AI LAB (CS39002) PROJECT REPORT

on

"Zombie Escape"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
Computer Science And Engineering

BY

SAURAV JHA	22053276
JANAMJAY KUMAR	22052986

UNDER THE GUIDANCE OF
Dr Sricheta Parui

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

March 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

"Zombie Escape"

submitted by

Saurav Jha	**22053276**
Janamjay kumar	**22052986**

is a record of Bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: 23 / 03/ 2025

Dr Sricheta Parui
Project Guide

Acknowledgement

We sincerely thank Dr. Sricheta Parui for her continuous guidance, support, and valuable feedback throughout our project. Her insights greatly contributed to the successful completion of "Zombie Escape."

Saurav kumar Jha

Janamjay kumar

ABSTRACT

"Zombie Escape" is a dynamic survival-based game developed using Python and Pygame. The player must navigate through a procedurally generated map using Perlin noise, avoid zombies using strategic movements, and reach the safe zone. The game employs AI algorithms like **Breadth-First Search (BFS)** for pathfinding, ensuring challenging gameplay. The weather system, resource management, and noise-based AI behavior add to the immersive experience.

The report thoroughly analyzes the game's design, focusing on the implementation of Perlin noise for procedural map generation, BFS for intelligent zombie pathfinding, and the innovative noise system that dynamically influences gameplay. Detailed descriptions of the codebase, along with class diagrams, state diagrams, and flowcharts, are provided. Additionally, the report highlights the extensive testing process, validation methods, and observed results.

By effectively blending AI algorithms with procedural generation and interactive game mechanics, "Zombie Escape" serves as a comprehensive demonstration of applied AI concepts. The findings from this project pave the way for further improvements, including adaptive AI behavior and enhanced visual effects. This report concludes with insights into the challenges faced, lessons learned, and the potential future enhancements to elevate the game experience further

Keywords: AI Game, Path-finding, Pygame, Breadth-First Search (BFS), Game Development, noise, safe Zone

Contents

Chapter 1: Introduction

"Zombie Escape" is an engaging survival game that challenges players to navigate a hostile, post-apocalyptic environment. The primary objective is to evade AI-controlled zombies, strategically manage resources, and reach the designated safe zone. The game's immersive design relies heavily on AI algorithms and procedural map generation to enhance the unpredictability and excitement of gameplay.
To add complexity, the game features a noise management system where player actions generate noise, attracting zombies. The artificial intelligence uses **Breadth-First Search (BFS)** for intelligent zombie pathfinding. Additionally, the implementation of Perlin noise generates realistic and diverse terrains, providing a unique gameplay experience in every session. A dynamic weather system introduces visual and gameplay challenges, affecting visibility and zombie behavior.

By integrating AI concepts into game mechanics, "Zombie Escape" serves as an ideal platform to demonstrate AI application in real-time decision-making scenarios. This report provides a detailed insight into the game's development process, including the algorithms and design choices that shaped its implementation.

