CS 467
Winter 2017

Crater: Project Plan

Crater Team Members:
Ryan Fleming
Rick Garnica
Adam Much

Team Crater has built a two dimensional tower defense game using HTML5, JavaScript, and CSS. All of us consider ourselves to still be beginners to web development, and this project really honed our web development skills and gave us more practice with JavaScript. Our tower defense game has a "Black Friday" theme.

Game Objective

The player's objective is to keep people from crowding into stores that are having Black Friday sales. If too many people get into the store, a fire code will be violated, which shuts down the store. Don't let 20 or more customers into each store by defeating all enemies with offensive towers. Successfully defend the storefronts on Levels 1,2,3 and 4.

Game Play

The user will view the game from a top-down perspective. This view will allow the user to easily view the flow of customers and place towers accordingly. The towers are security themed obstacles. The levels represent individual stores that resemble familiar stores at which we all shop. The customers attempting to enter the store have different characteristics. The towers work better or worse depending on the characteristics of the customers, which will force the user to develop a strategy for repelling varying customers. There are four different game boards that increase in difficulty as the user progresses. The target user has already played some type of tower defense game like Kingdom Rush and will have a familiarity for the objective and controls and tower defense game requires.

Health: This is the number of enemies that can enter the store without breaking fire code. When it reaches 0, you lose. This is replenished whenever you defeat a level. This is displayed on your HUD.

Money: This is how much money you have remaining to purchase towers. Different towers have different purchase prices. This is replenished by defeating enemies. Different enemies offer different bounties. This is displayed on your HUD.

Enemies Remaining: This is how many enemies are left on your current level. Once you have defeated all of them, You move on to the next level. This is displayed on your HUD.

Active Tower: This is the type of tower you will build when you select a Strategic Point. You can change what type of tower you build by using the `DIRECTIONAL KEYS` on your keyboard. This is displayed on your HUD.

Level: This is what level you are on. This is displayed on your HUD.

Wave: This is what wave number is currently trying to enter your store. This is displayed on your HUD.

Figure 1: In game screenshot of HUD

Strategic Points/Build Tiles: This is where you are allowed to build towers. They are highlighted neon green. You activate them by clicking on them. Whatever tower you currently have selected as your Active Tower is what you will build.

Enemies: They advance towards your store entrance and try to get inside. Some are stronger than others.

Bargain Shopper: The weakest enemy in the game. They are just trying to "get a deal" and aren't really sure what they want to buy from each store. They may pose a threat when in large groups, but are very weak by themselves.

- 16 hit points
- 20 (Medium) speed
- Counts for 1 against fire code
- Bounty $3

 **Figure 2: In game screenshot of Bargain Shopper**

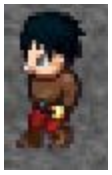
Athlete: Essentially the players' introduction to much more physical enemies. This enemy works out a lot, so is much stronger than the bargain shopper.They are still quite weak and shouldn't be worried about if the player has at least one or two Lightning Taser Towers.

- 64 hit points
- 20 (Medium speed)
- Counts for 1 against fire code
- Bounty $9

 **Figure 3: In game screenshot of Athlete**

Software Engineer: This is the fastest enemy in the game. Normally not found outside, this shopper is extremely excited about having free time after graduation and can move very quickly. They are motivated to find one of the few Nintendo Switches left at one of the stores. They are still very weak, and Candy Cane Barricade towers can stop them in their tracks.

- 28 hit points
- 35 (Very Fast) speed
- Counts for 1 against fire code
- Bounty $5

 **Figure 4: In game screenshot of Software Engineer**

Family with Small Children: This is the strongest enemy in the game, with much higher health. They prove to be a great threat against your defense. This family is extremely motivated to get the hottest seasonal gift at a very low price. Having to drag small children all the way, they move very slow. Using Candy Cane Barricades to weaken them to other Towers is your best bet for success.

