INTERIM REPORT STUDENT MANAGEMENT SYSTEM

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

– DATA SCIENCE WITH MACHINE LEARNING

By

ATHIRAD S PRAMOD

Registration No: 12306917

Section: K23HC

Roll No: 41

18/11/2024

Supervisor

Aman Kumar

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

October 2024

ACKNOWLEDGEMENT

I at this moment declare that the research work reported in the dissertation/dissertation proposal entitled "SHOOTER GAME PROJECT" in partial fulfillment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under the supervision of my research supervisor Mr. Aman kumar. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith directly complies with Lovely Professional University's Policy on plagiarism, intellectual property rights, and the highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents an authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Athirad S Pramod

REG.NO - 12306917

Table of Contents

I. INTRODUCTION

1.1 Project Overview

The Shooter Game Project is a 2D interactive game developed using Python and the Pygame library. In this game, the player controls a character that moves and shoots at enemies. The goal is to defeat enemies and progress through increasing levels of difficulty. The project is built on object-oriented principles to manage the various elements of the game like characters, enemies, and bullets. The project aims to deliver a fun and engaging experience through interactive mechanics and challenging gameplay.

1.2 Purpose and Significance

The purpose of this project is to design and develop an engaging 2D shooter game that will serve as an educational tool for learning game development concepts. By using Python and Pygame, the project aims to enhance skills in programming, game mechanics implementation, and user interaction. The game can be extended in the future with more complex features, like multiplayer functionality or advanced AI for enemies.

1.3 Project Features

- Player Movement: The player can move in four directions using keyboard controls and shoot using the mouse.
- Enemy Types: Different types of enemies with varied movement patterns and behaviors.
- Health and Score: The player's health decreases with enemy hits, and points are earned by defeating enemies.
- Levels: As the player progresses, the game introduces more enemies and faster-paced gameplay.
- Sound and Graphics: Incorporates sound effects for shooting and background music, as well as visual effects for explosions and bullet impacts.

II. Objectives and Scope of the Project

2.1 Project Objectives

- Create a 2D shooter game with player movement, shooting, and enemy interaction.
- Implement basic game mechanics like collision detection, scoring, and health systems.
- Design a progressive difficulty system where the game becomes more challenging with each level.

2.2 Project Scope

The project covers the development of a 2D shooter game, including:

- Player and Enemy Mechanics: Programming movement, shooting, and AI for enemies.
- Game Features: Health, score tracking, level progression, and game over conditions.
- Graphics and Sound: Design and integration of visual and audio assets to enhance gameplay.
- Future Enhancements: Possibilities include multiplayer, improved AI, additional levels, and more advanced graphics.

III. Application Tools

3.1 Software Applications

1. Python: The core programming language used for game logic, user interface, and object management.

2. Pygame: The game development library used to handle graphics, user input, sound, and game loop functionality.

3. Tiled: Used (optional) for designing game levels and backgrounds.

4. GIMP/Photoshop: Used to create and edit game sprites and assets.

3.2 Programming Languages

- Python: Ideal for rapid prototyping and simplicity in coding.

- Pygame: A Python library that provides tools for 2D game development, offering ease of handling game mechanics, rendering graphics, and processing input events.

IV. Project Structure of the Shooter Game

4.1 Main Components of the Project

- Player Class: Manages player movement, shooting, health, and interactions with enemies.
- Enemy Class: Handles the movement, AI, health, and damage of enemy characters.
- Bullet Class: Controls the movement and collision detection of bullets fired by the player.
- Game Class: Coordinates the game flow, including levels, scoring, and the main game loop.
- GUI: Displays in-game information like health, score, and current level.