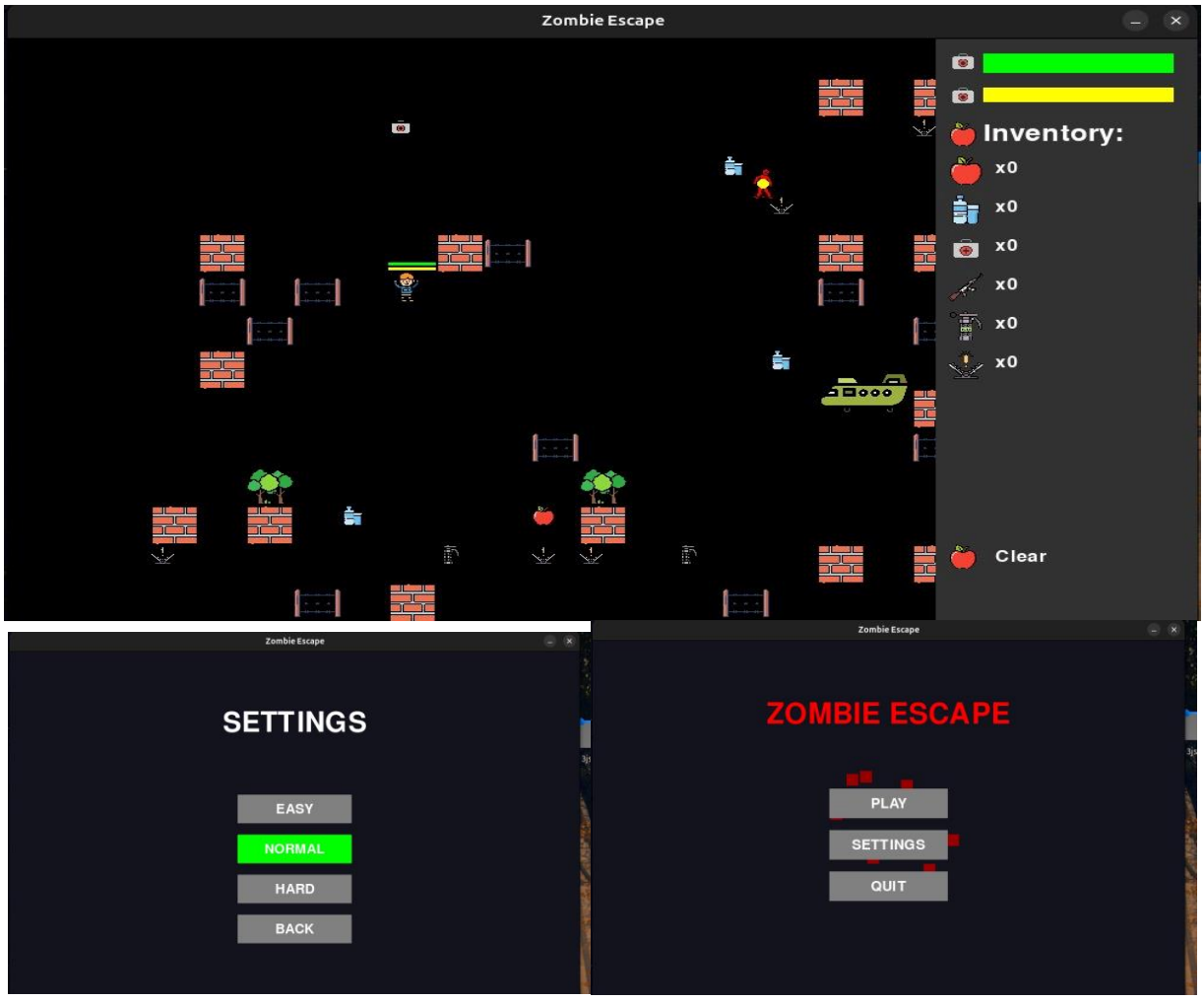


Fig: Game Display

Chapter 2: Literature Review

2.1 Game AI and Path-finding

Artificial Intelligence (AI) significantly enhances gaming experiences by creating immersive and responsive non-player character (NPC) behaviors. Pathfinding, a key AI capability, enables NPCs to navigate game environments while avoiding obstacles. Algorithms like Dijkstra's, A-Star (A*), and Breadth-First Search (BFS) are commonly used for efficient pathfinding.

In Zombie Escape, the BFS algorithm is employed for zombie pathfinding. By exploring the game map level by level, BFS ensures zombies follow the shortest path to the player. This guarantees optimal navigation, contributing to a dynamic and challenging gameplay experience.

The implementation of BFS in Zombie Escape reflects best AI practices, balancing efficiency, adaptability, and realism to deliver an engaging and realistic gaming experience.

2.2 Pygame for Game Development

Pygame is a versatile Python library designed for 2D game development, offering a comprehensive suite of tools for rendering graphics, managing input devices, and handling audio. It streamlines tasks such as sprite management, animations, and event handling, making it an excellent choice for indie developers.