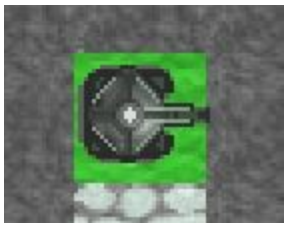
- 640 hit points
- 15 (Slow) speed
- Counts for 3 against fire code
- Bounty $50

 **Figure 5: In game screenshot of Family with Small Children**

Towers: These are used to prevent waves of shoppers from entering storefronts.

Packing Peanut Projectile Turret: This shoots packing peanuts into enemies, dealing physical damage. Select this tower and make it your Active Tower by pressing `LEFT` or `1` on your keyboard.

- $70 purchase cost
- 7 damage
- 90 (Very Fast) Cooldown
- 6 Tile (Vary Far) Range

 **Figure 6: In game screenshot of Packing Peanut Projectile Turret**

Candy Cane Barricade: This shoots a blank projectile into enemies, slowing them down and sending them into a "falling" animation which blocks their progression. It leaves enemies susceptible to 4x the damage from other towers. Select this tower and make it your Active Tower by pressing `RIGHT` or `4` on your keyboard.

- $70 purchase cost
- 9 damage
- 500 (Very Slow) Cooldown
- 3 Tile (Short) Range

 **Figure 7: In game screenshot of Candy Cane Barricade**

Lightning Taser Tower: This shoots a lightning taser projectile into enemies, dealing electrical damage. Select this tower and make it your Active Tower by pressing `UP` or `2` on your keyboard.

- $100 purchase cost
- 40 damage
- 180 (Slow) Cooldown
- 3 Tile (Short) Range


Figure 8: In game screenshot of Lightning Taser Tower

Pepper Spray Can Tower: This shoots pepper spray at enemies, dealing area damage. Select this tower and make it your Active Tower by pressing `DOWN` or `3` on your keyboard.

- $125 purchase cost
- 30 damage
- 180 (Slow) Cooldown
- 3 Tile (Short) Range


Figure 9: In game screenshot of Pepper Spray Can Tower

User Instructions

Play using Chrome
The game can be found hosted at this URL:

Black Friday Tower Defense Game:
http://people.oregonstate.edu/~fleminry/CS467FinalProject

Starting the Game
Going to the URL will start the game right up in your browser. If you want to test on a local machine make sure you have node installed. Then enter the following commands:

```
npm install
npm install -g grunt-cli
grunt serve
```

The game will then be running on https://localhost:8000

Navigating the Tower Defense Game

Title Screen: After the game engine loads the player will be brought directly to the Title Screen. This includes some text as instructions on how to launch the first level. Press `ENTER` on your keyboard or `LEFT CLICK` on your mouse to begin Level 1. Press `F` on your keyboard to enter fullscreen. **NOTE**. For instructor purposes to skip a level if desired, when you are at this screen enter `game.data.level = [desired level];` in the console to go automatically to the desired level when you press `ENTER.`

Figure 10: Sprite for Title Screen and Ready Screen

Ready Screen: Whenever the player defeats all enemies in a level, the game moves on to a Ready Screen before the next level is loaded. This includes some text as instructions on how to launch the next level. Press `ENTER` on your keyboard or `LEFT CLICK` on your mouse to begin the next level.

Game Over Screen: When a player's Health becomes zero, the player loses the game. This brings them to a Game Over Screen. This includes some text as instructions on how to start over and try again. Press `ENTER` on your keyboard or `LEFT CLICK` on your mouse to start at Level 1 again.

Figure 11: Sprite for Game Over Screen

Victory Screen: When a player defeats all enemies on the last and final level, the player is brought to the Victory Screen. Congratulations! You Won!

Figure 12: Sprite for Victory Screen

NOTE MelonJS has a feature called pauseOnBlur that pauses the game when the user enters a different tab or application. This is useful for most games but not for games that require timed functions such as waves for a tower defense game. Timeouts can be set to respect the game's pause state but currently MelonJS does not have the functionality for Timeouts to respect pauseOnBlur. This is something MelonJS is working on to include in future releases of the game engine. Turning pauseOnBlur off causes canvas rendering issues. **When you are in the play state of the following 4 levels, please refrain from using other tabs in your browser or other applications or you may run into wave timing problems.** Feel free to use those tabs or other applications in any of the other screens described above.

