

Ashoka Education Foundation
ASHOKA CENTER FOR BUSINESS AND COMPUTER STUDIES

Project Synopsis

Project Title:

Shooting Game using Python.

Developer: Noorsaba Shaikh.

Objective:

The objective of this project is to develop a 2D shooting game using the Python Pygame library. The game allows the player to control a character, navigate through levels, avoid obstacles, shoot enemies, and reach the exit to complete the level. The game incorporates basic game mechanics such as player movement, shooting, AI (enemy) behaviour, and collision detection.

Technologies Used:

- **Programming Language:** Python
- **Game Library:** Pygame
- **Data Management:** CSV (for level design)
- **Sound Effects:** Integrated using Pygame's sound functionalities
- **Graphics:** Sprites and images for the game characters and environment

Features:

1. Player Controls:

- The player can move left, right, jump, and shoot using keyboard inputs.
- Jump mechanics allow the player to avoid obstacles and enemies.
- The player's movement is restricted by level boundaries and platforms.

2. Shooting Mechanism:

- The player can shoot bullets to defeat enemies.
- A shooting cooldown prevents continuous firing, and the player has a limited amount of ammo.
- Shooting sound effects enhance the gameplay experience.

3. Enemy AI:

- Enemies (AI) patrol the level and detect the player when in range.
- AI can shoot back at the player when they are close enough.
- Random idling behaviour makes AI movements less predictable.

4. Level Design and Progression:

- Levels are designed using CSV files that contain the layout for obstacles, platforms, and enemies.
- The player must navigate through each level, avoid hazards (e.g., water), and reach the exit to complete the level.
- Collisions with certain objects (e.g., falling off the screen or hitting water) result in health reduction and game-over scenarios.

5. Collision Detection:

- The game includes collision handling for the player, AI, and objects such as platforms and bullets.
- Health is reduced when the player is hit by enemies or environmental hazards.

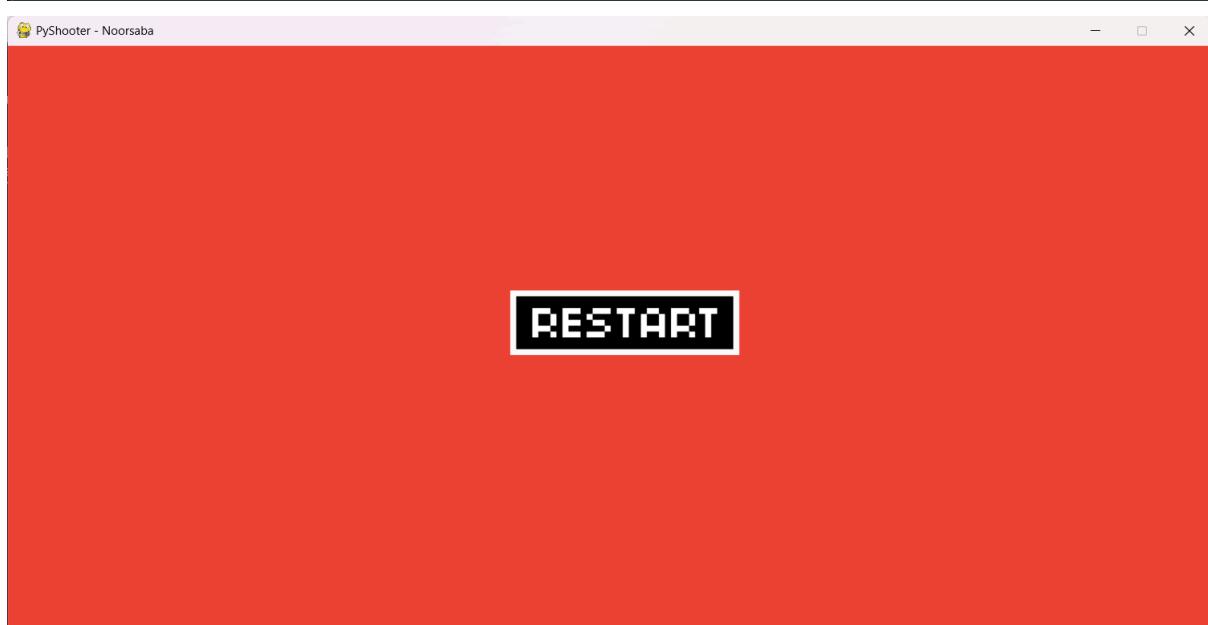
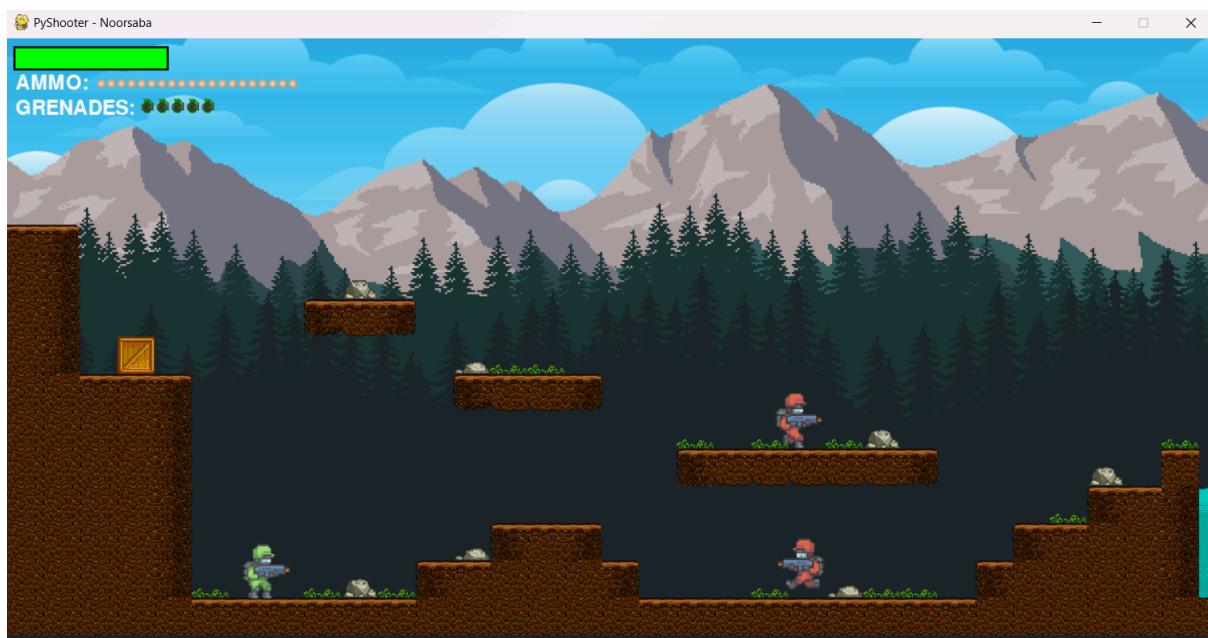
6. Game Over and Restart:

