|  |
| --- |
| TEAM VULPECULA An HTML Tower Defense Game **Adrian Buenavista**  **Chris Darnell**  **Johannes Sunarto**  CS467  06.09.2017 |

# Link To Game:

**http://people.oregonstate.edu/~buenavia/CS467**

**OR**

**https://adrian858.itch.io/cs467-teamvulpecula**

**Gameplay**

* **If being played in Game Editor in Unity, your game resolution needs to be set at 960x660.**
* Very similar to plants vs zombies.
* Click on button, click on tile to place a tower
* Enemies drop items to collect:
  + Gold (for buying more towers)
  + Pill bottle (reduces nearby zombie health by 10%)
  + Record player: Stops zombies in same row from moving forward.
* There are three levels. Beating all three beats the game.
* Each level has a new enemy to deal with:
  + Level 1: Female zombie
  + Level 2: Male Zombie
  + Level 3: Jack O Lantern dude
* A level can be restarted if you are defeated (or you can decide to go to main menu).
* Your score is displayed on main menu screen (this is not persisted between sessions due to limitations of WebGL and time available to implement an alternative approach)

Unity - Setup

1. Download/install latest version of Unity Game Engine at <https://unity3d.com/> If Unity asks to make an account, a free one is sufficient for this project.

2.. Extract the submitted zip

3.. Open the project repository in Unity as a project

*To Evaluate*

NOTE: Resolution should be set to 960x660 when playing in the Unity Editor.

* Go to assets folder and select main menu scene
* Click Play button at the top of the Editor screen

# Client Requirements

● Your game must allow different individual "towers" (at least three different types) to be

constructed that serve some purpose on a 2D board or level.

Ninja, Cow girl, robot

● It will be single player only.

Yes

● There must be at least three different boards or levels.

3 levels

● Enemy units will enter the play field and attempt to achieve some objective: your towers

attempt to stop the enemy units from completing their objective (e.g., steal power cores,

damage a shield, cross the bridge and leave the level, etc.), in order to win (i.e. the player

must lose if the enemies succeed).

Just like the original plants vs zombies, the enemies will try to reach the left side of the board. If they do, then the game ends. Player can restart the level or go to main menu.

● There must be at least 10 "waves" of enemy units that enter the board or level.

There are at least 10 zombies in level 1, 20 in level 2, and 30 in level 3 (this number might change, to try to balance between towers and zombies)

● There must be at least three different types of enemy units.

Male zombie, female zombie, Jack o lantern

● Each defeated enemy should provide some sort of benefit to the player, for example gold or

energy used to purchase or power buildings or towers.

Gold, pill bottle (enemy health reduced by 10%), record player (enemy players on row stop moving forward)

● If your game has units attacking or otherwise affecting something, appropriate animations

showing projectiles, beams, healing, etc. must be shown to indicate what's happening.

Walking, shooting, attacking, dying

● An individual game, screen, or board must be winnable, with reasonable to serious

effort/skill, within 10 minutes. A new player should not be able to beat the game.

Tested several times to go thru game.

Technology:

● Must be hosted somewhere to test for grading purposes; OSU web server would be

Preferred.

Right now hosted on itch.io

● Must be playable using the Google Chrome web browser.

Tested in Chrome

● Will use game libraries, frameworks, and toolkits such as Melon JS, Tiled, Texture Packer,

Sigma.js, vis.js, D3, as well as more general purpose libraries such as Node.js, bootstrap,

React, Angular, and jQuery, if you like.

We used Unity (C#)

● It will not use Adobe Flash or Javascript-style "alert()" pop-up boxes to relate information to

the player. The user interface will display all information without resorting to asking the web

browser to pop-up a dialog.

No pop up of these kinds

● Data about the individual enemies and towers must be loaded from separate files on the web

server (one unit per file)

Data are loaded from files in the Resources folder (enemyConfig.txt, towerConfig.txt)

● The player must be able to save their game/score (TBD)

Ran into limitation on WebGL, difficult to save data locally on the machine because of restrictions. Can probably set up a web server (using REST for example) to get around this, but this is now difficult to pull off. It works when testing the saving/loading in Unity itself. You can load sample data in there right now (enter Adrian in name field and hit load, this is dev data and works in dev environment)