In Zombie Escape, Pygame is used to render the game grid, display zombies and obstacles, and detect collisions. The library’s optimized functions ensure smooth animations and responsive real-time input handling, enhancing the overall gaming experience.

2.3 Procedural Map Generation with Perlin Noise

Zombie Escape uses Perlin noise to generate its map dynamically. This ensures that every playthrough offers a unique experience, encouraging replayability. Terrain features like obstacles, walls, and fences are procedurally placed, adding depth and realism to the game's environment..

2.3 Noise Management for AI

Noise gradually decays over time, reducing the chance of attracting more zombies. This mechanic introduces a strategic element, forcing players to carefully manage their actions and plan their movements to avoid overwhelming hordes of zombies. The dynamic noise system enhances the game's realism and keeps the player engaged.

Chapter 3: Problem Statement / Requirement Specifications

The objective of "**Zombie Escape"** is to create an immersive and challenging survival game that showcases the effective implementation of AI algorithms. The core problem addressed is the realistic simulation of zombie behavior using AI, while ensuring unpredictable and engaging gameplay. Additionally, implementing procedural map generation enhances the replayability of the game.

3.1 Project Planning

The project was planned in the following phases:

1. **Grid-based game environment setup** - Establish the game map using Perlin noise.
2. **Implementation of player movement mechanics** - Develop smooth and responsive player controls.
3. **Obstacle generation for enhanced gameplay dynamics** - Create dynamic obstacles using Perlin noise.
4. **AI-driven zombie movement using BFS path-finding** - Ensure zombies accurately track the player using BFS.
5. **Game over conditions and win/loss logic** - Implement clear conditions for player victory or defeat.

3.2 System Requirements

Functional Requirements:

- The system shall display a procedurally generated, grid-based game environment.
- The player shall be able to move in four directions (up, down, left, right)
- AI-controlled police shall track the player using BFS
- The game shall include obstacles, resources, and traps to create strategic challenges.
- The game shall determine win/loss conditions based on player and zombie positions.

3.3 System Design

The game is designed with the following key components:

1.User Interface: Manages the rendering of the game grid, player input, and smooth animations.

2. Zombie AI: Implements BFS to ensure zombies take the shortest path to the player.

3. **Noise Management System**: Tracks player noise generation and influences zombie movement.

4.Game Logic: Handles win/lose conditions and resets the game.

