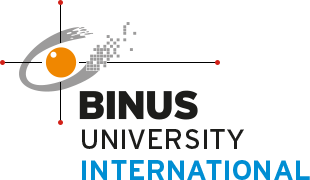
**Wolfenstein 3D Inspired Game**

Final Project Report



**Thristan Widjaja (2802503690)**

**L1BC**

# Jude Joseph Lamug Martinez MCS

**Algorithm and Programming**

**Binus International University**

**Jakarta**

**Table of Contents**

**Introduction** ………………………………………………………………………………………….3

1.1 Background and Objectives ………………………………………………………………….. 3

1.2. Project Overview …………………………………………………………………………….. 3

**Game Concept and Features** ……………………………………………………………………..4

2.1. Key Gameplay Features ……………………………………………………………………… 4

2.2. Tools and Technologies Used ………………………………………………………………… 5

**Solution Design** ……….……………………………………………………………………………6

3.1. Architecture Overview …………..…………………………………………………………… 6

3.2. Key Modules and Their Roles ...……………………………………………………………... 7

**Evidence of Working Program** …………………………………………………………………..9

4.1. Screenshots of Gameplay ……………………………………………………………………. 9

4.2. Video of Gameplay …………………………………………………………………………. 12

4.3. Diagrams ……………………………………………………………………………………. 12

**Conclusion** ………………………………………………………………………………………….13

4.1. Summary of Achievements …………………………………………………………………. 13

4.3. Lessons Learned …………………………………………………………………………….. 14

4.2. Future Improvements ……………………………………………………………………….. 14

**References** ………………………………………………………………………………………….15

**Introduction**

**Background and Objectives**

This project is a game inspired by the classic first-person shooter **Wolfenstein 3D**, developed by Bethesda. I chose this project because I wanted to challenge myself by creating something related to my personal interest in playing video games.

The main goal of this project is to recreate the nostalgic gaming experience of one of the earliest first-person shooter games. Wolfenstein 3D has been a major influence for countless modern games in the first-person shooter genre, making it a great source of inspiration. By developing this project, I aimed to better grasp and understand the challenges of developing a game, which has always interested me.

The game was created using the **Pygame module**, and focused on implementing the main mechanics of the original game. I intentionally kept the process simple to avoid overcomplicating the development while trying to deliver an enjoyable gaming experience.

**Project Overview**

The game uses a ray-casting engine in order to create a 3D-like environment. It takes inspiration from the original Wolfenstein 3D by using similar map textures and map layouts. The NPCs from the original game are also included as enemies. The player can hunt down the NPCs, but they can also be defeated by the numerous amounts of NPCs that spawn randomly on the map, resulting in a game over. The main.py file serves as the central script for initializing and running the game.

The GitHub Repository for the project can be found here:

[GitHub Repository Link](https://github.com/ThristanW/Algoprog-Final-Project)

**Game Concept and Features**

**Key Gameplay Features**

1. **First-Person View**

* The game allows the payer to look left and right by using the mouse.
* The user controls the player by using the W, A, S, and D keys.
* Every time the player takes damage from an enemy, there is a visual indicator in the form of a blood splatter around the peripheral of the players POV.

1. **Shooting and Combat**

* The player spawns in with a pistol that has a delay before the player can shoot the gun again.
* When the player shoots the gun, it hits whatever target is on the centre of the screen.
* After successfully shooting an enemy, the enemy will give feedback in the form of a pain frame in the animation.

1. **Enemies**

* Enemies use a simple pathfinding algorithm to locate the player’s location.
* When the enemy spots the player, it will move towards the player until a certain distance, and it will start shooting at the player. The enemy has a chance to miss or hit the player.
* The enemies have a set amount of health and they will die when the player hits the enemy a number of times.

1. **Sounds**

* When the player shoots, a distinct gunshot can be heard. The player also makes sound in response to pain.
* Enemies also have unique sounds when shooting. When they are shot by the player, they also make sound according to the pain and when they die.

**Tools and Technologies Used**

1. **Pygame**

* Using Pygame, the main game window, where the game is displayed can easily be set up.
* Pygame is also used to handle the inputs made by the user by detecting them and responding to the actions such as moving or looking around.
* The images, animations and sound effects are displayed using the Pygame module during the gameplay.
* Lastly, Pygame handles the main game loop, allowing the game to run smoothly.