Level 1: The Mall This is the first, introductory level in the game comprised of ten waves. The player starts with 12 Strategic Points to build towers at. The player starts with $265 to spend on defenses to keep shoppers out of the mall. Shoppers spawn at different points on the map and try to progress to the mall entrance on the right of the screen.

Wave	Enemies	Income in $	Time in Seconds
1	3x Bargain Shopper	9	15
2	3x Bargain Shopper	9	40

3	6x Bargain Shopper	18	65
4	6x Bargain Shopper	18	90
5	9x Bargain Shopper	27	115
6	9x Bargain Shopper	27	140
7	4x Bargain Shopper 1x Athlete	21	165
8	3x Athlete	27	197
9	10x Bargain Shopper 4x Athlete	66	229
10	16x Bargain Shopper	48	261

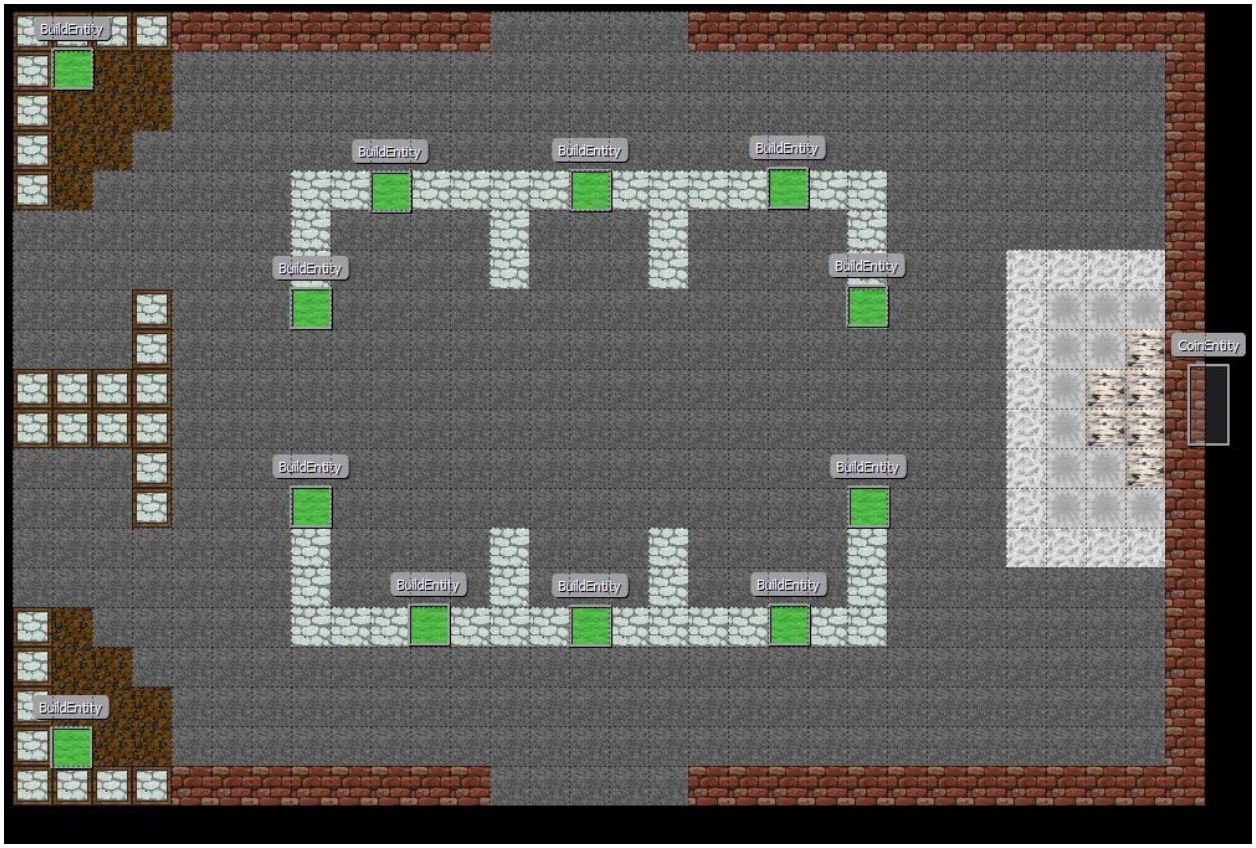


Figure 13: Tiled Map for Level 1: The Mall

Level 2: Bullseye Discount Store This is the second level in the game comprised of ten waves. The player starts with 12 Strategic Points to build towers at. The player starts with $220 to spend on

defenses to keep shoppers out of the store. Shoppers spawn at different points on the map and try to progress to the store entrance on the left of the screen.

