

# **CDR Tower Defense: Project** **Documentation**

## **Oct 4th, 2017:**

- Basic templates have been laid out for potential code classes. This will help our team stick to the initial design we have created and discourage creation of unnecessary classes. Our goal is to make our code as simple and clean as it can be from the very beginning. This way, in the later stages of the project, our large code collection should be as easy as possible to understand and work with.

## **Oct 9th - Oct 12th, 2017:**

- We have created our prototype data file for our game to read from. It is a text file that contains characters which encode the information for the game map. Included in it is symbols representing the following: blank tiles, directed path tiles (up, down, left, right), start tile, end tile (the base). The position of these symbols in the data file encode the locations of every possible spot on the map.
- The code for reading the data file and producing the map is completed. The next step is to have the GUI take this map data and create a graphical map for our game window.
- The implementation for the map graphics is complete. The program now allows you to select "Start" from the main menu which results in the game's map being displayed.

## **Oct 15th, 2017:**

- Logic has been written which causes an animation of an enemy travelling along an arbitrary path on the map (each map only has a single path which is defined by the data file). When enemies reach the base, they disappear so as to appear that they are self-destructing (however, health has not yet been implemented).
- Several new map data files have been added to the project in order to test different path configurations and ensure that path creation and enemy movement logic is robust.

**Oct 26th - Oct 27th, 2017:**

- Code has been written that allows the following: a custom number of enemies to travel along the map path simultaneously, a pause/play button feature to use during gameplay, and a tower store where defenses are to be purchased from.
- The enemy movement animation is now smoother. Enemies also smoothly enter the map from the very edge for a more natural graphical effect.
- Some minor refactoring has been done to tidy up the code our team has written so far. We are still focussing on keeping the software as simple as it can be, with minimal coupling between classes and no unnecessary complexity.
- An initial version of a base health and a currency system has been implemented. Enemies interacting with the base causes its health to be affected negatively. Enemies do not yet have health implemented.
- The project's primary gameplay algorithms have been connected onto a custom new timer system, which should allow for better efficiency and smoother gameplay.

**Oct 29th - Oct 30th, 2017:**

- Tower purchasing using currency, an enemy health system, and enemy death currency reward all implemented.
- Our team had a long meeting in which we discussed the direction to proceed in with our project. Our next task is to do the following:
  - Build logic for towers interacting with enemies.
  - Create a game over window to be displayed if the player loses.
  - Add a map selection feature to the main menu.
  - Write a game guide that can be viewed from the main menu to teach players how to play CDR Tower Defense.
  - Add tower descriptions in a text box area which describes the tower currently selected in the store.

**Nov 1st - Nov 3rd, 2017:**

- Losing a game now causes a game over window to show on screen
- Towers you own can now be sold back to the store for a fraction of the currency you originally paid for them.
- Added the game guide feature to the main menu.
- Created a grey and cyan color theme for our menus and map. We want our game to be soothing to a user's eyes! Also, our game is now dressed up in a computer science theme with a circuit board map and a CPU base.
- We have finalized the title of our game as "CDR Tower Defense".
- Logic for creating waves (rounds) of enemies has been implemented.

- An in-game label displaying the current round the player is on has been added.
- Two new tower types have been added to the game (Melee and Sniper).

#### **Nov 14th - Nov 17th, 2017:**

- The visual size of enemies has been increased to better display their health value.
- Every ten rounds, a single boss enemy with a large amount of health appears.
- Map size has been finalized with a maximum of 11rows x 13columns (can have less if desired, but not more).
- Another map has been added.
- The game's health and money systems have been balanced so as to not be too hard or too easy.
- The ability to combine any fully upgraded tower with any other different tower has been implemented.
- Attack range indicator for towers has been added.
- A "speed attack" tower has been added to the store.

#### **Nov 20th - Nov 22nd, 2017:**

- Boss enemies are now a unique color.
- Added background music and enemy death sound effects.
- Fixed the attack radius of towers to improve attacking enemies within range.
- Added example maps to show how our program gives an error message when a map has invalid dimensions.
- The main menu map list now dynamically updates when new map files are added to the "maps" folder.

#### **Nov 24th - Nov 28th, 2017:**

- Modified the cost of combining towers.
- Added unit tests using JUnit.
- Created an executable jar file to run our game easily.
- Added a toggle button for sound effects.
- Added text file giving credit to the music creator and the sources of images utilized in our game.
- Renamed the towers in our game with appropriate computer defense terms which best suited the towers individual attributes (Antivirus, Firewall, Quarantine, Encryption, & Surge).
- **/\*GAME DEVELOPMENT COMPLETE.\*/\***

# CDR Tower Defense: Code Overview

## *Code Packages:*

- **entities** – Contains the classes for the basic entities which exist in CDR Tower Defense.
  - **Base** – Logic related to the base.
  - **MapData** – Logic for parsing the data file and priming the data to be used by our algorithms.
  - **Path** – Logic for defining, creating, and validating paths.
  - **Tile** – Properties for anything that exists on a map tile are defined here. Many of the classes in “entity” inherit from the Tile class.
  - **Tower** – Tower logic is defined here.
- **graphics** – Contains the classes which draw the basic graphical components of the game.
  - **EnemyComponent** – Defining of all graphic properties of enemies are done here.
  - **MapComponent** – Defining of all graphic properties of maps are done here.
  - **TileComponent** – Defining of all graphic properties of tiles are done here.
- **music** – Contains the classes for running the games music and sound effects.
  - **BackgroundMusic** – This class has the code for being able to add background music to our game.
  - **DeathSound** – This class has the code for sound effects (sound for enemy deaths is implemented right now).
- **ui** – Contains the classes for creating the different types of windows for the user to interact with.
  - **GameGuideWindow** – This class builds the game guide window.
  - **GameOverWindow** – This class builds the game over window.
  - **GameWindow** – This class builds the game play window.
  - **Main** – This class only contains the main method for running our game.
  - **TitleWindow** – This class builds the main menu window.