1. **Visual Studio Code**

* By using extensions in visual studio code, the process was much faster, by means of autocompletion, syntax highlighting and error detection.
* VSC allows for easy navigation of files and folders in the directory.
* Bugs and errors were easily detected and identified which helped prevent issues during the development of the game.

**Solution Design**

**Architecture Overview**

The architecture of the game is very modular and simple, which allows for easy understanding of the modules and what they do, along with an easier way to expand or add more modules into the game code. Each module has its own unique role to handle specific tasks.

**Key Modules and Their Roles**

1. **Main Game Module (Main.py)**

The main.py module handles the initialization of the game window and the game loop, along with all the activities and interactions that happen in the game. This module:

* Starts the window and handles the game loop.
* Calls functions responsible for managing the game, rendering the map, and handling user inputs.
* Processes user inputs
* Coordinates any interactions between other components of the game and its modules, such as ray casting.

1. **Settings Module (settings.py)**

The settings.py module stores the constants and configuration values for the game. This module:

* Defines the resolution, FPS and other display settings for the game.
* Acts as a centralized place to store values that are responsible for certain gameplay mechanics.

1. **Player Module (player.py)**

The player.py module defines the character and its behaviours. This module:

* Is responsible for the handling of WASD keys for movement through the game world.
* Handles the shooting mechanics of the player’s gun
* Handles the players health and how much damage they deal and keep track of their values, updating them if they decrease with accordance to interactions in the game (when the enemy hits the player).

1. **Map Module (map.py)**

The map.py module manages the layout and structure of the game map. This module:

* Loads and stores the map layout as values of ‘\_’, ‘1’ and ‘2’, representing different wall textures and defines the floor in which the player can move on.
* Includes data for the ray casting engine in order to render the 3D map.

1. **NPC Module (npc.py)**

The npc.py module defines the enemies and their behaviours. This module:

* This module is responsible for handling the main elements of the enemies, such as their size, health, damage, attack distance and more.
* The location that the NPC spawns is also defined in this module as a random position.
* The collisions and animations are also handled by this module so that the NPC cannot simply walk through walls.

1. **Ray Casting Module (raycasting.py)**

The raycasting.py module is responsible for implementing the 3D rendering effect in the game. This module:

* Simulates a first person view by casting rays that act as the visible space within which the player can see.
* Determines the distances of the wall and the player in order to render the textures based on the location and orientation of the player.

1. **Object Renderer Module (object\_renderer.py)**

The object\_renderer.py module handles the visual representation of the game itself. This module:

* Draws the 3D world, like the walls, textures, and sprites.
* Gives the game more fluidity, by ensuring smooth transitions between each frame, to make the game more immersive.

1. **Object Handler Module (object\_handler.py)**

The object\_handler.py module handles in game objects and enemies. This module:

* Processes and updates the behaviour of the NPCs and also their animations, health, and damage values.
* Is responsible for object and player interactions, like collisions of the sprites and the player.

