Project Title:
Snakes and Ladders With Prediction Challenge

Submitted By:
● Ali Bilal Siddiqui (22k-4227)
● Huzaifa Ayaz (22k-4583)
● Uzair Ahmed (22k-4189)

Course:
AI

Instructor:
Miss Alishba Subhania

---

1. Executive Summary

Project Overview:
This project presents a modified version of the classic Snakes and Ladders game that incorporates AI prediction capabilities. The core innovation is the integration of a prediction challenge where both the player and AI predict the next dice outcome. Correct predictions grant bonus options, influencing the gameplay. The AI uses a frequency-based approach to forecast the next dice roll, enhancing interactivity and strategic depth.

---

2. Introduction

Background:
Snakes and Ladders is a traditional board game designed for 2+ players, where players move across a grid based on dice rolls, climbing ladders and avoiding snakes. It is a game of chance, traditionally without strategy. This project modifies the classic format by integrating predictive AI mechanics and decision-driven gameplay, creating an enriched user experience.

Objectives of the Project:

- To implement an AI model that predicts future dice rolls.
- To allow both AI and player to make predictions for bonus opportunities.

- To enable strategic decision-making based on correct predictions.
- To visualize and interact with the game using Python's Pygame library.

---

3. Game Description

Original Game Rules:
Players roll a dice to move forward on a board filled with ladders (which advance the player) and snakes (which push the player back). The first player to reach the end wins.

Innovations and Modifications:

- Added prediction mechanism before every turn.
- AI uses a frequency counter to predict dice rolls based on history.
- Correct predictions trigger bonus options like doubling moves, skipping opponent's turn, or gaining points.
- Option for AI to neutralize snake effects using bonus points.
- Visual feedback for AI predictions and player choices.

---

4. AI Approach and Methodology

AI Techniques Used:

- Frequency-based prediction using Python's `collections.Counter` to predict the most frequent past dice value.

Algorithm and Heuristic Design:

- AI predicts the next dice roll based on historical frequency.
- If the AI prediction is correct, it chooses a bonus option based on strategic game state (e.g., advancing vs. blocking opponent).

AI Performance Evaluation:

- Performance was assessed via accuracy of predictions, usage of bonus opportunities, and the ability to avoid setbacks using prediction advantages.

---

5. Game Mechanics and Rules

Modified Game Rules:

- Players must predict the outcome of their own dice roll.
- Correct predictions allow choices: double the move, skip opponent's turn, or gain 10 bonus points.
- Bonus points (20) can be used to neutralize snake penalties.

Turn-based Mechanics:

- Red (AI) and Blue (Player) alternate turns.
- Dice is rolled after prediction.
- The game continues until one player reaches the top-left cell.

Winning Conditions:

- First player to reach the (162, 6) coordinate (top-left of the board) is declared the winner.

---

6. Implementation and Development

Development Process:

- Developed the game loop and GUI using Pygame.
- Integrated prediction interface and AI logic.
- Linked prediction correctness with reward options.

Programming Languages and Tools:

- **Programming Language:** Python
- **Libraries:** Pygame, collections
- **Tools:** Pygame for GUI, Python IDLE for scripting

Challenges Encountered:

- Designing prediction interface using Pygame components.
- Integrating real-time AI decisions into the GUI loop.
- Handling snake/ladder logic alongside dynamic reward systems.

---

7. Team Contributions

- **Ali Bilal:** Designed the prediction interface and implemented AI logic.
- **Huzaifa Ayaz:** Integrated bonus systems and user interaction handling.
- **Uzair Ahmed:** Focused on testing, debugging and enhancing game visuals.

---

8. Results and Discussion

AI Performance:

- The AI successfully predicted the correct dice outcome in ~25-35% of cases depending on historical variability.
- Prediction-driven decisions added a strategic layer, making gameplay less reliant on random chance.
- The AI strategically chose between doubling, skipping, or gaining points to improve its position or hinder the opponent.

9. References

- Python Official Documentation
- Pygame Documentation
- StackOverflow discussions on event handling and AI heuristics
- Online articles on turn-based game logic and AI in board games