Project Title: Pac-Ludo Fusion: AI-Powered Maze Chase

Submitted By:
Kainat Faisal (22k4405) | Fiza Farooq (22k4546) |
Samra Shahid (22k4585) | Laiba Khan (22k4610)

Course: AI
Instructor: Ravia Ejaz
Submission Date: 12/03/25

---

1. Project Overview

- **Project Topic:**
  A hybrid of Ludo and Pac-Man set on a dynamic maze board. Players race to collect pellets and return tokens to "home" while avoiding AI-controlled ghosts. Innovations include modular maze tiles, power-ups, and adaptive ghost behaviors.

- **Objective:**
  Develop an AI system where ghosts use Minimax, A* pathfinding, and reinforcement learning to challenge players. The game will test strategic planning under pressure, with AI adapting to player tactics.

---

2. Game Description

- **Original Game Background:**
  Traditional Ludo involves moving tokens around a fixed path. Pac-Man focuses on maze navigation and avoiding enemies.

- **Innovations Introduced:**

  1. **Modular Maze Board:** Hexagonal tiles rearrange periodically, creating new paths and obstacles.
  2. **AI Ghosts with Personalities:**
     - **Blinky:** Uses Minimax to anticipate player moves.
     - **Inky:** Employs A* pathfinding for efficient chasing.
     - **Pinky:** Randomizes behavior using probabilistic models.
  3. **Power-Ups:** Collect pellets to freeze ghosts, teleport tokens, or unlock shortcuts.
  4. **Cooperative-Competitive Play:** Players can collaborate to defeat ghosts or sabotage rivals.

---

3. AI Approach and Methodology

- **AI Techniques:**
    - **Minimax Algorithm** (modified for 4-player interaction) for Blinky's strategic decisions.
    - *A Algorithm** for Inky's optimal pathfinding.
    - **Reinforcement Learning** (Q-learning) to let Pinky adapt to player patterns over time.
    - **Alpha-Beta Pruning** to optimize Minimax depth.
- **Heuristic Design:**
    - Evaluate game states based on:
        - Pellet density in player's vicinity.
        - Distance to home for each token.
        - Ghost threat level (proximity and behavior type).

---

4. Game Rules and Mechanics

- **Modified Rules:**
    1. Players start with 3 tokens; all must return home after collecting 10 pellets.
    2. Ghosts respawn after 3 turns unless a "power pellet" is active.
    3. Maze tiles shift every 5 turns, altering movement options.
- **Winning Conditions:**
  First player to bring all tokens home with the required pellets wins.
- **Turn Sequence:**
    1. Player moves one token.
    2. Ghosts take actions based on AI logic.
    3. Board updates (e.g., tile rotation, power-up activation).

---

5. Implementation Plan

- **Programming Language:** Python (Pygame for GUI).
- **Libraries:**
    - Pygame (visualization), NumPy (board state tracking), TensorFlow (for Q-learning).
- **Timeline:**
    - **Weeks 1-2:** Design maze mechanics and ghost AI frameworks.
    - **Weeks 3-4:** Implement Minimax and A* for ghosts.
    - **Weeks 5-6:** Code pellet/board dynamics and GUI.
    - **Weeks 7-8:** Integrate AI, playtest, and refine heuristics.

---

6. References

- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
- Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.
- Online resources and game mechanics from: https://www.pygame.org/ and https://www.redblobgames.com/pathfinding/a-star/

Feasibility Adjustments

To simplify without sacrificing core innovation:

- Simplify the Grid: Use a fixed square maze initially, then add hexagonal modularity later.
- Reduce AI Complexity:
- Focus on one ghost behavior first (e.g., A* for Inky), then add Minimax/RL.
- Use rule-based systems for Pinky instead of RL (e.g., probabilistic chasing).
- Limit Players/Ghosts: Start with 2 players and 1-2 ghosts to streamline testing.
- Phase Out Power-Ups: Implement core pellet mechanics first, then add power-ups incrementally.