Wave	Enemies	Income in $	Time in Seconds
1	20x Bargain Shopper	60	15
2	20x Bargain Shopper	60	37
3	24x Bargain Shopper	72	59
4	24x Bargain Shopper	72	81
5	12x Bargain Shopper 4x Athlete	72	103
6	12x Bargain Shopper 4x Athlete	72	125
7	6x Software Engineer	30	140
8	20x Bargain Shopper 6x Athlete	114	155
9	10x Bargain Shopper 6x Athlete 6x Software Engineer	114	170
10	20x Bargain Shopper 10x Athlete 10x Software Engineer	200	185

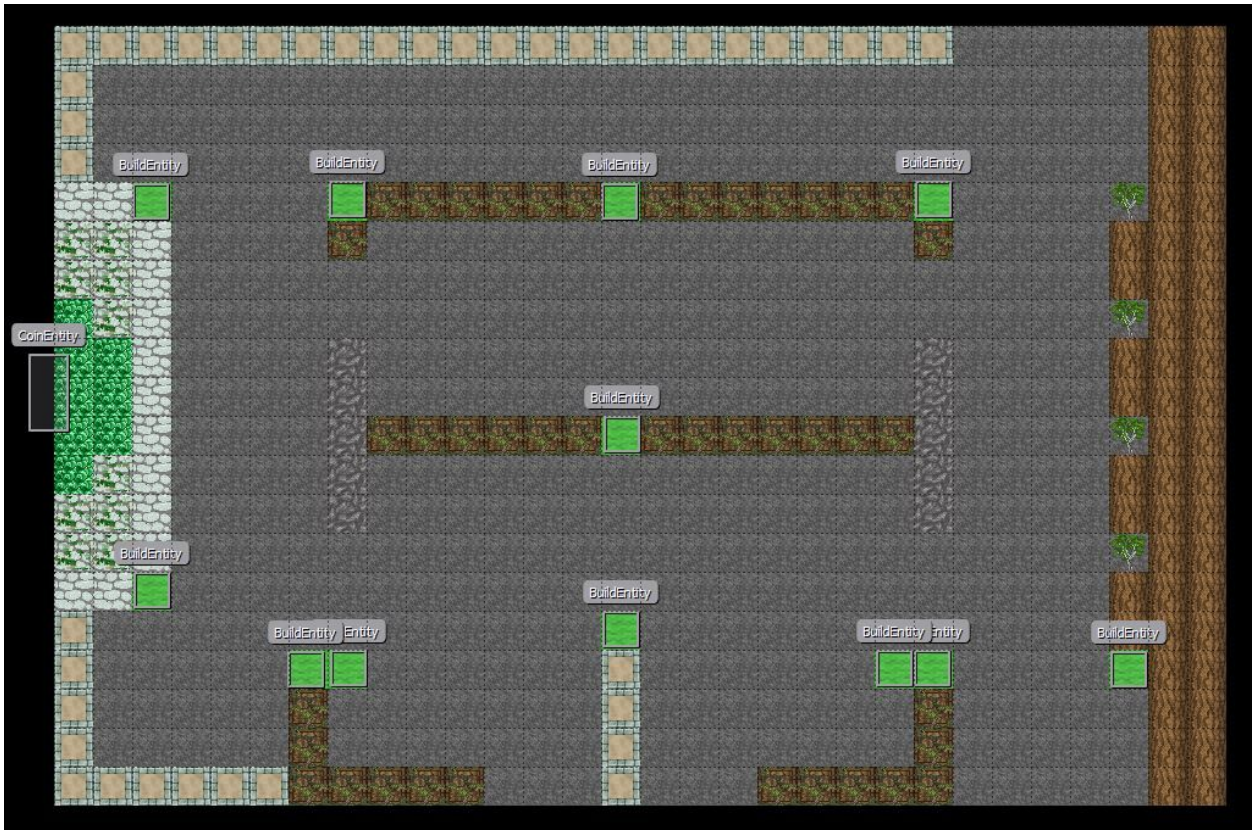


Figure 14: Tiled Map for Level 2: Bullseye Discount Store

Level 3: Good Purchase Consumer Electronics This is the third level in the game comprised of ten waves. The player starts with 24 Strategic Points to build towers at. The player starts with $300 to spend on defenses to keep shoppers out of the store. Shoppers spawn at different points on the map and try to progress to the store entrance on the right of the screen.

Wave	Enemies	Income in $	Time in Seconds
1	24x Bargain Shopper	72	15
2	5x Bargain Shopper 5x Athlete	60	36
3	12x Software Engineer	60	75
4	40x Bargain Shopper	120	90
5	25x Bargain Shopper 10x Software Engineer	125	108
6	10x Athlete	90	126

7	25x Bargain Shopper 12x Athlete	183	151
8	25x Bargain Shopper 12x Athlete	183	169
9	20x Bargain Shopper 16x Athlete 10x Software Engineer	254	187
10	20x Bargain Shopper 1x Family with Small Children	110	212

