

Tower Defense [2018]

CSC 335 – Final Project

All information in this document is subject to change



Credit:

<https://www.epicwar.com/assets/p/10/2294.jpg>

Background

RTS (Real-Time Strategy) games have become popular due to their demanding nature in requiring players to employ a long-term strategy with upkeep of units or resources in order to fulfill a certain objective. Tower defense games are a specific type of RTS games – they require the player to prevent enemies from reaching a target location by building towers to defeat them. Some tower defense games rely on the player to build a path for the enemies whereas others specify a static map with a defined path for enemies and have the player build towers alongside the path. Examples of tower defense games are: Bloons Tower Defense, Gemcraft, Kingdom Rush, and Plants vs. Zombies, Warcraft III Tower Defense Custom maps, etc...

Overview

Your job is to design a tower defense game. The background and story of your game is entirely up to you. The maps should contain pre-defined paths for enemies to travel, but it is up to you to decide how the game is being played.

Players must be able to purchase towers and place them in the map to attack enemies. These towers prevent enemies from entering the End-Zone to prevent losing player lives. Your game should be single player. You are allowed to draw inspiration from existing games, but you must strictly adhere to the requires listed in this spec. Creativity is highly encouraged.

Requirements (90%)

Starting Screen:

- You should have a starting screen prior to the maps.
- The players should have the choice of choosing between maps.

Maps

- You should have at least 3 Maps.
- Each map has a background image and contains a pre-defined path for enemies such that enemies originate at their start location and move along this path towards the player's destination and toward the end-zone.
- Different maps have different backgrounds and paths that enemies travel.
- One of the map contains more than one path for enemies to travel on (so it presents a more difficult gameplay, not a linear path! The path should diverge at some point)
- A player can select and play on at least three maps. The path the enemies travel must be different – you cannot just change a background image and call it a new map). These 3 maps should be built-in and selectable at game startup.
- Maps should have space on the sides of the path for players to build towers.

Enemies

- Enemies continuously Spawn from the map at some location. We call this the Spawn-Zone.
- Enemies move towards the destination that the player will defend. We call this the End-Zone
- Movement of enemies is animated and should not move like the hunter in your Wumpus Project.
- Each enemy has certain stats/characteristics. Example:
 - Speed
 - Defense
 - High/Low Armor
 - Resistance to certain buffs (Poison, fire, ice, etc..)
 - Be Creative!

You can choose whether this difference applies to either each individual enemy that is spawned on the map or whether it applies to each type of enemy.

- A player can view information about an enemy by clicking one that has been placed. Information should include the characteristics of that enemy (speed, defense, armor, ... your choice).
- Information should be viewed in an inclusive manner within the game. Stay away from Java Pop-ups!
- Each map should have 3 different type of enemies. A Minimum of total 9 enemies. You can mix and match this maps.

Towers

- Towers are purchased by the player and placed strategically along the paths.
- Towers will attack enemies that pass by (in its range of attack). Examples:
 - Gemcraft uses towers powered by gems and the combination of gems determine damage, range, etc.
 - Kingdom Rush uses a medieval setting with archery towers and warrior towers.
 - Be creative! You can decide how towers should attack enemies. It can be single target, or multi-target!
- Each tower has a different damage amount, range of attack, and rate of attack.
- Each tower has a different way of attacking enemies. Possibilities include
 - attack multiple enemies,
 - teleport enemies,
 - slow enemies,
 - poison enemies,
 - confusing enemies so that they either walk back towards their start or maybe they spin in a circle for a few seconds, etc.
 - Be Creative!
- A player can view information about a tower before purchasing it and/or by clicking one that has been placed.
- Each tower should cost the player something, could be gold, etc..
- Towers must be upgradeable to have more range, fire rate, or damage.
- Tower upgrades change the look of the tower.
- Towers must have a minimum of 1 tower upgrades.
- A tower's range is visible when a tower is clicked. If a tower is upgraded, this information should also be updated.
- Tower attacks are animated.
- Towers cannot be built such that they block the path of the enemies.
- The player can build at least three different towers. These three towers must be inherently different. You cannot just change the initial damage and range of a tower, for example, for a tower to qualify as different. A different tower might involve a different type of attack or a different animation. (e.g. a lightning tower that attacks 3 enemies at the same time)

Animation

- When the game begins in the map, it should be animated. That includes all sprites moving.
- All attacks from the towers must be animated.

Player

- Players can buy towers through some form of currency (money, points, mana power, life-force, etc.).
- There is a graphical way for the player to place the tower at the desired location (e.g. change the cursor to look like the tower and use a mouse click, select a grid square, etc.)
- Players should start with an appropriate amount of currency and can obtain this currency during game-play by defeating enemies.
- Player should be able to interact with his tower always.
- Players can see their current amount of currency throughout the game.
- Player should largely use the mouse as device to play the game

Win/Lose Conditions

- The player loses the game after a certain number of enemies breach the defenses and reaches the End-Zone.
- The player can win the game by fulfilling some objective. Example:
 - Successfully defeating some number of enemies
 - Defeating a boss enemy
 - Defending the destination for a certain amount of time
 - Be Creative!

Sound

- Your game should have all sound effects for major events as discussed with your Scrum Master.
- Examples:
 - Spawning of enemies
 - Each enemy dying
 - Enemies entering spawn zone
 - General Sound music throughout the game
 - Think Creatively!

Persistence

- The player should be able to save the game and load a saved game outside gameplay of the map or when the game is paused. That is, after completion (win or lose) of a map, at the starting-menu, a save option should be available (to keep update on his latest progress), or when the game is pause momentarily while in the map.

User-Guide

- You are required to have an instruction page. Treat this as a user-guide to how to play your game. This guide should be found in the starting menu
- The form of this can come in a variety of ways but should be within a game. No java-pop ups.
- The minimum expectation is that a new player of your game will have all the knowledge of the game mechanics to successfully play once through.

Summary:

- The player can read the User-Guide of the game anytime before game-play.
- The player first reaches the Starting Menu before selecting a Map
- Persistence is implemented in the Starting Menu
- In-game message should be displayed after a map is won/lost. This can be in the form of statistics (tower kills, etc...)
- You may invent a version of gameplay or use one of the two popular versions of game-play:
 - Continuous: Enemies continuously spawn and the player builds towers to defend the destination. The game ends when the destination has been defended for a specified time period or when a certain number of enemies have been defeated.
 - Waves: A collection of enemies form one wave. Each wave of enemies spawns with a short pause in between waves. The game ends when all waves have been defeated.
- The player can pause and resume the game in single-player mode.
- The player can toggle the speed of the game between a normal speed and a fast speed. This feature is not required for multi-player mode.
- Save/Load Games: When playing single-player mode, the player can choose to save the game and play at a later time. A Load Game option is in the main menu. This feature works well with the Pause Game feature, since it is more elegant to load a game, have it paused initially, and then resume to play the game.

More Information:

More visualizations and inspirations can be found here:

Gemcraft:

<https://youtu.be/nJu6Dq1KSms?t=14m49s>

Plant vs Zombies:

<https://youtu.be/hQLsVTXb5RA?t=2m20s>

Warcraft III Elemental Tower Defense Custom Map:

<https://youtu.be/e3YXZoUorv0?t=5m12s>