4.2 Classes and Their Functions

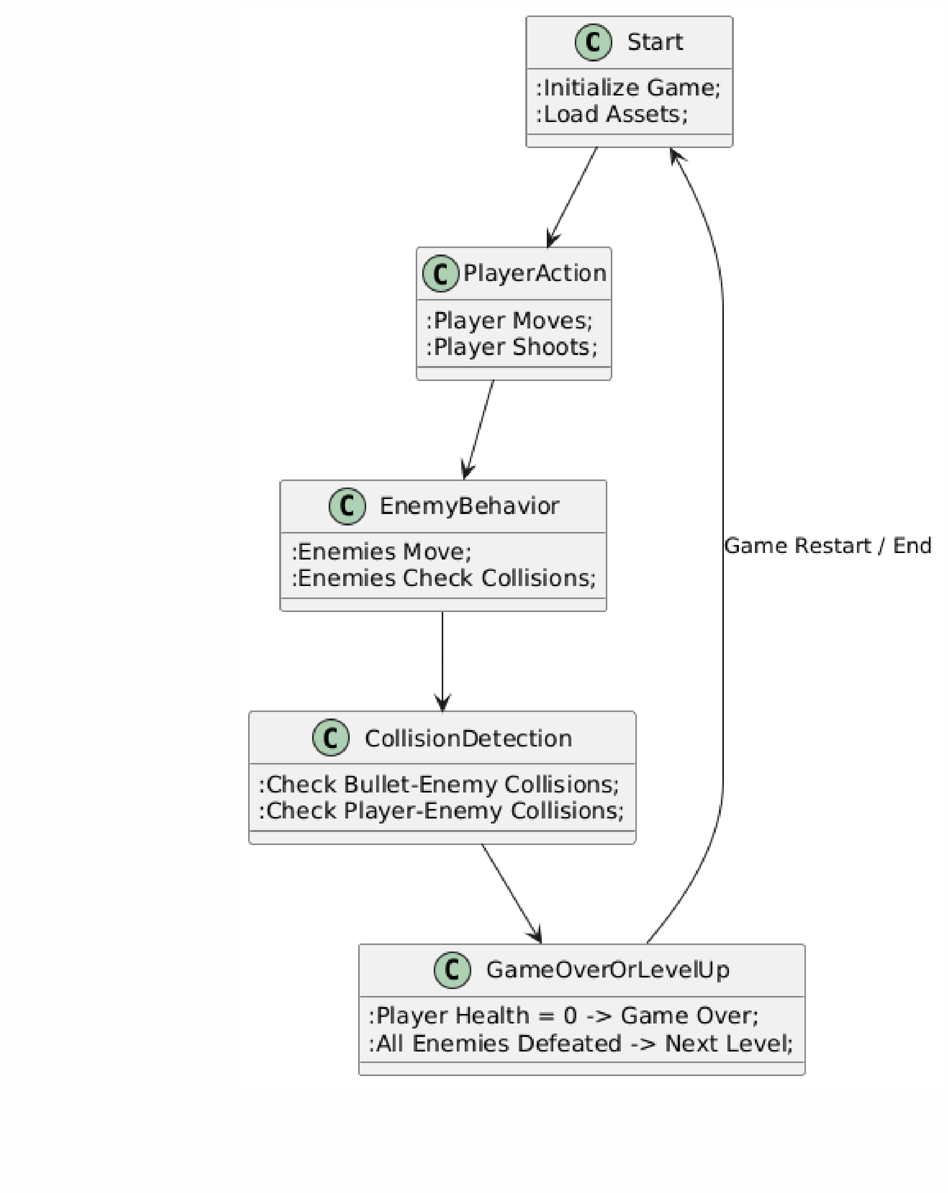
- Player: Manages player position, movement (via keyboard input), health, and shooting logic.
- Enemy: Contains behavior for moving and responding to player actions, including detecting collisions with bullets.
- Bullet: Defines bullet movement and checks if it collides with enemies.
- Game: Controls the flow of the game, including initializing levels, tracking scores, and updating the display.

4.3 Interaction Between Components

- The Player class interacts with the Bullet class to fire bullets when the player clicks the mouse.
- The Bullet class checks for collisions with Enemy instances, causing enemies to take damage or be destroyed.

- The Game class manages all the classes by updating the state of the game, handling input, and displaying the current game status to the player.

V. Flowchart or Algorithm of the Project



5.1 Explanation of the Flowchart

- Start: The game initializes, loading assets and setting up the first level.
- Player Action: The player moves and shoots bullets by pressing keys and clicking the mouse.
- Enemy Behavior: Enemies move across the screen and attempt to collide with the player.
- Collision Detection: The game checks for collisions between bullets and enemies, as well as the player and enemies.
- Game Over or Level Up: If the player’s health reaches zero, the game ends. If all enemies are defeated, the game progresses to the next level.
- End: The game loops through levels or ends when the player’s health reaches zero.