**Project/Code Description**

My project was an attempt to improve basic enemy avoidance games, similar to those found on ROBLOX, by improving enemy pathfinding. This consisted of plotting paths around static obstacles (walls) and turning away from nearby enemies in a real-time, non-grid environment.

For the pathfinding, I created a heuristic-based version of depth-first search to create a path, then trimmed it down with a path-trimming algorithm (both of these are found in the Graph class). The resulting paths aren’t always optimal, but they’re reasonably short.

To make enemies turn away from each other, I had enemies find the nearest enemy (taking the size of the other enemy into consideration) and use vector calculations (included in a Vector2 class I made) to determine which direction to turn in to avoid it. This method is under the Enemy class (turnAwayFromEnemies).

**Gameplay**

The gameplay is very simple. Before starting the game, the player can place/remove walls to create a map. The player can specify the walls’ dimensions. Also, a pre-created map can be loaded by pressing the appropriate button in the menu.

Once the game is started, enemies spawn at random intervals, though the player can spawn more enemies manually by pressing a button. The number of active enemies is capped at 10 to prevent lag from getting out of hand. After killing enough enemies, the game level increases; for the first two levels, only one type of enemy appears (Explosive in level 1, Zombie in level 2). From then on, both types can appear together.

Killing enemies gives the player money, which can be spent to upgrade the player’s weapon. Also, a stats menu can be opened to see the current level, the number of enemies killed, the progress towards the next level, and information about the player’s weapon.

**User Interface**

The game starts with a help screen open, which explains how everything works. A menu on the right takes up about a fifth of the screen. The top half contains buttons to purchase weapon upgrades and has a display of the player’s health and money. The bottom half contains various buttons for starting, resetting, and pausing the game, viewing the player’s stats, editing walls, and opening the help screen. The buttons disable themselves when appropriate. For instance, opening the help screen pauses the game and prevents you from unpausing until you exit the screen, and starting the game disables your ability to edit walls.