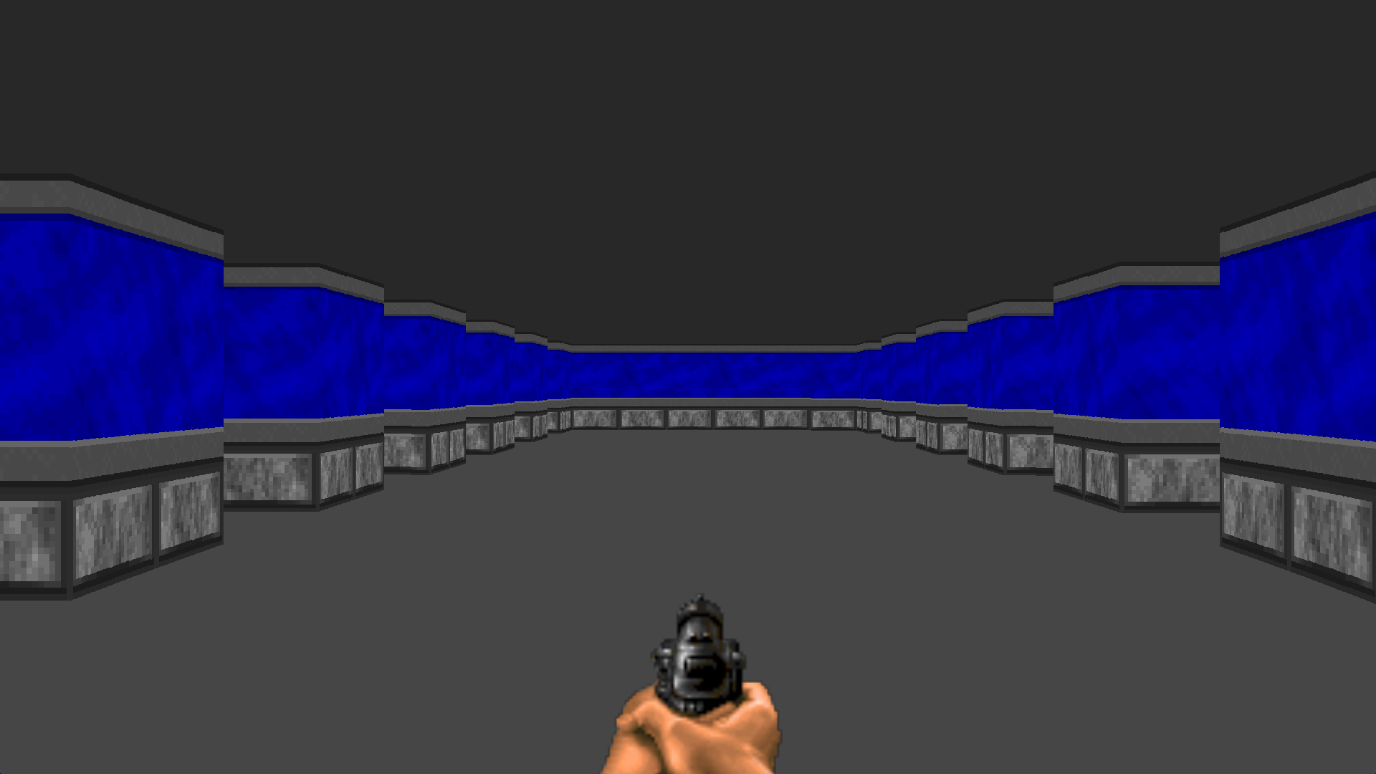
1. **Pathfinding Module (pathfinding.py)**

