FAST National University Karachi Campus

Artificial Intelligence

Project Title

a Pac-Man game with AI-controlled ghosts

Submitted By

Hamza Rashid (22k-5097)

Tanzil –Ur–Rehman (22k-5088)

Instructor

Ms. Almas Ayesha Ansari

Submission Date

28-March-2025

1. Project Overview

Project Topic:

This project involves developing a **Pac-Man game with AI-controlled ghosts** using the A* pathfinding algorithm. The ghosts will autonomously chase Pac-Man using optimized movement strategies. The game will be built using **Python and Pygame**.

Objective:

The primary goal of this project is to **implement an intelligent ghost AI** that dynamically pursues Pac-Man using A* pathfinding, making the game challenging and engaging. The AI will adapt its path based on Pac-Man's movements to simulate a strategic chase.

2. Game Description

Original Game Background:

Pac-Man is a classic arcade game where the player controls Pac-Man, navigating a maze while collecting dots and avoiding ghosts. The goal is to collect all dots without being caught by ghosts. Traditionally, ghosts have specific movement patterns.

Innovations Introduced:

- **AI-Powered Ghosts:** Unlike classic Pac-Man, where ghosts follow preset paths, the ghosts in this project use the *A pathfinding algorithm** to find the shortest route to Pac-Man dynamically.
- **Adaptive Behavior:** The ghosts will **randomly deviate** from optimal paths at times, making their movement less predictable.
- **Game Over Mechanic:** If the ghost reaches Pac-Man, the game displays a **Game Over** message and stops.
- **Potential Future Enhancements:** Multiple ghosts with different AI behaviors, varying difficulty levels, and reinforcement learning for ghost decision-making.

3. AI Approach and Methodology

AI Techniques to be Used:

- *A* Pathfinding Algorithm:* Used to determine the shortest path from the ghost to Pac-Man.
- **Randomized Decision Making:** Occasionally, ghosts will follow a random path to avoid predictability.

Heuristic Design:

- The heuristic for A* will be the **Manhattan distance** (sum of absolute differences between x and y coordinates) since movement is constrained to a grid.

Complexity Analysis:

- *A* Algorithm Complexity* The worst-case time complexity is **O(b^d)** (where **b** is the branching factor and **d** is the depth of the search tree). Given the maze constraints, performance will be optimized by caching paths when possible.
- **Challenges:** Efficiently recalculating paths in real time without causing performance issues.

4. Game Rules and Mechanics

Modified Rules:

- The player controls **Pac-Man** using arrow keys.
- **Ghosts actively chase Pac-Man** instead of following predefined routes.
- If Pac-Man **collides with a ghost, the game ends**.

Winning Conditions:

- The game currently **has no win condition**—the objective is to survive as long as possible.
- A future improvement could involve Pac-Man collecting all dots to win.

Turn Sequence:

- **Pac-Man moves first**, controlled by the player.
- **Ghost AI moves next**, recalculating the path using A*.
- The sequence repeats until Pac-Man is caught.

5. Implementation Plan

Programming Language:

- **Python**

Libraries and Tools:

- **Pygame** (for game rendering and user interaction)
- **Heapq** (for A* priority queue implementation)
- **Random** (for AI variation)
- **NumPy** (if additional optimizations are needed)

Milestones and Timeline:

- **Week 1-2:** Game design, maze creation, and rule finalization
- **Week 3-4:** Implement Pac-Man movement and basic game loop
- **Week 5-6:** Implement ghost AI using A* pathfinding
- **Week 7:** Optimize AI behavior and add game over condition
- **Week 8:** Final testing and report preparation.

6. References

- **A* Pathfinding Algorithm:** Red Blob Games [https://www.redblobgames.com/pathfinding/a-star/]
- **Pygame Documentation:** [https://www.pygame.org/docs/]
- **Classic Pac-Man Game Analysis:** [https://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior]