PES UNIVERSITY – Python for Computational Problem Solving (UE25CS151A)

Project Report

Jackfruit Mini Project – Maze Runner

Name	SRN
Nikhil Kandur Srinath	PES1UG25EC170
Ninad Pramoda Hebbar	PES1UG25CS342
Patel Arya Ajaykumar	PES1UG25CS361
Praful Kumar CM	PES1UG25CS366

Problem Statement

- To create a game using python.
- Traditional maze games are usually single-player.
- The objective of this project is to design and implement a two-player maze game where each player independently navigates a randomly generated maze and competes to reach the destination first.
- To enrich the competitiveness a scoring system as well as a timer is required to be implemented.

Approach / Data Structures Used

The project is built using Python and the Pygame library for graphical rendering and event handling.

Maze Generation Algorithm: We utilized the Recursive Backtracker algorithm. This ensures that the generated maze is a proper maze which ensures the solvability of the maze.

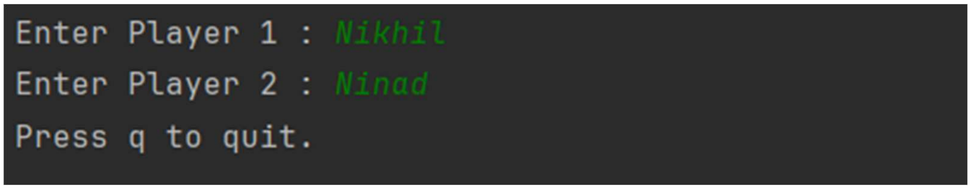
Data Structures Used:

1. List of Dictionaries: A list is used to contain every single cell within the grid of the maze. And each one of these cells are represented by a dictionary containing values like the walls and position.
2. Tuples**:** Used for coordinate pairs (x, y).
3. Stack: A stack is used during the maze creation for the purpose of backtracking.

The game runs at 60 frames per second and thus registers keystrokes and applies changes 60 times a second hence allowing a smooth game.

Inputs

The program requires initial input via the console for player names, followed by real-time keyboard input for movement.

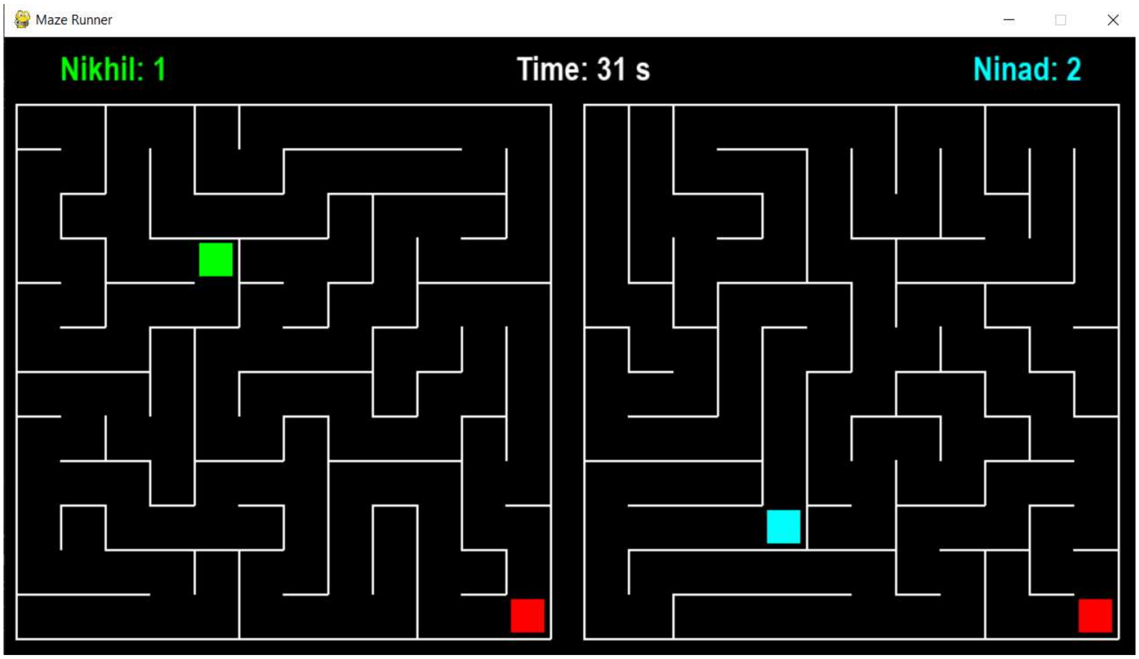


```
Enter Player 1 : Nikhil
Enter Player 2 : Ninad
Press q to quit.
```

