menu is shown

Case: Saving the game

Action:	System Response:
1. Click on open pause menu	2. Pause menu opens up, game is paused in background
3. Click on save game.	4. Game data is saved

Case: Tower targeting enemies

Action:	System Response:
1. Enemy comes into range of tower	2. Tower attacks enemy
2. Enemy dies or goes out of range	Tower stops attacking and searches for next target if there is one in range

Case: Moving the camera

Action:	System Response:
1. Press up arrow key	2. Camera moves upward on the map
3. Press right arrow key	4. Camera moves towards the right on the map
5. Press left arrow key	6. Camera moves leftwards on the map
7. Press down arrow key	8. Camera moves downwards on the map