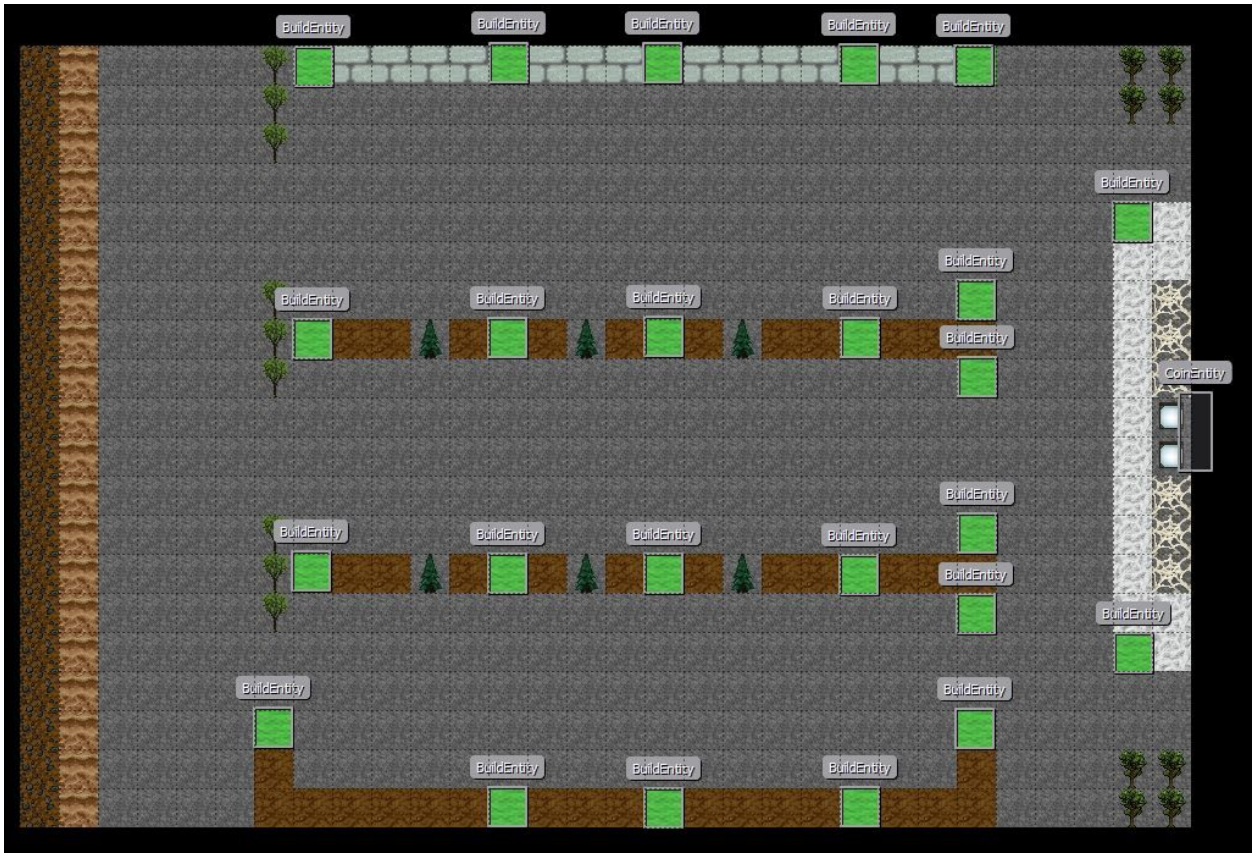


Figure 15: Tiled Map for Level 3: Good Purchase Consumer Electronics

Level 4: Floormart Hypermarket This is the fourth and final level in the game comprised of ten waves. The player starts with 21 Strategic Points to build towers at. The player starts with $680 to spend on defenses to keep shoppers out of the store. Shoppers spawn at different points on the map and try to progress to the store entrance on the top of the screen.

Wave	Enemies	Income in $	Time in Seconds
1	20x Bargain Shopper 6x Athlete	114	15
2	20x Bargain Shopper 8x Athlete	132	36
3	20x Bargain Shopper 5x Athlete	105	75
4	50x Bargain Shopper	150	90
5	20x Bargain Shopper 9x Athlete	141	108
6	18x Software Engineer	90	126
7	5x Athlete 1x Family with Small Children	95	151
8	14x Athlete	129	169
9	8x Athlete	72	187
10	2x Family with Small Children	100	212