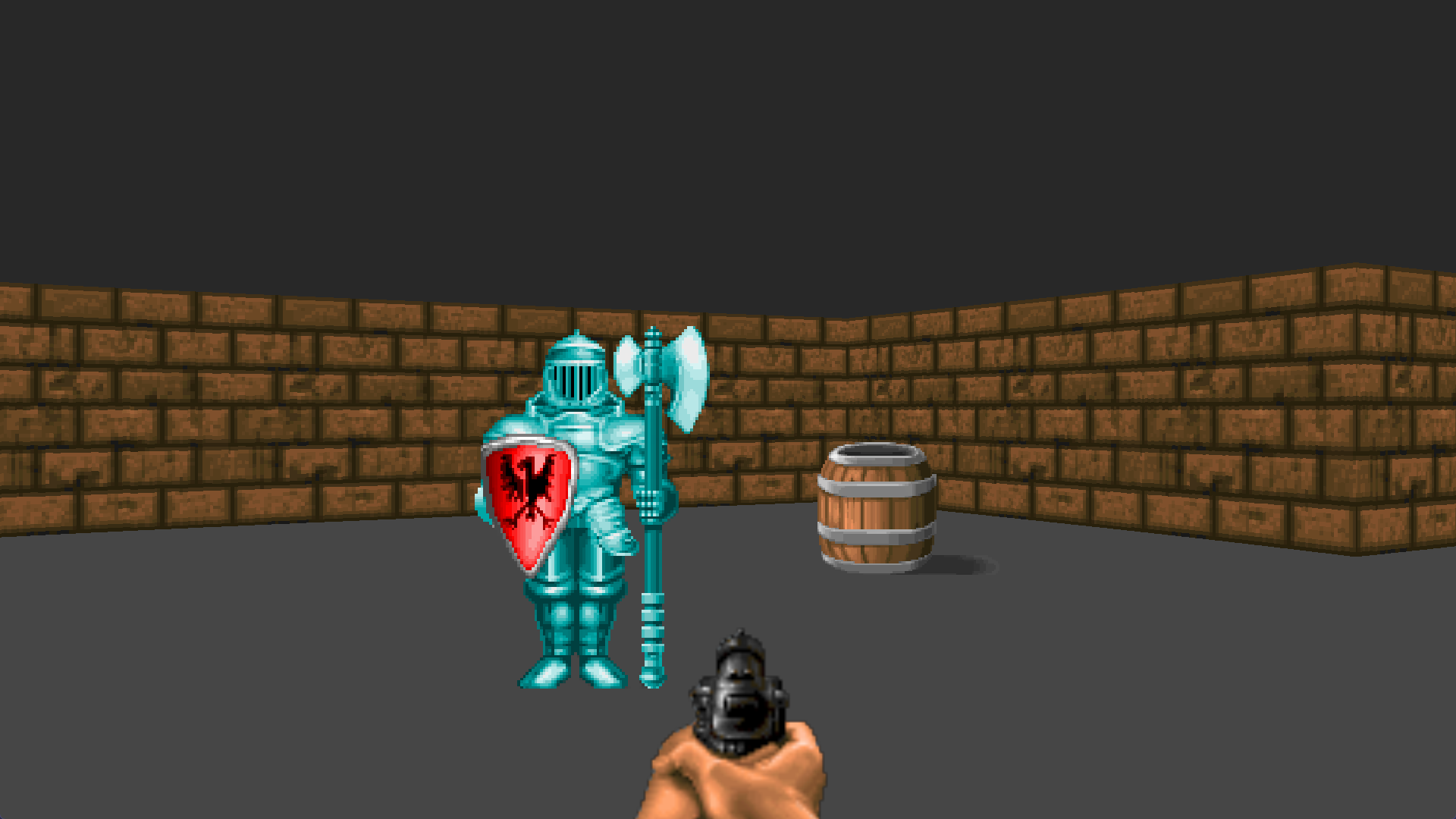
The pathfinding.py module implements enemy movement and logic. This module:

* Uses an algorithm to determine the shortest path in order to reach the player.
* Guides the NPC around obstacles such as the walls of the map and attacks the player it is a certain distance away from the player.

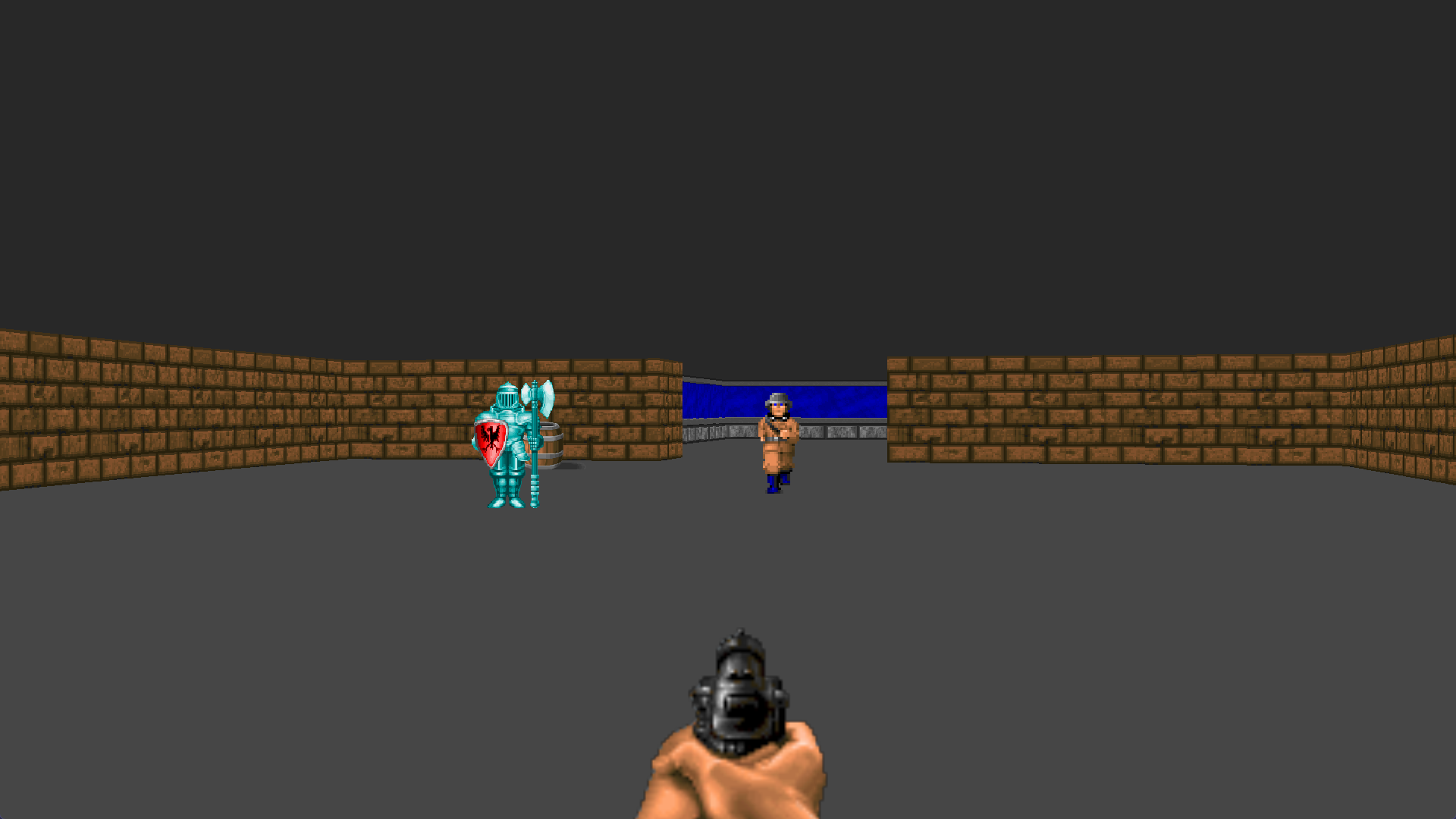
**Evidence of Working Program**

**Screenshots of Gameplay**

****Custom Map Layout

****

Static Sprites

Enemy NPC walking animation and pathfinding

NPC aggravation animation and death sprite

NPC shooting animation



Player pain textures

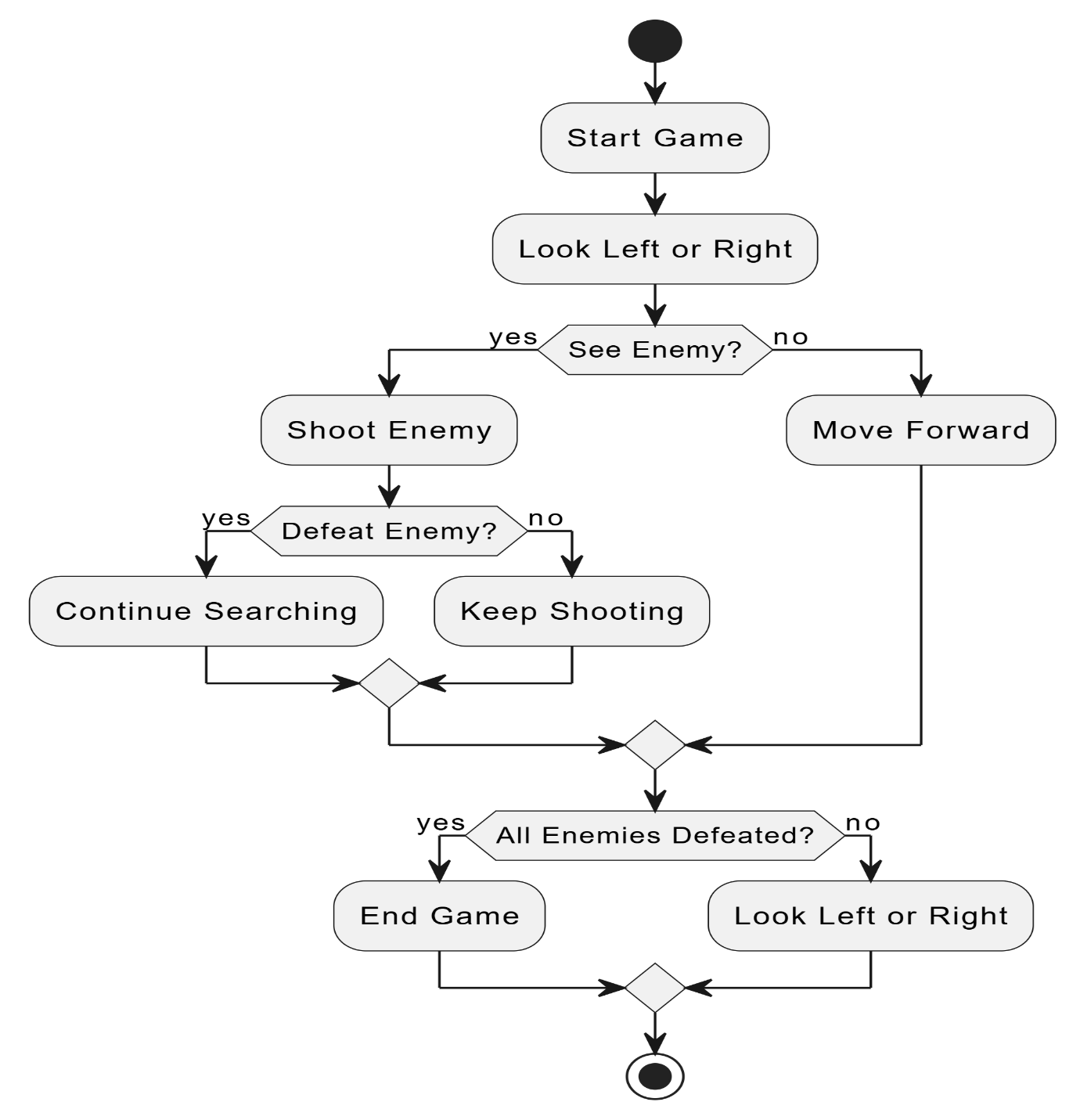
**Video of Gameplay**

The link for the video demo of the game is provided below:

[Video Demo Link](https://drive.google.com/file/d/1Kwh-C6vFtV6n3h0e1DLwUJZDHBnGP-NA/view?usp=drive_link)

**Diagrams**

Activity Diagram

****

**Conclusion**

In the final conclusion, I will summarize the overall results of this project and give some of my insights and suggestions for future projects

**Summary of Achievements**

In conclusion, I am proud to say that I have done a good job implementing the core mechanics of game along with a bit of my own twist while respecting the game mechanics of Wolfenstein 3D. While I did enjoy the process of the development of the game, I won’t lie, some parts of it were extremely challenging with bugs that I had to manually find and fix, which sometimes made me a feel a little overwhelmed, but other than that, to some extent I enjoyed confronting into the issue and spotting bugs as it helped me gain an overall feeling of satisfaction when the program started to work. I will be carrying forward this feeling while hopefully developing new and more improved programs in the future.

**Learning Outcomes**

During the process of this project, I have acquired a number of new skills:

Understanding Pygame: During the developing of this project, I learnt to understand how to implement the Pygame library to develop and control a game, such as processing user inputs, displaying graphics, and controlling the game loop.