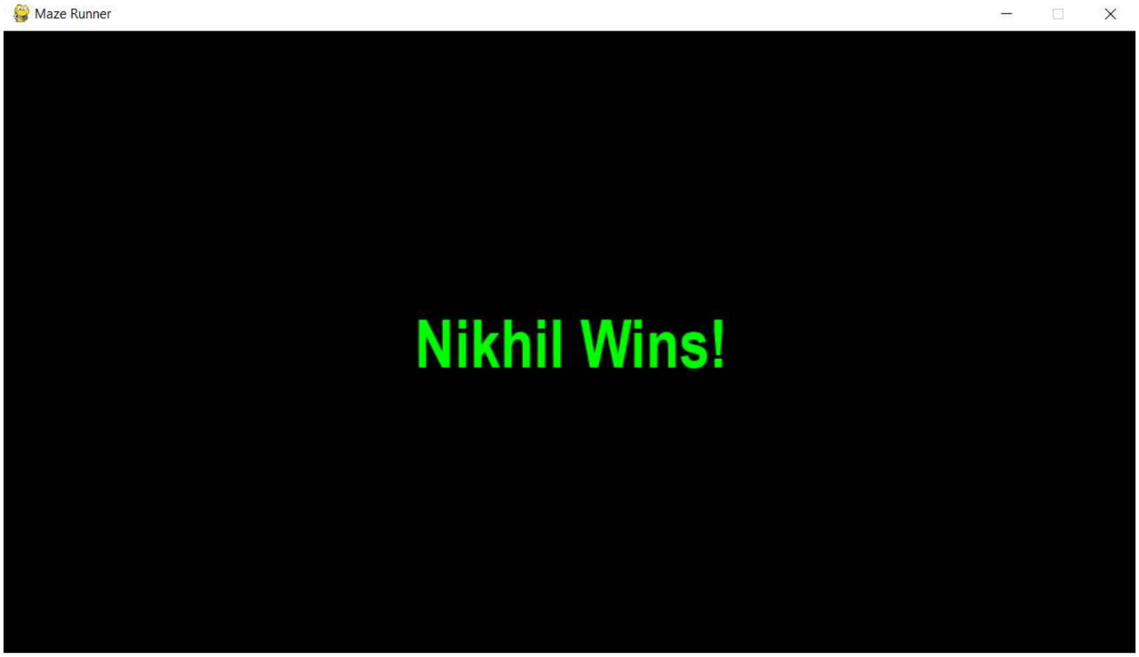
User Inputs

Output

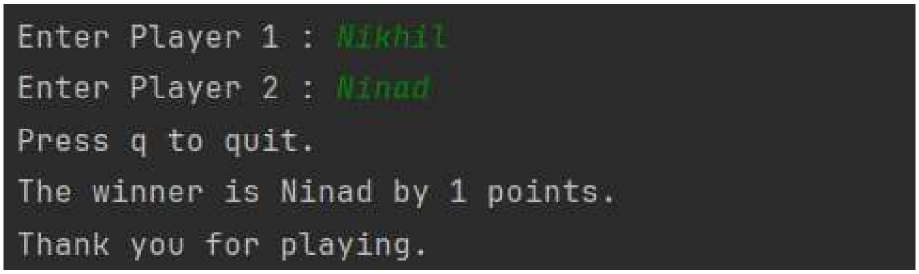
The output is a display consisting of two differnet mazes, a timer, and the current score. Upon completion, the winner is announced on the display and a summary is printed to the console.



Game GUI



Winner Display



```
Enter Player 1 : Nikhil
Enter Player 2 : Ninad
Press q to quit.
The winner is Ninad by 1 points.
Thank you for playing.
```

Final Game Output

Challenges Faced

- Maze Generation: Implementing the backtracker algorithm correctly to ensure the creation of a solvable and truly random maze.
- It was difficult to contain the grid of cells into a list as the maze grid is a 2-dimensional structure which needs to be stored in a 1-dimensional list.

Scope for Improvement

- The GUI can be improved with input within the Pygame interface.
- More levels can be added to increase the difficulty by increasing size of the maze.

GitHub Link

https://github.com/Nikhil-Kandur/Maze-Runner