Maze Runner

Group Members

- Piyush Pranav (201701109)
- Ranjan Kumar (201701123)
- Rohit Kumar Singh (201701127)
- Vaibhav Maheshwari (201701169)
- Sakir Shah (201711237)
- Shubham Raj Gupta (201711238)

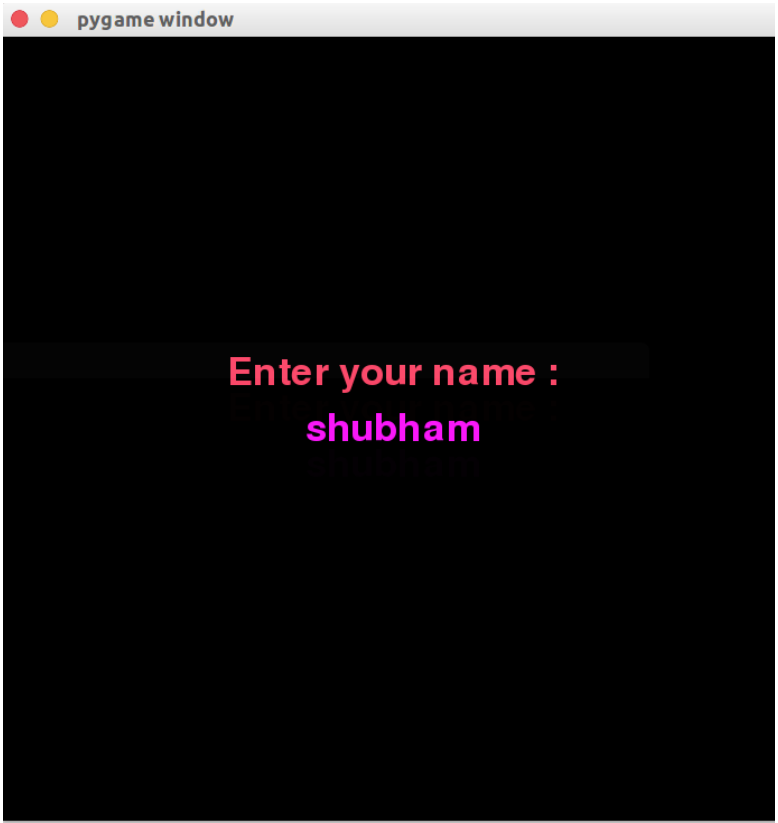
Overview

A 2-D game where u have to run to escape the maze. Beware of the monster on the way, if they sense your presence, will follow you until you are dead. Keep collecting treasures on the way to get good points. It's a combination of Pac-Man and simple maze solver, with improvements and impressive gameplay.

Gameplay

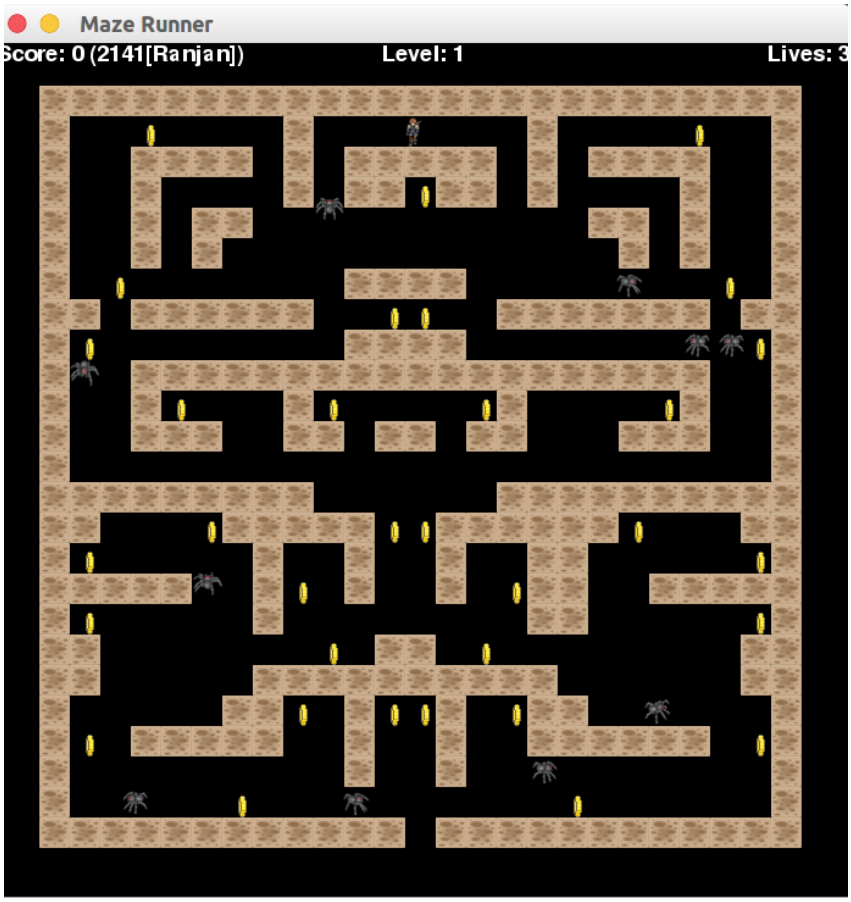
In this game, firstly the plyer has to enter his/her name in order to start the game and keep records. There are 5 levels in the game which needs to be completed to win the game which are interconnected. If the player finishes the first levels the next level automatically starts. The difficulty level rises with the levels. The player can fire bullets also which has a range of 200 units that can kill enemy and save themselves.

