Recreating a Shooter Game Inspired by "DOOM" Using Python

Project By: Dama Charan-[damacharan1234@gmail.com]
P. Bhanu Reddy-[pullurbhanuprakash620478@gmail.com]
P. Lakshmi Deepak-[putturudeepak2005@gmail.com]

S.V. College of Engineering, Tirupati

March 19, 2025

Abstract

This paper presents the design and development of a Python-based shooter game inspired by *DOOM*. The game utilizes Pygame to handle graphics, player movement, and shooting mechanics, while AI-controlled enemies enhance the challenge. The study explores game physics, sprite animations, and collision detection within a 2D environment. Performance optimizations and future improvements, such as procedural level generation and multiplayer functionality, are also discussed. The study demonstrates the feasibility of developing an engaging shooter game in Python and highlights key technical considerations.

1 Introduction

The classic 1993 game *DOOM* revolutionized the first-person shooter (FPS) genre with its fast-paced action and 3D-like environments. While the original *DOOM* was built using C, our project aims to recreate a simplified shooter experience using Python, focusing on 2D mechanics with sprite-based rendering.

Python is not traditionally associated with game development due to its interpreted nature, which can lead to performance limitations. However, with frameworks like **Pygame**, developers can create interactive games efficiently. This paper outlines the methodology used to develop our shooter game, the challenges faced, and future improvements.

2 Related Work

Several Python-based game projects have attempted to replicate the mechanics of classic shooters. Pygame is widely used for 2D games due to its simplicity and ease of integration with Python's extensive libraries. Other frameworks such as **Panda3D** and **Godot (with Python scripting)** have been used for more complex game projects. However, our project focuses on a lightweight implementation suitable for beginners and those interested in Python-based game development.

3 Methodology

3.1 Tools and Technologies Used

The game was developed using:

- **Python 3.x** – Core programming language
- **Pygame** – Handles graphics, input, and events
- **Tiled Map Editor** – Used for level design

3.2 Game Mechanics

3.2.1 Player Movement and Shooting

The player character moves in a **2D top-down environment**, controlled using keyboard inputs (`WASD` for movement, mouse for aiming/shooting). Shooting mechanics involve:

- A **bullet sprite** that moves toward the mouse cursor
- Collision detection using **Pygame's rect-based collision system**
- A shooting cooldown to balance gameplay

3.2.2 Enemy AI

Enemies are programmed to:

- Patrol designated areas using **random movement patterns**
- Detect and chase the player using **line-of-sight calculations**
- Attack the player when in range, using **simple projectile mechanics**

3.3 Collision and Physics

Pygame's built-in **Rect** collision system was used for detecting:

- Player interactions with walls and obstacles
- Bullet collisions with enemies and environment
- Enemy-player melee attacks

3.4 Level Design

Levels were designed using **Tiled**, allowing for dynamic map loading. The environment consists of walls, doors, and destructible objects that enhance gameplay.

4 Results and Discussion

4.1 Performance and Optimizations

The game runs at **60 FPS** on most mid-range computers, with optimizations including:

- Using **spritesheets** instead of individual images to reduce memory usage
- Implementing **quad-tree partitioning** for efficient collision detection
- Reducing redundant computations in enemy AI behavior

4.2 Comparison to DOOM

While *DOOM* features **3D environments and complex AI**, our game provides a simplified **2D shooter experience**. Despite the differences, we replicated:

- **Fast-paced combat**
- **Multiple enemy types** with distinct behaviors
- **Weapon variety**, including hitscan and projectile-based shooting

5 Conclusion and Future Work

This project demonstrates that **Python and Pygame** are viable tools for developing 2D shooter games inspired by *DOOM*. By leveraging sprite-based graphics, enemy AI, and collision mechanics, we created an engaging game experience.

5.1 Future Improvements

- **Multiplayer Mode** – Implementing networked gameplay
- **Procedural Level Generation** – Generating random levels for replayability
- **Advanced AI** – Implementing pathfinding (A* algorithm) for smarter enemies
- **Graphics Enhancements** – Using shaders and particle effects for explosions

6 References

1. Shinners, P. (2000). *Pygame Documentation.* Available at: `https://www.pygame.org/docs/`
2. Abrash, M. (1997). *Graphics Programming Black Book.*
3. Id Software. (1993). *DOOM.* [Video game].