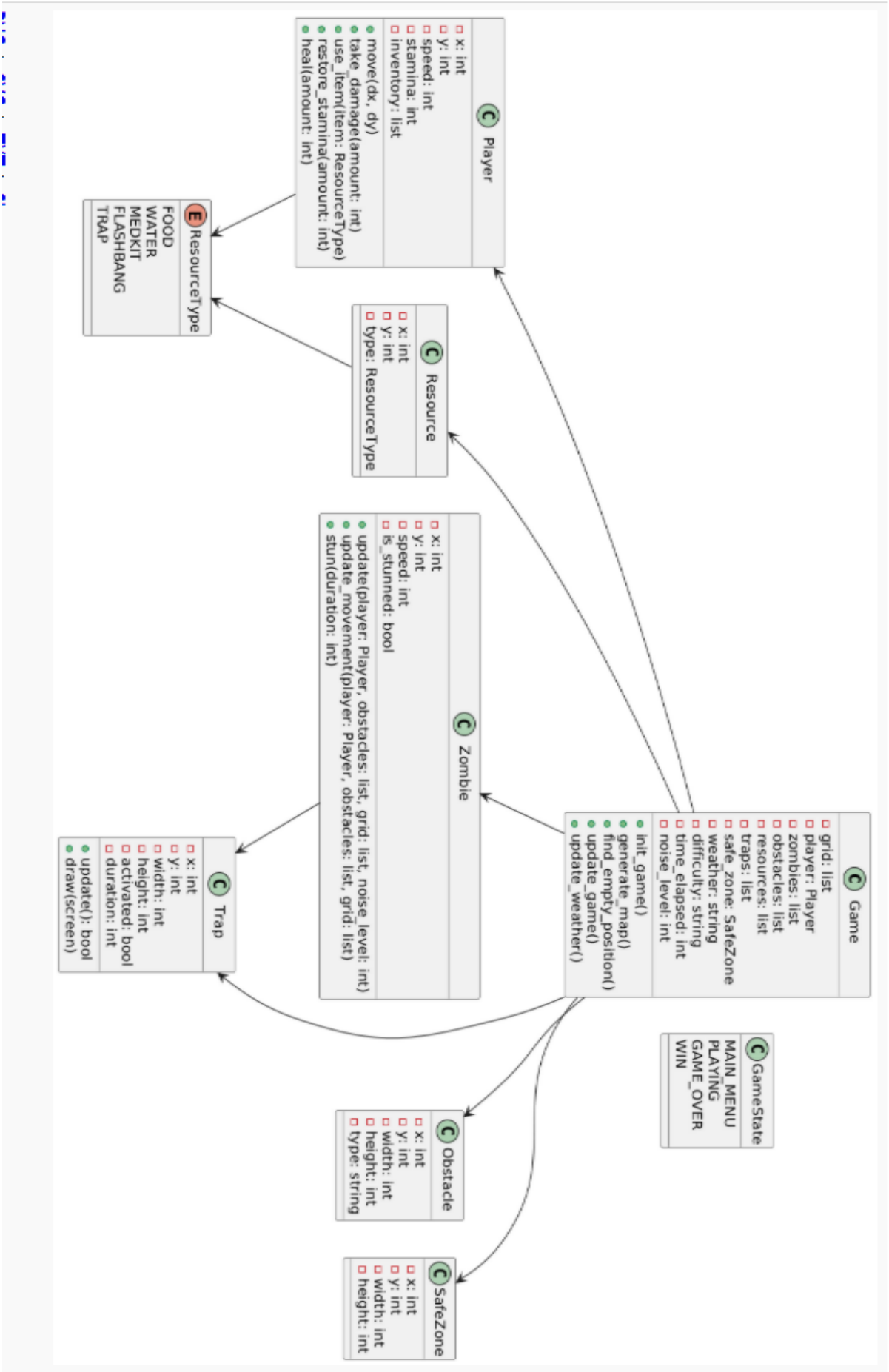


Fig1. UML Class Diagram

Chapter 4: Implementation

The implementation phase of the Zombie Escape project successfully translated our design concepts into a functional Python game using the Pygame library. This chapter details the technical implementation, challenges encountered, and solutions developed.

4.1 Development Environment

- **Programming Language**: Python 3.9
- **Game Development Library**: Pygame 2.1.2
- **Additional Libraries**:
  - NumPy (for mathematical operations)
  - Perlin Noise (for procedural terrain generation)
  - Collections (for deque data structure used in BFS algorithm)

- **Development Tools**:
- Visual Studio Code with Python extension
- PyCharm Professional
- Git for version control

4.2 AI Component Implementation

The AI component manages zombie behavior using a state machine architecture with three primary states:

- **Idle**: Random wandering with occasional pauses
- **Investigate**: Moving toward a noise source
- **Chase**: Directly pursuing the player using BFS pathfinding

The Breadth-First Search algorithm was implemented for zombie pathfinding.Each zombie maintains its own state and can transition based on environmental stimuli:

- Detecting player noise causes transition from Idle to Investigate
- Direct line of sight to player causes transition to Chase
- Losing track of player reverts to Investigation or Idle

The zombies' intelligence scales with difficulty settings, with higher difficulties featuring:

- Faster pathfinding updates
- Wider detection radius
- Better memory of player's last known position