1.Entering name when the game starts

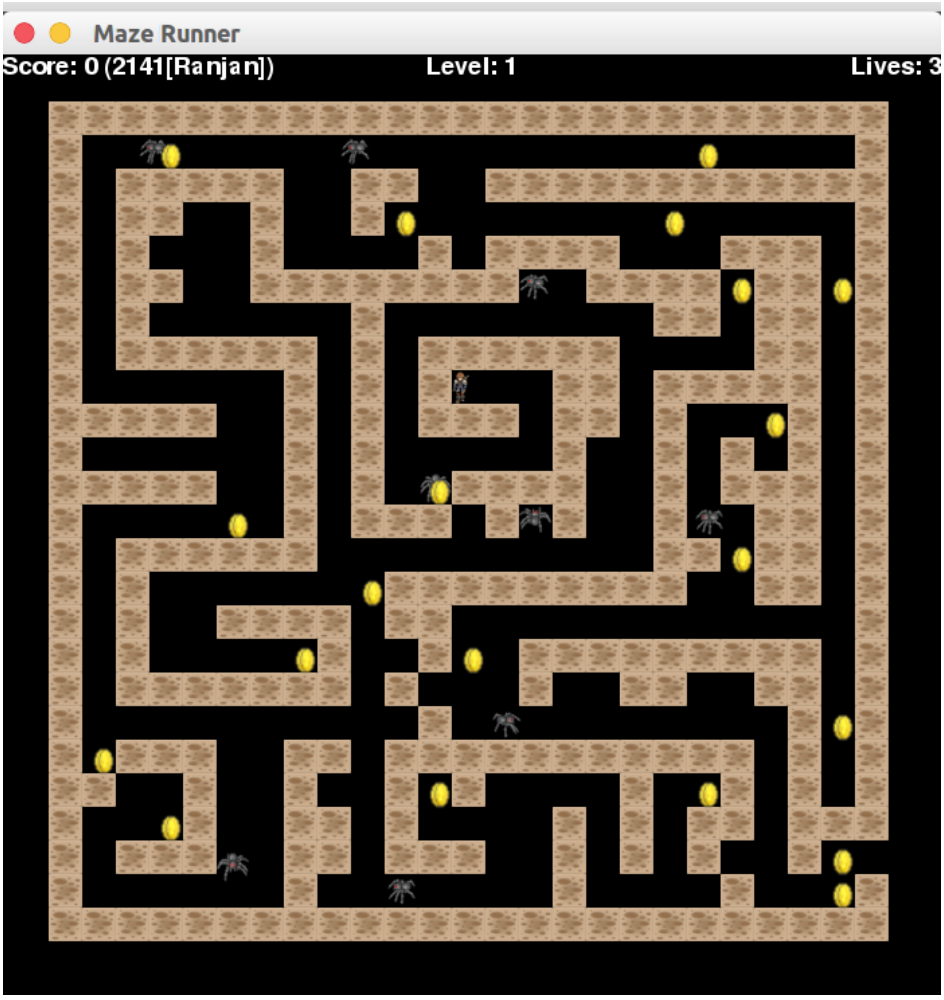


Maze Runner

Levels

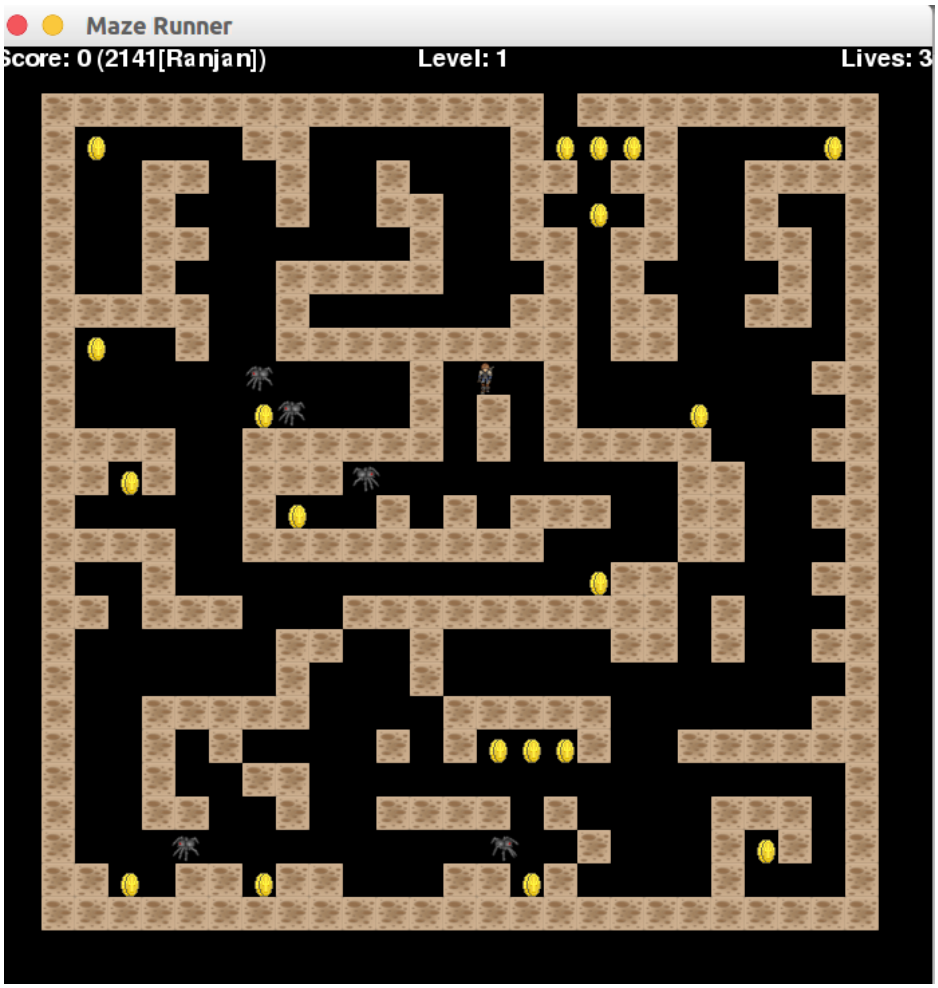


Maze Runner


Maze Runner
Score: 0 (2141[Ranjan])
Level: 1
Lives: 3


Maze Runner
Score: 0 (2141[Ranjan])
Level: 1
Lives: 3

Maze Runner



Milestones

Stage1:

First we researched about the idea about the methods of making games using python and pygames and how can we add graphics animations and sounds to a particular part of the game. We found games and read their codes to gain some knowledge about game development.

Learning:How to code for game development.

Stage 2:

We wrote some codes for window sizes , fonts and basic game play, like the movement of our player. At that time there was no enemy and treasures in the game. We inserted rectangular color boxes in this phase for walls, player .There was only one level which we were working on. We divide our parts for furthur working.

Learning: Basic controls using pygame

Maze Runner

Stage 3:

We added coins and enemies in the game and set their working in the game. We added the scoreboard to calculate overall score by collecting coins. We also set 3 lives as total to the player as when collision with the enemies the player shouldn't die in the first chance.To make the game a bit difficult,we are reducing the score by one unit per second.So, player needs to finish the game as fast as possible in order to score high.

Learning: Adding extra periferals.

Stage 4:

We added images in parts for movement of player and enemies. The images of coins are set to look like moving in circular order through multiple different images which were fed in a list and we kept changing them within certain interval to look like moving. Same thing we did for palyer and enemies, We fed 12 different images for each direction fot the movement of player and enemies and changed them within intervals to look like actual movements.

Learning: Adding images and moving them.

Stage 5:

We added five different levels by hardcoding and selfdesigning blocks like mazes, we added treasures,enemies at differnt places.

Learning:

Stage 6:

We finalized our game by adding several sound effects for next level, game over and for regular gameplay, and connected all levels together.

Learning: Working with media.

Learning Outcomes

1. Designing basic gameplay.
2. Adding graphics and sound effects to make it more interactive.
3. Adding multiple levels.
4. Working with pygame.
5. Deep learning of class and OOPS concept.
6. Clean coding.