<u>BRAINY LADDERS</u>

Submitted By: Hamza Salam(22k-4698), Fatima Salman(22k-4743) and Umer Ahmed(22k-4684)

Course: AI

Instructor: Miss Mehak Mazhar

Submission Date: 3-08-2025

1. Project Overview

This project modifies the traditional board game of Snakes and Ladders by introducing an interactive puzzle-solving mechanism. Players must solve a puzzle or a small riddle to advance after rolling the dice, adding an element of strategy and cognitive engagement.

Objective:

The main goal of this project is to enhance the traditional Snakes and Ladders game by integrating AI-based puzzle generation and solving mechanisms. This will encourage critical thinking, making the game more than just luck-based. Additionally, AI will be implemented to play against human players or simulate automated gameplay.

2. Game Description

Original Game Background:

Snakes and Ladders is a classic board game played on a numbered grid. Players roll a dice to move forward. Landing on a ladder allows the player to climb, while landing on a snake sends the player backward. The goal is to reach the final square first.

Innovations Introduced:

- Puzzle Mechanic: After rolling the dice, players must solve a small puzzle or riddle to proceed with their move. If they fail, they lose the turn.
- Adaptive Difficulty: AI will adjust the difficulty of puzzles based on the player's performance.
- AI Opponent: AI-controlled players will use predefined strategies for dice rolling and puzzle solving.
- Multiplayer Mode: Players can compete against each other or AI opponents.

Impact on Gameplay Complexity and Strategy:

- The game introduces an intellectual challenge rather than relying purely on luck.
- Strategic planning will be required as puzzles vary in difficulty.
- AI-driven puzzles will ensure engaging and dynamic gameplay.

3. AI Approach and Methodology

In our game, we use a concept called Bayesian win estimate to determine how easy or hard the AI's riddle should be, based on its current position compared to the human player. This is achieved using a sigmoid function, mathematically written as:
1 / (1 + math.exp(human_pos - ai_pos))

How It Works:

- ai_pos: The current position of the AI player on the board.
- human_pos: The current position of the human player.
- human_pos - ai_pos: Measures whether the AI is ahead or behind.
- The formula converts this positional difference into a probability-like value between 0 and 1.

Interpreting the Output:

- If AI is far behind (human is much ahead), the result is close to 0.
- If AI is far ahead, the result is close to 1.
- If both players are near each other, the result is around 0.5.

Purpose and Benefit:

This adaptive logic ensures **dynamic difficulty balancing**. It prevents the AI from being too dominant when ahead and gives it a chance to recover when behind. The result is a more **engaging and fair gameplay experience** for the human player.

4. Game Rules and Mechanics

Modified Rules:

- Players roll the dice as usual.
- Before moving, they must solve a puzzle within a time limit.
- Correct answers allow the move; incorrect answers forfeit the turn.

Winning Conditions:

- The first player to reach the final square wins.
- AI players use strategic puzzle-solving techniques to compete with humans.

Turn Sequence:

1. Player rolls the dice.
2. A puzzle or riddle appears.
3. Player attempts to solve it within the time limit.
4. If successful, the player moves forward; otherwise, the turn is lost.

5. Implementation Plan

Programming Language: Python **Libraries**

and Tools:

- Pygame: For GUI and game board representation.
- NumPy: For data handling.
- TensorFlow/NLTK: For NLP-based riddle generation and solving.

Milestones and Timeline:

- Week 1-2: Game rule finalization and puzzle database creation.
- Week 3-4: AI development for puzzle adaptation and difficulty scaling.
- Week 5-6: Implementation of board mechanics and AI player strategies.
- Week 7: Integration of AI puzzle system with gameplay.
- Week 8: Final testing, bug fixes, and report documentation.