4.3 Game Flow and Mechanics

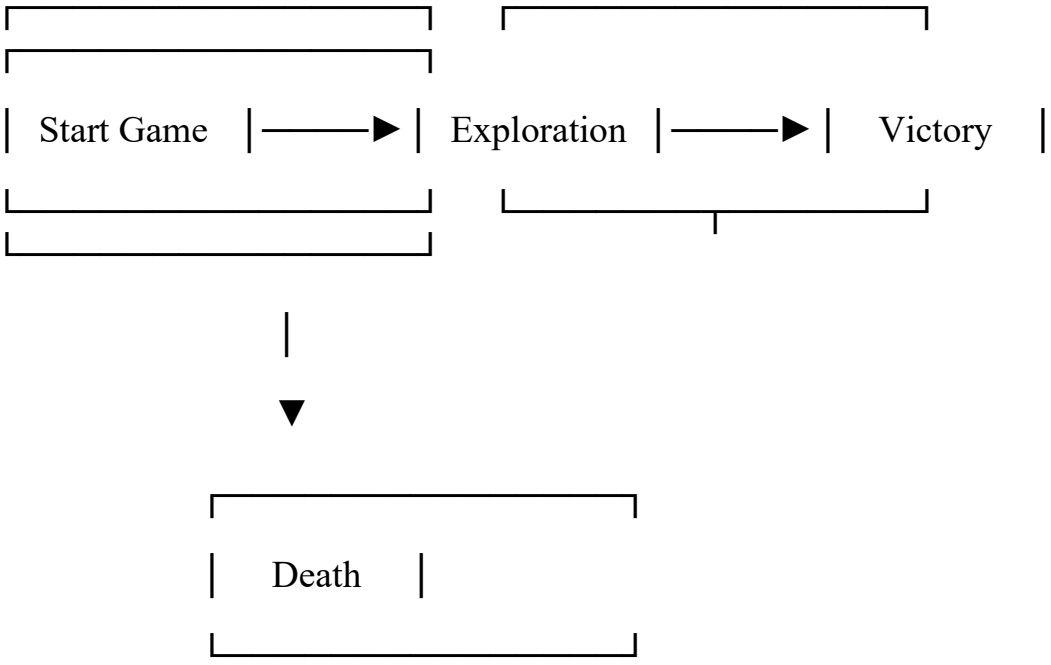
The game logic component manages the core gameplay loop:

- Game state transitions
- Win/lose conditions
- Event handling
- Resource management
- Trap and item mechanics

The main game loop follows this structure:

1. Process player input
2. Update game state
3. Update AI behavior
4. Check for collisions
5. Update noise propagation
6. Update weather effects
7. Render the game
8. Check win/lose conditions

Game Flow Diagram:



4.4 User Interface Implementation

The user interface was implemented with several components:

- **HUD**: Displays player health, stamina, and inventory
- **Minimap**: Shows player position relative to safe zone

- **Menu System**: Main menu, settings, and game over screens
- **Tutorial Overlay**: Guides new players through basic controls

4.5Audio System Implementation

The audio system manages all sound effects and ambient audio:

- **Spatial Audio**: Zombie sounds get louder as they approach
- **Dynamic Music**: Tension music increases with zombie proximity
- **Ambient Sounds**: Weather effects and environmental ambiance
- **Feedback Sounds**: Resource collection, trap deployment, damage

4.6 Weather and Environmental Effects

We implemented a dynamic weather system to add variety and challenge:

- **Fog**: Reduces visibility radius
- **Rain**: Slows movement and creates ambient noise
- **Night**: Darkens the environment and increases zombie aggression

4.7 Procedural Map Generation

The map generation system uses Perlin noise to create varied and interesting gameplay environments. By adjusting noise parameters, we can control terrain features, obstacle density, and resource distribution.

Key implementation aspects include:

- Seed-based generation for reproducible maps
- Multi-octave noise for natural-looking terrain
- Thresholding techniques for distinct terrain features

4.8Testing and Debugging

Throughout development, we employed various testing methodologies:

- Unit Testing: Validating individual functions and methods
- Integration Testing: Ensuring components work together correctly
- Playtesting: Gathering feedback on gameplay experience
- Performance Testing: Identifying and resolving bottlenecks

Key challenges addressed during testing included:

- Optimizing BFS pathfinding for multiple zombies
- Balancing noise propagation for fair gameplay
- Ensuring procedurally generated maps are completable

Maintaining consistent frame rates during complex scenarios

Chapter 5: Standards Adopted

5.1 Design Standards

The game follows UML-based design principles to maintain clarity and scalability. Key design elements include:

- Class Diagrams: Representing the Zombie Escape structure
- State Diagrams: Illustrating various game states and transitions
- Sequence Diagrams: Defining event handling between game components
- ISO/IEC 25010 Compliance: Ensuring software quality attributes like functionality, reliability, and performance
- Modular Grid-Based Design: Optimizing object placement and movement logic
- Procedural Generation Standards: Following best practices for balanced map creation

5.2 Coding Standards

The code adheres to best practices and clean coding principles, ensuring maintainability and efficiency:

- Consistent Naming:
  - snake_case for variables and functions
  - CamelCase for classes
  - UPPER_CASE for constants
- Modular Functions: Following the Single Responsibility Principle (e.g., move_player(), check_zombie_collision())
- Optimized Logic: Efficient loops, short-circuiting conditions for performance
- Error Handling:
  - try-except for missing assets
  - Boundary checks for valid movement
- OOP Principles: Implementing encapsulation and modular game logic within classes
- Documentation: Clear docstrings explaining function purpose, parameters, and return values

5.3 Testing Standards

To ensure software reliability and functionality, the game follows internationally recognized testing standards:

- ISO/IEC 29119: Guidelines for functional and performance testing
- IEEE 829: Standardized test documentation framework
- Testing Types:

    - Unit Tests: Verifying core functions (move_player(), get_valid_moves())
    - Integration Tests: Ensuring smooth interaction between movement, collision detection, and AI path analysis
    - System Tests: Validating end-to-end gameplay scenarios
- Manual & Automated Testing:
    - Checking performance, memory efficiency, and frame rate consistency (targeting 60 FPS)
    - Ensuring stable AI decision-making and response times

5.4 Security and Data Integrity Standards

- ISO/IEC 27001: Information security best practices applied to ensure secure data handling
- Input Validation: Preventing unexpected crashes from invalid user inputs
- Safe File Handling: Ensuring game saves and logs do not corrupt game state

5.5 User Experience and Accessibility Standards

- ISO 9241-110: Guidelines for interactive systems to ensure a smooth user experience
- Color Contrast & Readability: Ensuring UI elements are easily distinguishable
- Keyboard & Controller Support: Providing accessibility for different control schemes
- Customizable Difficulty: Allowing players to adjust challenge levels according to their skill

Chapter 6: Conclusion and Future Scope

6.1 Conclusion

The development of "Zombie Escape" successfully implements AI-driven zombie behavior, procedural map generation, and dynamic gameplay mechanics using Pygame. The game leverages pathfinding algorithms and noise propagation to create engaging and strategic survival scenarios, ensuring an immersive experience for players.

The project follows structured coding practices, modular design principles, and industry standards, making it scalable and maintainable. Comprehensive testing ensures the stability of AI behavior, collision detection, and player interactions. The game serves as both an entertainment tool and a demonstration of AI-driven decision-making in a simulated environment.

Key achievements include:

1. Successful implementation of BFS for efficient zombie pathfinding
2. Effective procedural map generation using Perlin noise
3. Dynamic noise system that meaningfully impacts gameplay
4. Weather effects that create varied gameplay experiences
5. Balanced resource management that encourages strategic decision-making

Overall, this project highlights the practical application of artificial intelligence in game development, reinforcing how AI can enhance game realism and challenge players in an engaging manner.

