15-112 TP1 Design Proposal

Maxwell Chien

November 2021

Project proposal

Project Description

The (tentative) name will be cuddly basement zombies since I'm having trouble naming my project and I need to start coding. This project is a procedurally generated maze dungeon trawler game with mobs scattered throughout the maze. The final idea includes items that can be found and utilized as well as a fightback mechanism and different mobs to fight. There will also be a point/money system to incentivize optimal play.

Competitive Analysis

This project is similar to many rogue-like dungeon games due to the exploring and fighting mechanics as well as the procedurally generated maps that increase in difficulty. Additionally, the map generation and mob spawning locations draw a similarities to the Pokemon Mystery Dungeon franchise. However, like all roguelike games, there are differences in the specific combat mechanisms implemented as well as how the mobs will be programmed. For combat mechanisms, the ideas are still tentative, but as of 11/18, the boss fights are planned to be some combination of osu + bullet hell while the regular fights are more of a kiting/blocking (depending on the mob) while shooting something (circle for now for easier coding) to do damage.

Structural Plan

Currently, the plan is to have a main file to run all of the code, therefore every module somehow gets imported into the module whether directly or indirectly. There will be a mob module to create the mob class and all its properties, a map module to generate the map and return information from the map, like spawning locations, an astar module for the pathfinding algorithm and getting the next location, a player module for the player class, a rectangle module for a rectangle class for easier drawing.

The plan is to include fight and travel modules such that the main file is cleaner (doesn't need to include many messy lines of code in the draw file).

Algorithmic Plan

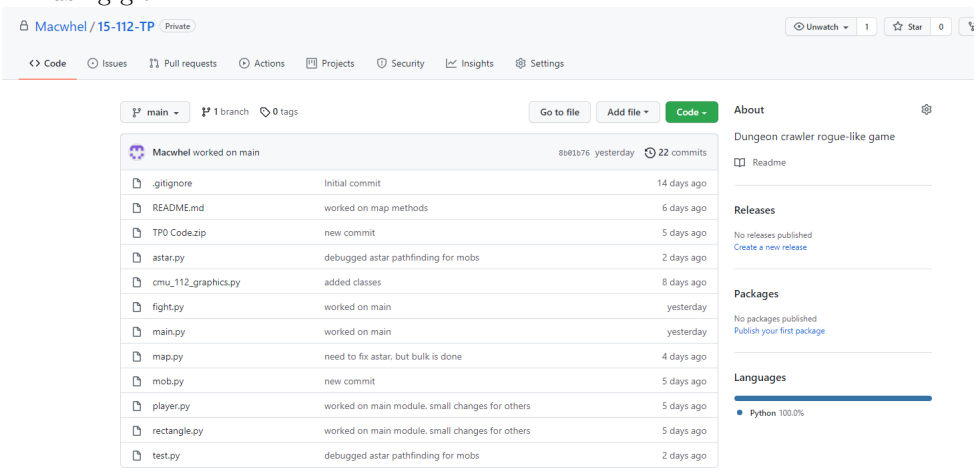
The algrorithms that are for sure going to be implemented are A star and random walker. A star is used for mob to player pathfinding. The random walker algorithm is for the mob generation as it generates imperfect mazes, which makes it better for games with mobs in them. Later, the plan is to implement Kruskals, Prims, or Growing Tree algorithm to generate perfect mazes, but only for special levels, thus it's lower on the priority. The approach for each of them was to heavily research the pseudocode and visual representation of each of them, and then break up the algorithm and test it in a separate, controlled file.

Timeline Plan

- 11/15: Implemented A*, Implemented random walker 11/18: Basic mob fight implementation 11/20: Begin boss fight implementation 11/22: Finish up boss fight. Polish it up. Thanksgiving break: Polish up mechanics (moving speed, mob fight shooting/moving, work around the annoying refresh rate) Add special levels (20 dfs mobs in one level, perfect maze generation, 2 wide corridors) - 12/1: Research login/save/pause mechanics and consider implementation

Version Control Plan

I'm using git.



Module List

Currently, the only module that I plan to import is queue for it's priority queue, but it's very minor since it's just for efficiency. Later, more modules related to

saving a game or having log in info may be implemented, but research for that has not started because it is low on the priority list.

TP3 Update

The name will now be "Basement Exploration." The basic mobs are squares now. Structurally, most is the same, but it's a bit rougher. I decided to keep all the graphics/tkinter functions in one file, all the hostile mob classes in one file, and all the map generation/related functions in one file, thus I don't exactly have separate files for every single function, and I think this approach is cleaner. Algorithmically, I ended up only implementing Kruskals as well as a derivative which utilizes Kruskals but generates a weaving maze. I didn't actually utilize any pseudocode and only used figures and demos of the maze generations. I also implemented a shop for the game, and I didn't end up adding any special levels due to time constraints, as most of my time was actually spent tweaking the parameters for the fight.