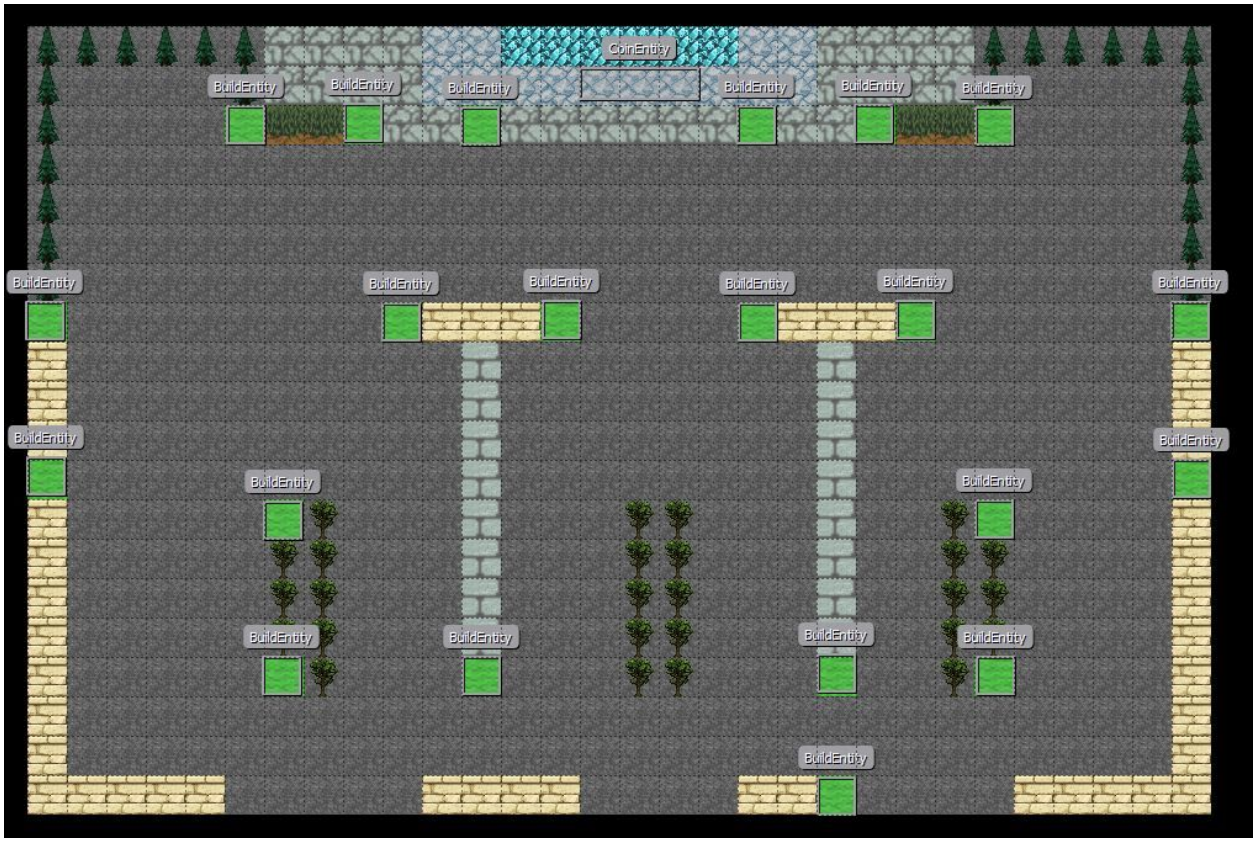


Figure 16: Tiled Map for Level 4: Floormart Hypermarket

Technologies Used

- Basic Languages
  - HTML5
  - CSS
  - JavaScript
- Frameworks and Libraries
  - Melon.js for game engine
  - Tiled for creating game boards
  - Texture packer for creating sprites
  - Piskelapp for creating sprites
  - http://gaurav.munjal.us/Universal-LPC-Spritesheet-Character-Generator/# for creating sprites
- Sources for images and sounds
  - Music from https://www.dl-sounds.com/royalty-free/category/game-film/video-game/
  - Sound effects from http://freesound.org/
  - Level tiles from Game Art Dev
  - Fonts from Littera
- Text Editors and IDEs

    - Ryan Fleming's preferred Text Editor: WebStorm 11
    - Adam Much's preferred Text Editor: notepadqq
    - Rick Garnica's preferred Text Editor: Sublime 3
- Version Control
    - https://github.com/RyanFleming/CraterCS467/
- Communication
    - https://cs467towerdefense.slack.com/messages/general/

MelonJS structure

`/build/js/resources.js`
Loads all of the resources into the game engine. This file is automatically produced by MelonJS upon a build.
`/data/bgm/`
`/data/fnt/`
Contains font files
`/data/img/gui/`
Contains sprites for our screens.
`/data/img/map/`
Contains tiles used for creating maps in Tiled
`/data/img/sprite/`
Contains all sprites for enemies and towers
`/data/map/`
Contains all tmx files for our Levels
`/data/sfx/`
Contains all sound effects and music
`/icons/`
`/js/entities/`
Contains all source code for game elements. This includes towers, projectiles, enemies, pathfinding, the enemy spawn manager, and HUD.
`/js/screens/`
Contains all source code for game play states. This includes, the title menu, loss, victory, loading, and the active gameplay screen.
`/js/game.js`
Contains the source code for the game itself. It loads in the resources in `/build/js/resources.js,` the elements in `/js/entities/,` the play states in `/js/screens/,` and runs them together as a complete game.
`/lib/plugins/debug/debugPanel.js`
Contains the debug panel. This came with MelonJS.
`/lib/melonJS.js`
The game engine
`/tasks/grunt-resources.js`
Used for building and deploying the game

`Gruntfile.js`

Used for building and deploying the game

`Index.css`

Used for displaying the game in browser

`Index.html`

Used for displaying the game in browser

`main.js`

Used for displaying the game in browser.