6.2 Future Scope

The project has significant potential for future enhancements:

1. Advanced AI for Zombies
    - Implement adaptive enemy behavior and learning algorithms
    - Add varied zombie types with unique abilities and behaviors
    - Implement more sophisticated group behavior and coordination
2. Enhanced Procedural Generation
    - Add building interiors and underground areas
    - Implement story-driven procedural generation
    - Create more diverse biomes and environmental features
3. Multiplayer Mode
    - Enable cooperative gameplay for 2-4 players
    - Implement competitive modes (e.g., player vs. player with zombies)
    - Add online leaderboards for score tracking
4. Advanced Graphics and Effects
    - Implement dynamic lighting and shadows
    - Add particle effects for weather and interactions

    - Enhance animation quality and variety
5. Expanded Gameplay Mechanics
    - Add crafting and base-building systems
    - Implement a more detailed survival simulation (hunger, thirst, etc.)
    - Create a progression system with unlockable abilities
6. Cross-Platform Compatibility
    - Optimize for mobile devices
    - Create web browser version using WebAssembly
    - Support for console platforms

These enhancements would significantly expand the game's depth, replayability, and appeal to a wider audience.

Individual Contribution Report

Zombie Escape

Janamjay Kumar
22052986

Abstract

The project Thief and Police Chase is a strategic grid-based game where a player The project Zombie Escape is an AI-based game where zombies chase a player using a BFS pathfinding algorithm. The game is set on a grid with obstacles, requiring the player to strategically navigate to the safe zone while avoiding the zombies.

Individual Contribution and Findings

My primary role involved gathering information, structuring the project report, and assisting in the presentation preparation. Key contributions include:
Report Preparation: Collected relevant information regarding AI algorithms, game mechanics, and pathfinding to create well-documented sections of the report.
Documentation: Explained the project's background, methodology, and results in a clear and comprehensive manner.
Presentation Support: Curated detailed information for the project slides, ensuring the content aligned with the report.
Collaboration: Coordinated with my groupmate to ensure technical accuracy in the report and presentation.

Individual Contribution to Project Report Preparation

I contributed significantly to the report's structure and writing, particularly in explaining the game's purpose, AI behavior, and results. I also ensured the report adhered to academic guidelines and maintained clarity and coherence.

Individual Contribution to Project Presentation and Demonstration

I contributed in gathering and organizing the content. This ensured the presentation effectively conveyed the game's core concepts. I provided insights into how the AI algorithms enhanced the gameplay experience and assisted in answering questions during the demonstration.

Saurav Kumar Jha

22053276

Abstract

The project Zombie Escape is a survival-based AI game where a player navigates through a grid while being pursued by AI-controlled zombies. The zombies use the Breadth-First Search (BFS) algorithm for pathfinding, ensuring realistic and challenging movements. The objective is for the player to reach a designated safe zone without being caught

.Individual Contribution and Findings

I designed and developed the entire game architecture, handling core mechanics, UI, obstacle placement, and AI implementation. My key contributions include:

- Game logic and structure – Designed the complete game system, ensuring smooth interaction between the thief and AI police.
- Obstacle placement & collision detection – Developed realistic movement constraints and strategic path obstructions to challenge the player.
- AI behavior & BFS implementation – Designed and implemented the BFS-based pathfinding to ensure police efficiently chase the thief.
- User interface & game rendering – Used Pygame to develop the grid-based visual representation, making the gameplay interactive and engaging.
- Optimization & debugging – Refined movement mechanics, improved AI logic, and ensured smooth performance across different grid sizes.

Individual Contribution to Project Report Preparation

I wrote main sections on game logic, UI rendering, obstacle placement, and AI mechanics, providing in-depth explanations of how different components work together.

Individual Contribution to Project Presentation and Demonstration

For the project presentation, I prepared slides related to police movement mechanics, AI pathfinding, and game rules and assisted in demonstrating how AI pursues the thief in real time.

Signature:

Full Signature of Supervisor: _____
Full Signature of the Student (Saurav Jha): _____
Full Signature of the Student (JanamJay Kumar): _____