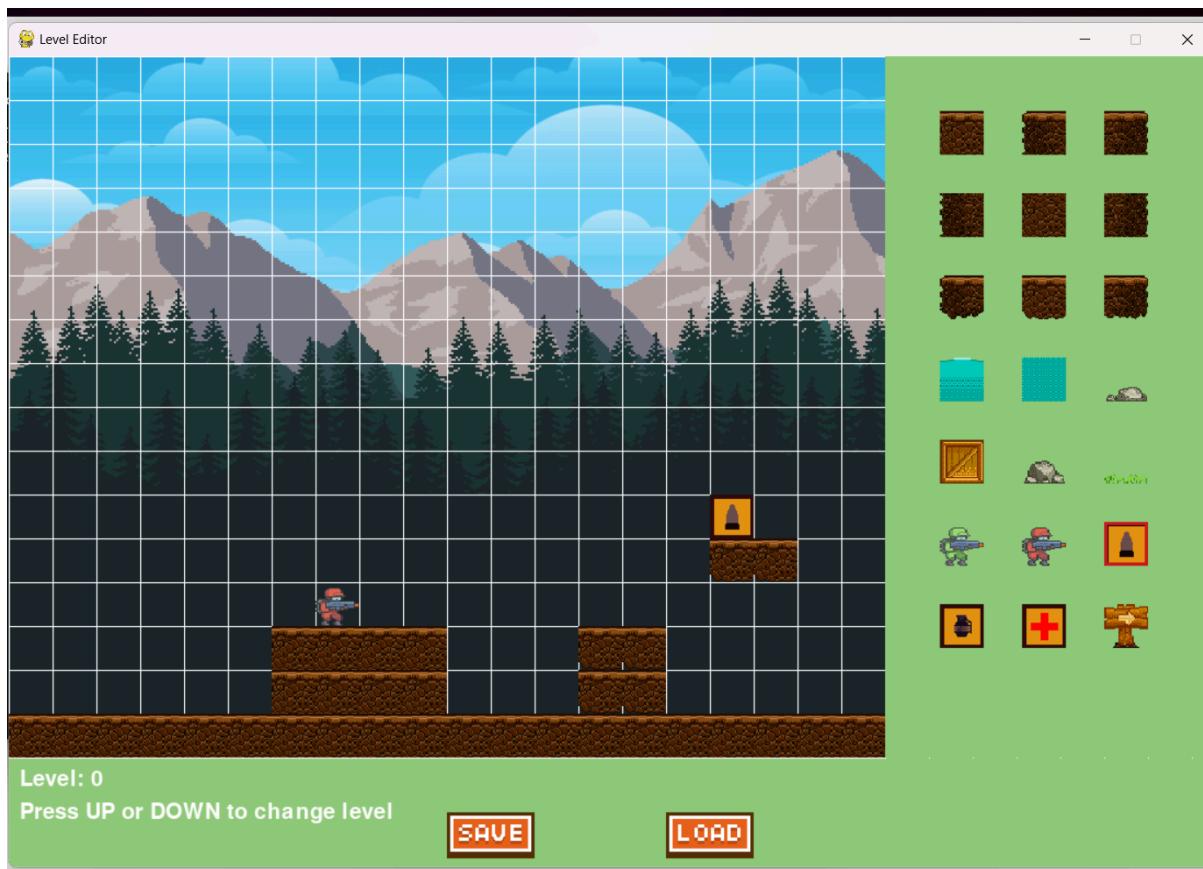
- The game triggers a fade-out effect when the player loses all health, and a restart option is provided.
- Upon restart, the player's stats are reset, and the current level is reloaded.

7. Sound Integration:

- Background music and sound effects are included for shooting, jumping, and other actions.







Future Enhancements:

- **Additional Levels:** More levels can be added with increasing difficulty.
- **Power-ups:** Introducing items like health packs or ammo refills.
- **Multiplayer Mode:** Implementing a two-player option.
- **Leaderboards:** Adding score tracking and a leaderboard to increase competition.

Conclusion: This shooting game project demonstrates the application of object-oriented programming concepts, game design principles, and the use of Python's Pygame library. The project provides an engaging experience while showcasing core gameplay elements such as player movement, AI interaction, collision detection, and level progression. The modularity of the game makes it highly customizable and scalable for future developments.