blackEverythingwhite
Aryan Bhatt, James Cruse-Mulhall, Eugene Thomas
APCS2 pd4

Seeker

Basic Description

- Imagine that you are placed inside of a labyrinth, unsure of how to escape. At this point, you question whether or not you will make it out alive. This fear persists and exacerbates until four lights are shone into your line of sight. You can now see the squares to the front, left, right, and back of you. You must seek your way out. You are now the seeker.

Basic Rules

- You will be prompted to select a level of difficulty. please choose whichever level suits you and your seeking abilities the best.
- Once you are placed onto the world, you will be granted with three lives and be able to see the four orthogonal squares from your character.
- Select which direction you would like the advance in.
    - If there is an obstacle at that square, you lose a life, are respawned at the same place, and forced to restart your journey with the same exact board setup.
    - If this square is free of obstacles, the path you have already trekked will be shown and you will be prompted to pick from the other three orthogonal squares you have. **Note: There's no going back on your journey out.**

    - If this square reveals a heart, this means you have an added life on your journey and will similarly be asked to choose from the three remaining orthogonal squares.
- The game is over when…
    - You run out of lives, prompting a very taunting "YOU LOSE! Wanna play again?"
    - You reach the exit of the labyrinth. **Note: This is a special kind of Labyrinth that does not always have the exit at the end.** Once you reach this exit you will be praised as the greatest person of all time and be prompted to try your luck again (potentially at a harder round).

Utilization of APCS2 Information

- The most heavy algorithm in our code will be setting up the board itself as we must gauge:
    - If there is a possible solution (a maze-solving code that we might add in as a part of our prompt to play again).
    - If the solution is too easy (an algorithm that sees how many moves it would take the character to optimally win the game, if it is below that number, we must create a new board).
    - If all of the hearts and obstacles are accessible, not obstructed by other obstacles.
    - The construction of these three similar algorithms will utilize a strategy that we utilized throughout the second semester course: backtracking.

- In addition, we will be using a Priority Queue for the various setups of the board, that are ranked by increased difficulty.
    - For example, the simplest maze (which will be noted by a variable for difficulty), will be the first line in the queue, which means it will be summoned first, while the most difficult maze will be last in, therefore it will be last out as well.