

Tower Defense

Project Manager: Rohit Umashankar

Overview:

RTS (Real-Time Strategy) games have become popular due to their demanding nature in requiring players to juggle 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 (or a multilevel system) 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, to name a few.

Your Final Project:

For your final project, you will be constructing your own tower defense game. It will contain a pre-defined path for enemies to travel, but it is up to you to decide how the game is played. Players can purchase towers and place them in the map to attack enemies. Your game will have a single player mode and a multi-player mode (additional modes of play can be included for extra credit). You are allowed to grab ideas from existing games, but you must adhere to the requirements listed below. Creativity is highly encouraged.

Requirements:

- ✘ Maps
 - + Maps must be static – there is a pre-defined path for enemies such that enemies originate at their start location and move along this path towards the target.
 - + Maps should be laid out in a grid. Elements, such as towers and enemies, should not overlap. (Overlap during animation is okay.)
- ✘ Enemies
 - + Enemies must have a source from which they come into the map. They will continuously spawn at an appropriate rate.
 - + Enemies must have a destination that they target.
 - + You must have at least three different types of enemies. Enemies can differ in their defense (some enemies can be harder to defeat than others) or their speed

- (some enemies can move faster through the map) or any other characteristic or combination of characteristics of enemies. You can choose whether you want enemies to attack the towers (see extra credit section).
- + Enemies must move towards the destination. Movement of enemies must be animated.
 - ✘ Towers
 - + Towers must have some way to attack enemies that pass by. For example: 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. (See extra credit for multiple tower types)
 - + You decide how towers should attack enemies (e.g. whether they should attack one enemy until it moves out of the tower's range or whether it attacks enemies at the edge of its range before attacking enemies close to it ... whatever makes sense for your game)
 - + Towers must have a damage amount, range of attack, and rate of attack. Players can view this information before purchasing the tower.
 - + Tower attacks should be animated.
 - + Towers cannot be built such that they block the path of the enemies.
 - ✘ Game-play & Rules
 - + The player should be able to read the rules of the game before game-play.
 - + There is some main menu or “home page” where the player can choose to play a single-player game or a multi-player game. After a game ends, the player should return to this menu.
 - + There must be a purchase system – towers cost something (money, points, mana power, etc.). Players should start with an appropriate amount and also be able to obtain this currency during game-play by killing enemies. Players should be able to view their currency during the game.
 - + The player should lose the game after a certain number of enemies breach the defenses and reach the destination. Again, you can exercise your creativity here. One way (out of many) is to have the “destination” as the headquarters for the player, that also has health. Enemies can be designed to do some amount of damage to the headquarters.
 - + The player can win the game by fulfilling some objective (either successfully defeating some number of enemies, defeating a boss enemy, etc.)
 - ✘ Multi-player PVP (Player versus Player)
 - + Players should have the option of playing a game against another player over the

network.

- + Each player only sees their own map, but both players' maps have the same path for enemies. Players will start with some amount of currency (usually larger than single player mode).
- + Both players will be responsible for building towers and spawning units. (Here, "units" refers to the "enemies" in single-player mode). Spawned units will appear on the other player's map as the enemies that they need to kill.
- + All units (at least 3 types of enemies required) should be available to spawn. They should have different costs to spawn, based on their stats.
- + Each player will also need to see some helpful information about their opponent, the units they send, etc. that will aid in their strategy.
- + If at any point in the game there are no units in play and both players do not have sufficient funds to spawn units, the game ends in a tie.
- + A player wins when a certain number of his units breach the opponent's defenses.
- + A win/lose message should be shown when the game ends. This requirement can be replaced with a game statistics message (see extra credit).

Expectations

This project lends itself well to object-oriented design concepts. You are expected to follow good object-oriented design principles and ideas we've covered this semester. Please make use of inheritance, polymorphism, and design patterns when appropriate.

Extra Credit (10 points max):

This is by no means an exhaustive list of extra credit opportunities. Discuss any extra credit ideas you have with your project manager to ensure that your extra features will receive the points that you expect. Some of the suggested extra credit opportunities will be much harder to implement than others, and are worth more accordingly. Extra credit ideas not listed will receive points based on their difficulty.

- ✘ Attacking enemies (2 points): Enemies can attack towers as they pass. Towers have health and will get destroyed when this health goes to 0. Players can repair towers for a small cost.
- ✘ Upgradeable towers (2 points): Towers can be upgraded at least twice to have more range, fire rate, or damage. Upgrades change the look of the tower.
- ✘ Chat System (2 points): You include a chat system in multi-player mode.
- ✘ New Tower Types (1 point each): Introduce new tower types. These new towers

must be inherently different than towers you may already have. 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)

- ✘ Story Mode and Quickplay Mode (2 points): The requirements do not explicitly necessitate a story mode, but you can add one. Develop your single player game to follow a simple plot. Include a quickplay mode, where a player can play a normal single-player game. Add Story and Quickplay Mode options to the main menu.
- ✘ Map Editor (5 points): The player can access the map editor from the main menu. The map editor allows for the creation of new maps in a graphical window and includes load/save functionality for maps. Created maps must have a defined start and end locations as well as a complete path for enemies from this start to end. Players should be able to play on created maps.
- ✘ Multiple Maps (1 point): The player can play on at least 3 different 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 (you can't say that you have a map editor, so a player *can* play on 3 maps). However, you can use your map editor to create three maps and save them for this purpose. This feature is well-suited to go along with the story mode and quickplay mode.
- ✘ Pause Game (1 point): The player can pause and resume the game.
- ✘ Toggle Speed (1 point): The player can toggle the speed of the game between a normal speed and a fast speed.
- ✘ Save/Load Games (1 point): When playing single-player mode, the player can choose to save the game and play at a later time. A Load Game option should be 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.
- ✘ Sounds (1-2 point(s)): Incorporate sounds into gameplay for single-player mode. More points for well-incorporated sounds. No points for random sounds.
- ✘ Game Statistics (1 point): Collect statistics about the game (e.g. enemies destroyed, towers built, etc.) and show them to the player after a game ends. Statistics need to be shown for both single-player and multi-player modes.
- ✘ Temporary Powerups (3 points): Killing an enemy can sometimes give you a powerup. You can decide whether that powerup applies to nearby towers or all towers. You must have at least two different kinds of powerups. Players must know when a powerup is active and how long it's active. For example, a “Slow” powerup might slow enemies for 20 seconds.