Plants vs Zombies Tower Defense

TP1(no change - same design)
TP2(no change - same design)
TP3:
Game does not end at a certain point, it just keeps going until the player dies. No search function for help tab. Zombies get stronger every 10 waves. Projectiles have acceleration. Instead of using files to store stats, they are stored as dictionary variables.

Project Description: Similarly to the game bloons tower defense, it will consist of towers (characters from the game plants vs zombies) and bloons (zombies). The game will create a random track for the zombies to follow and give the player a starting amount of money. The player can use the given money to purchase plants as well as upgrade the plants to make them stronger. The game will consist of many stages, with each stage increasing in difficulty. The player will also gain more money for killing zombies as well as passing stages. The game ends when the player has successfully made it through each stage without dying.

Similar Projects: There are many other projects similar to mine that have been done in the past. Similarly to what I will be doing, previous projects have all randomly generated a track and randomly generated waves. They also use some sort of currency system in order to purchase towers that are used to defend against the waves. However, unlike other projects, mine will have an upgrade system in which one can upgrade their existing towers into stronger towers. My game will also be different in the sense that it will have a search function within the help tab that allows users to search up plants/zombies based on their attributes or name and get to a page that displays all relevant information regarding that plant/zombie.

Structural Plan:
- Files
  - File containing all of the towers and their corresponding stats, etc.
  - File containing possible combinations of all zombies for each wave (or some algorithm for generation a possible combination)
- Classes
  - Create a class for each type of tower
    - Will show stats, appearance, available upgrades, etc.
  - Create a class for each type of zombie
    - Will include stats, etc.
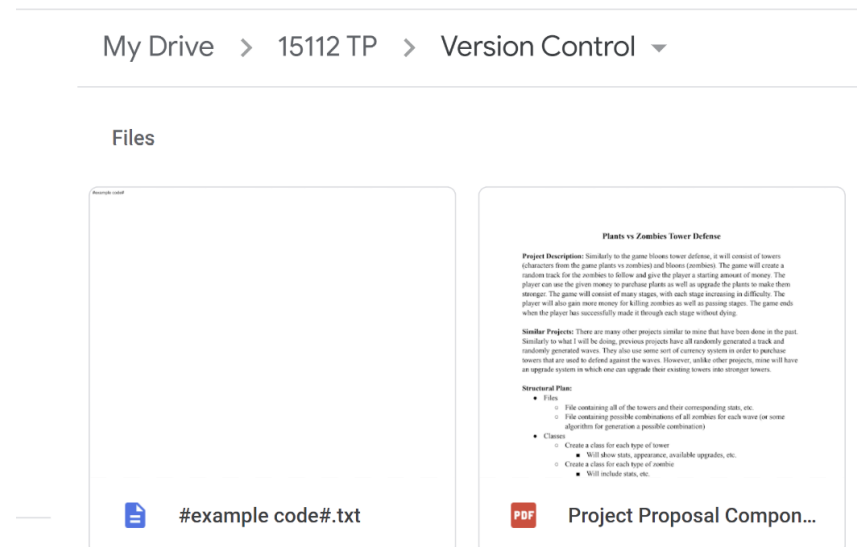 - Create a class for available money

Algorithmic Plan:
- The most algorithmically difficult part of the game will be creating an algorithm for creating randomly generated waves. I will likely use a ranking system that ranks the "power" of each zombie, sets a certain total power for each wave, and makes sure that the combined power of each zombie adds up to the total power for each wave. Another difficult part of the game will be implementing a search function that allows the user to search up a certain plant/zombie or a certain stat/type and have the search function display the closest matches. The function will have to be able to sort through the set of strings and also display the closest matching strings so I will also have to find an algorithm that finds what strings are closest to what the user has typed.

Timeline Plan:
- By TP1
  - Have all the files with the stats, graphics, etc for all plants/zombies as well as code for their classes completed
  - Finish the home screen for the game
- By TP2
  - Code in graphics for attacks as well as movements for each zombie/plant
  - Create algorithms for creating map
  - Create functions that check where it's legal for something to be placed, etc.
  - have a basic functioning game (just one wave or something)
- By TP3
  - Implement upgrade system for plants
  - Implement currency system (changes based on zombies killed/waves passed)
  - Implement wave algorithm that continues to generate waves

Version Control Plan: Each time a significant part of the project is completed (some file, block, etc) is completed, I will store it within a google drive folder.

My Drive > 15112 TP > Version Control

Files

#example code#.txt

Project Proposal Compon...

Module List: As of now, None