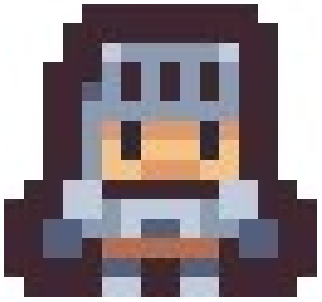
CS108 Maze Game



Yash Sabale
23B1043
Prepared as SSL Project 2023

Contents

1 Modules

In the making of this project the following modules were used:

- Pygame
- Random

1.1 Pygame

Pygame[1] is a set of Python modules designed for writing video games. Pygame allows you to create fully featured games and multimedia programs in the python language.
I used Pygame to render process the tilemap to load in the sprites and the rendering of the images to the screen.

1.2 Random

This module implements pseudo-random number generation in python.
I used it to generate random mazes and also handle spawning of background decoration, enemies and chests.

2 Directory Structure

The Project folder consists of a total of 11 python files with a folder named assets to store all the images and font files for the game

- **game.py:** The main file which runs all GUI elements and also the game
- **mazehandling.py:** This file handles the random maze generation
- **utilityfunctions.py:** This file has some functions like leaderboard handling and maze rendering
- **Player.py:** This file has the player class which handles collision detection and player controls
- **Graves.py:** This file has the Grave class whcih manages ghost spawning with graves
- **Ghost.py:** This file has the ghost class which handles all things related to the ghost enemy
- **EvilWizard.py:** This file has the evilwizard class which handles all things related to the wizard enemy.
- **fireball.py:** This file has the fireball class which handles the fireball which the wizard shoots
- **Chests.py:** This file has the chest class whcih handles the chests which can be found scattered around in the maze
- **Collectibles.py:** This file has all the items which can be found inside a chest as classes

- **Settings.py:** This file has all the game customizations which can be done at the start of a level

3 Running Instructions

3.1 Starting the Game

To run the game just run game.py using python3. Once you run it you will see a home screen like this.

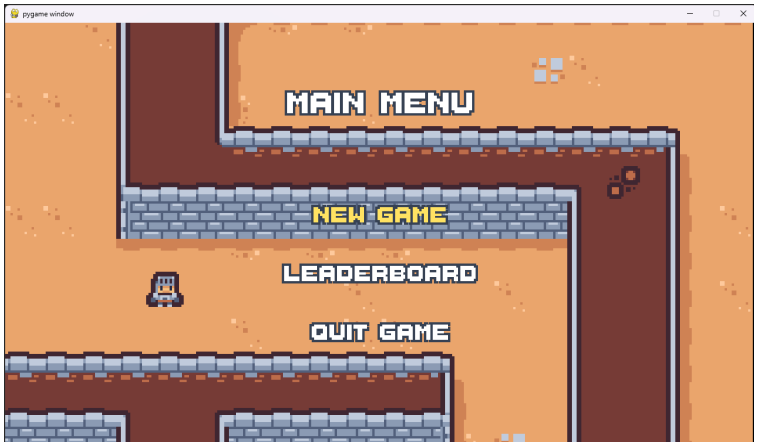


Figure 1: Home Screen

Then you can use WASD or the arrow keys to move to the option you want and then press Enter to select the option.

You then can select the game difficulty from one of these

- Normal
- Hard
- Insane

the specifics of the game difficulties can be found in the appendix

You can control the knight using WASD or the arrow keys to move the knight and use the space bar to use the sword for attack.

4 Basic Implementation

4.1 Maze Generation

The Game uses recursive backtracking as the means to generate the maze [2]. I chose this algorithm because it allows creation of long winding passages which could be used for combat.

```
[[0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1],
 [1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1],
 [0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1],
 [0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1],
 [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1],
 [0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1],
 [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
 [0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1],
 [0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1],
 [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1],
 [0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1],
 [0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1],
 [0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1],
 [0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1],
 [0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1],
 [0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1],
 [0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1],
 [0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1],
 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
```

Figure 2: Sample maze generated using the algorithm

The maze is generated as a 2D array of zeroes and ones where ones represent the walls and the zeroes represent the playable area.

Then the maze is expanded to three times its size to make both the walls and the paths 3 wide as the wall in the tileset i was using were 3 wide.

4.2 Rendering the Maze

I used *Tiny Dungeon By Kenny [3]* for the tileset for this game and used *Thaleah pixel font [4]* as the font for the game

Figure 3: Tileset used in this game

The game is rendered on two layers

- **Foreground Elements:**

  The enemies and the player are rendered on this layer

- **Background Elements:**

  The walls and the ground are rendered onto this layer, with there being some random elements like pillars on the wall and pebbles on the ground.

And then the rendered layers are rendered onto the actual screen with an offset to emulate the screen scroll effect.

4.3 Collision Detection

Before updating the player's position we map the player's expected position on the next frame to the map and check whether the player is intersecting with the walls and if he is

then we don't move the player on that frame.

5 Game Mechanics

You can move around and attack using the sword by pressing the space bar. The aim of the player is to reach the exit at the bottom right corner while fighting enemies and collecting boosts.

