

## Project Proposal

### Project Description

Name: Dungeon Explorer

Description: A dungeon crawler-like game in which a player tries to collect all the hidden treasure scattered across a maze while avoiding/defeating enemies.

### Competitive Analysis

My game is relatively similar to the game of Pacman. In Pacman, the player navigates a maze-like map while collecting dots and power pellets while avoiding the ghosts that chase them around the map. If the player collects all the dots, the level is passed. If Pacman loses too many lives from running into ghosts, the player loses the game. In my game, the game is won if all the treasure is collected and is restarted if the player loses too many lives from running into enemies. Occasionally fruits (e.g. cherries) show up on the map and can be collected for extra points. My game does not have a points feature. Pacman is controlled with arrow keys, and has no way of defeating ghosts unless a power pellet is eaten. The players in my game will use their mouse to control the character, have the ability to fire “bullets” at enemies to defeat them, and can blink a limited distance away from their current position. Pacman’s ghosts use artificial intelligence to determine how to chase the player. The enemies in my game use A\* pathfinding to determine the best path to reach the player.

### Structural Plan

I will most likely have multiple files for my project; for the main executable code, the player/enemy classes, the A\* pathfinding code, and map data. I will also include a .png file of my spritesheet so it can be accessed by my executable file. In the main executable file, I will have an `appStarted` function to store/initialize variables, functions for grid/coordinate organization, `redrawAll` to create the sprites, `mousePressed` and `mouseMoved` for player movement, `keyPressed` for player abilities (shooting bullets, blinking), and `timerFired` for the actual running of the game. In the A\* code file, I will include all the functions needed for pathfinding, such the equation for Manhattan distance and a function to get the path in terms of coordinates.

### Algorithmic Plan

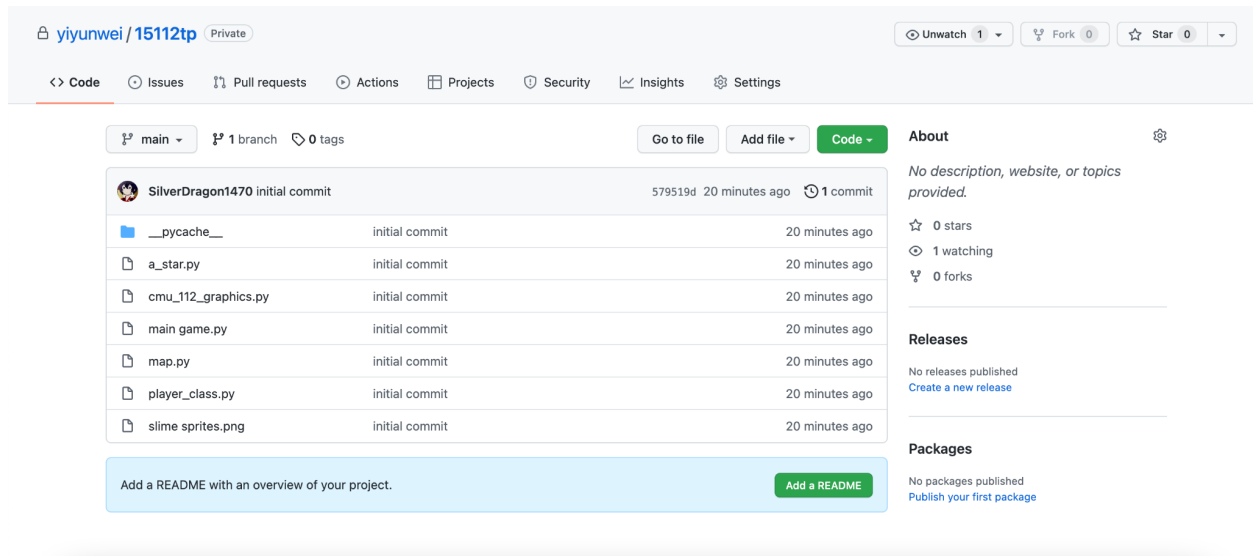
The hardest part of my project will probably be implementing the pathfinding algorithm. I will research online how the A\* pathfinding is supposed to work, and look at examples of pseudocode to understand how it should work in the context of my own project. Then, I will code the enemies to move according to the pathfinding algorithm only when the player is able to see them in their window.

## Timeline Plan

I will have the enemy pathfinding/movement done by the tp1 deadline. For the tp2 deadline, I will make it so that the game is actually playable (enemies scattered across the map, player movement that automatically shifts window placement, treasure placement, etc.). For the tp3 deadline, I will add features for the user experience (pause button, background music, score keeping, level select, etc.).

## Version Control Plan

I will use Github to save backups of my code.



## Module List

I did not use any additional modules.

## TP3 Update

Level select was not included in the final product, as I didn't have time to look into maze generation to create new levels. I also made it so that the view window has to be manually adjusted with arrow keys instead of being automatically moved. I added a splash screen mode that is present at the beginning of the game. For the pause menu, I made it so that players can pause/unpause the music, as well as display how much total treasure there is to collect. Otherwise, everything else is as planned.