Implementing Ray-Casting: I learned how to properly use ray-casting and learnt a bit of the theory behind it in order to make my game appear 3D and implement a first-person view.

Pathfinding Algorithms: I became more familiar with the process of using pathfinding on enemy AI for certain games, so that the AI can find the player.

Problem-Solving Skills: Debugging and troubleshooting of unwanted errors has taught me to identify problems from a different perspective, overall developing my problem-solving skills and efficiency to apply solutions.

These abilities have not only increased my knowledge of game development but also my problem-solving and programming skills.

**Future Improvements**

Many improvements can be added to this game to make it more consistent and enhance player experience. Features such as more flexible health mechanics, an ammo and reloading system, multiple weapons, or even a stealth system could add depth to the gameplay. The sprites could also be improved, as some are inconsistent, with issues like floating sprites or mismatched shadows, and they lack smooth animations. Furthermore, a more advanced algorithm can be used to improve the pathfinding of the enemy AI, such as implementing a cover system where enemies seek cover when shot, or implementing an awareness system, where the enemy has a field of view where in which they can detect the player. By refining these aspects and optimizing performance, I hope to create a smoother and more polished game in the future.

**References**

* <https://www.youtube.com/watch?v=ECqUrT7IdqQ&list=PLi77irUVkDatlbulEY4Kz8O107HO8RGH8&ab_channel=CoderSpace>
* <https://www.youtube.com/watch?v=yJXuvK_eLrQ&ab_channel=CoderSpace>
* <https://www.spriters-resource.com/ms_dos/wolfenstein3d/>
* <https://soundspunos.com/game/1161-sounds-and-music-from-the-game-wolfenstein-3d.html>
* <https://forum.zdoom.org/viewtopic.php?t=52873&start=180>
* <https://www.areyep.com/RIPandMCS-TextureLibrary.html>