Figure 4: The exit of the maze

Once you exit the maze the game then asks for your name to update the leaderboard.

5.1 Enemies

There are 2 types of enemies in the game:

- **Ghost:**

  They spawn from graves and follow the player and deal damage on contact.

Figure 5: Ghost

When the player is looking in the direction of the ghost moves around in a circle around the player and when the player has his back towards the ghost they approach the player in a straight line.

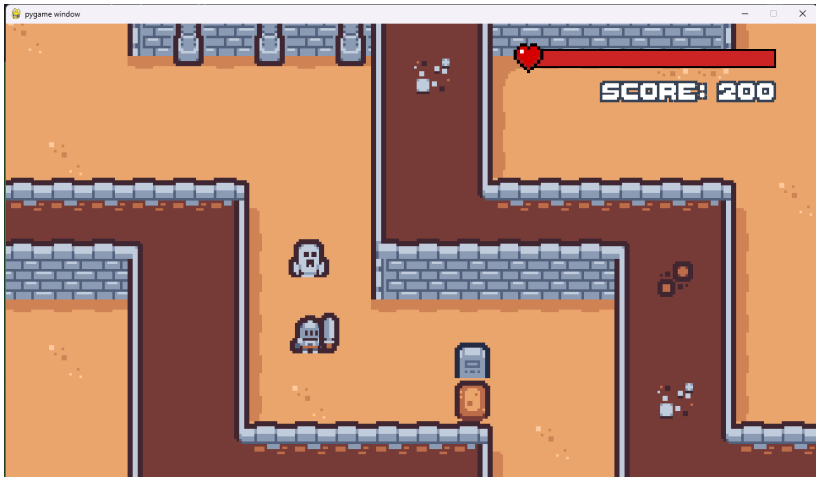
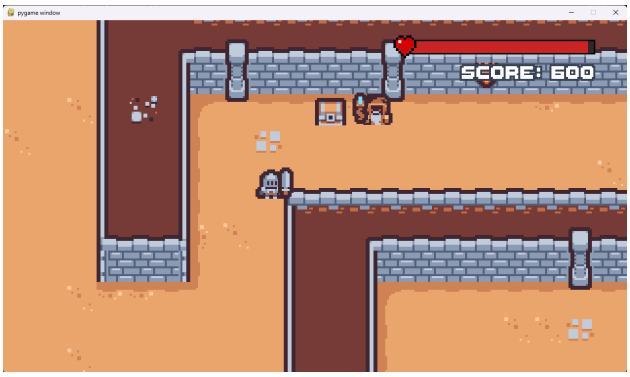


Figure 6: Ghost following the player

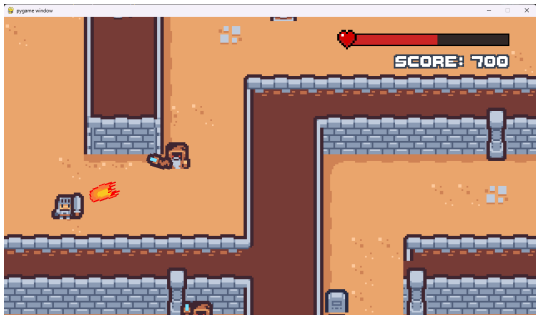
- **Wizards** They can spawn near chests and shoot fireballs which can damage the player.

Figure 7: wizard

The wizard uses BFS to find a path towards the player and stop 5 blocks before reaching the player and then starts to shoot fireballs directly towards the player.



(a) Wizard Guarding a chest



(b) Wizard Shooting a fireball

Figure 8: Wizard Mechanics

5.2 Collectibles

In the game you can open chests by attacking it. The chest has chance to spawn the following items.

- **Health Potion** it heals your hp by a set amount decided by the game difficulty

Figure 9: Health Potion

- **Sword** Its a better sword which boosts attack damage of the player.

Figure 10: Golden Sword

6 The Project Journey

Learning how to manage a project along with planning the file structure and how the functions and classes interact with one another was a great experience. Also solving technical issues with optimization as pygame couldn't handle loading the whole maze at once was extremely interesting.

Also as I couldn't implement all the little details I wanted to due to time constraints leaves a lot for future expansion to this project.

7 Appendix

7.1 Difficulty Settings

Settings	Normal	Hard	Insane
Maze Dimension	(20,20)	(30,30)	(30,30)
Ghost HP	1	2	4
Ghost Damage	0.5	1	3
Grave Spawn Chance	1	2	1
Max Grave spawns	2	3	5
Grave spawn time	800	600	600
Wizard spawn chance	40	70	70
Fireball Speed	400	500	600
Sword drop rate	10	20	30

References

[1] Pygame Documentation. https://www.pygame.org/docs/.

[2] *Recursive Backtracking*. https://en.wikipedia.org/wiki/Maze_generation_algorithm.

[3] Kenny assets. https://kenney-assets.itch.io/tiny-dungeon.

[4] Thaleah fat. https://tinyworlds.itch.io/free-pixel-font-thaleah.