Team Member Accomplishments

Rick Garnica

- Created 4 Levels using the Tiled level editor
- Created sprite sheets for Bargain Shopper, Athlete, and Software Engineer
- Create sprite screens for Title, Game Over, and Victory Screens
- Created sprites for Packing Peanut Projectile, Turret, Lightning Taser Tower, Candy Cane Barricade, and Pepper Spray Can Tower
- Created sprites for Packing Peanut, Lightning Taser, Pepper Spray, and Blank projectiles
- Sourced game music and sound effects for towers
- Implemented full screen option for game
- Implemented music into screen states
- Sourced font for game

Adam Much

- Developed pathfinding for enemy class
- Developed tile array for enemies to find direction.
    - directions.js
- Developed walking animation system for enemy class
- Implemented source code for Bargain Shopper, Athlete, Software Engineer, and Family with Small Children
    - enemy.js
    - enemy2.js
    - enemy3.js
    - enemy4.js
- Spliced together sprites to make the Family “cluster” sprite complete with walking animation
- Developed tower targeting system
- Developed projectile collision system
- Developed sound effects
- Implemented source code for Packing Peanut Projectile Turret, Candy Cane Barricade, Lightning Taser Tower, and Pepper Spray Can Tower
    - turret.js
    - barricade.js
    - lightningTower.js

    - sprayCan.js
- Implemented source code for Packing Peanut Projectile, Blank Projectile, Lighting Taser Projectile, and Pepper Spray Projectile
    - peanut.js
    - blank.js
    - lightning.js
    - pepperSpray.js
- Developed animation for Packing Peanut Projectile Turret, Candy Cane Barricade, Lightning Taser Tower, and Pepper Spray Can Tower
- Developed animation for Packing Peanut Projectile, Blank Projectile, Lighting Taser Projectile, and Pepper Spray Projectile
- Developed enemy falling animation to be used for Candy Cane Barricade
- Implemented clamp to not let enemies get knocked off maps.

Ryan Fleming

- Developed enemy archetypes
- Developed tower archetypes
- Implemented source code to render canvas for game
    - index.html
- Developed game data to be used ie. health, gold, enemy tracker, level tracker, tower tracker, wave tracker.
- Implemented screen loader for game class
    - game.js
- Implemented object entity pool
- Implemented source code for ready, title, loss, and victory game states.
    - Intermediate.js
    - loss.js
    - play.js
    - title.js
    - win.js
- Implemented HUD to display game variables
    - HUD.js
- Implemented Enemy Target entity
- Implemented Strategic Point/Build Tile entity to allow player to build towers upon a clicking action
    - entities.js
- Implemented tower switching system using keyboard
- Implemented functions to change game state to next level or victory screen after all enemies defeated
- Implemented function to change game state to loss screen if health reaches 0
- Implemented Spawn Manager
    - enemy_manager.js
- Implemented incremental waves so that information can be relayed to player.

- Implemented collision function for enemies when they reach target
- Implemented state change functions for the game state to change if an enemy's health reaches zero and it is the last enemy
- Balanced enemy strengths and speeds to make game competitive
- Balanced tower damages, cooldowns and ranges to make game competitive

In our original plan we had anticipated using several other technologies in our final product. We wanted to use jaws for canvas rendering, tween for an animation engine, and howlerjs as a sound engine. However The way melonJS loads resources into the game made these all unnecessary as we could use melonJS to handle the rendering, animation and sound.

MelonJS also utilizes Tiled very well, especially for 2D games like this. We were able to set properties of objects in tmx files and MelonJS would pick them up when it would load the level. This significantly reduced the amount of raw code we needed to write for objects that would be instantiated when the level loads.

There was only one element that changed from our original project plan to our final product. We had originally wanted to make a pop up menu that would activate on clicking a build tile. Then the player could select the tile on the menu and the tower would be built. We ran into a lot of problems in testing several different forms of this method. Eventually we created a selection system based off of keyboard inputs for the purpose of testing towers in gameplay. We liked this gameplay element so much we decided to keep it in the final product instead of using a pop up menu. Instead we refined the key input selection system until we felt it was worthy to be in a final game. It allows for quicker input from the user and better game performance with less framerate dips.

We are very satisfied with our project and our decision to use MelonJS. After many Google searches, it seems like we are one of the first to make a tower defense game with this engine. This led to there not being a lot of examples to look at for inspiration, and no tower defense specific tutorials we could use to learn. This ended up forcing us to really learn every component and understand how the game engine works to really use it. It also led to us getting a lot of JavaScript practice in.