Project Title: A 5-Player Ludo Variant

Submitted By: Haiqa Khan
Course: AI
Instructor: Abdullah Yaqoob
Submission Date: 10-3-25

---

1. Project Overview

Project Topic:

We propose an innovative version of Ludo, a 5-player variant of the traditional game. Unlike standard Ludo, which accommodates four players, our version introduces a **pentagonal board design**, **three tokens per player**, and **special power-up tiles** to enhance gameplay complexity and strategy.

Objective:

The main goal of this project is to develop an AI agent that can strategically play this 5-player Ludo using **Minimax with multi-player adaptation**. We will incorporate **heuristics for decision-making**, ensuring optimal moves and strategies based on board conditions. The project also aims to introduce a visually interactive and engaging game experience.

2. Game Description

Original Game Background:

Ludo is a classic turn-based board game where players roll dice to move tokens from their starting zone to the center of the board. Players must navigate their pieces through a **fixed path while avoiding being captured by opponents**. The first player to get all tokens home wins.

Innovations Introduced:

1. **5 Players Instead of 4:**

    - The board is redesigned into a **pentagonal layout**, each player having a unique home base and path.
    - Each player has **3 tokens instead of 4**, making the game faster-paced.

2. **Power-Up Tiles:**

    - **Double Roll:** The player gets to roll the dice twice.
    - **Safe Zone:** Tokens on these tiles cannot be captured.

3. **New Winning Condition:**

    - Instead of all 3 tokens reaching home, a player **wins by securing at least 2 tokens in the center before others**.
    - Players can also **block** opponents from reaching their home.

4. **Faster Gameplay Mechanics:**

    - Players must **move at least one token per turn** if possible, reducing idle turns.
    - The dice mechanism will include a fair-randomization model to prevent bias.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm (Multi-Player Variant)**: Modified to handle 5 players.
- **Alpha-Beta Pruning:** To optimize decision-making and speed up calculations.
- **Basic Heuristics:** AI will evaluate board states using strategies such as:
    - **Prioritizing token safety:** Avoiding opponent capture zones.
    - **Maximizing mobility:** Moving the most advantageous token per turn.
    - **Using power-ups optimally:** Deciding when to swap or use extra rolls.

Complexity Analysis:

- Traditional Ludo AI uses **O(b^d) complexity**, where b is the number of legal moves, and d is the depth.
- Due to 5 players, we optimize with **pruning techniques to reduce redundant calculations**.
- The game remains computationally manageable since moves are discrete and deterministic.

4. Game Rules and Mechanics

Modified Rules:

1. Each player has **3 tokens** instead of 4.
2. The board is **pentagonal**, with 5 home bases and movement paths.
3. New **power-up tiles** grant strategic advantages.
4. The first player to **secure 2 tokens in the center wins**.

Winning Conditions:

- A player wins by getting **2 out of 3 tokens into the home zone first**.
- If multiple players reach the home zone simultaneously, the one with more remaining active tokens wins.

Turn Sequence:

- Players roll a dice (1-6) and **must move if possible**.
- If a token lands on a power-up tile, the player gets an immediate effect.
- If an opponent's token is on a non-safe tile, it can be captured and sent back to start.

5. Implementation Plan

Programming Language:

- Python

Libraries and Tools:

- **Pygame** (for GUI and board visualization)
- **NumPy** (for handling AI computations and probabilities)
- **AI Libraries:** Implementations using standard AI techniques (Minimax, Alpha-Beta Pruning)

Milestones and Timeline:

- **Week 1-2:** Game design, board layout, and rule finalization.
- **Week 3-4:** Implement basic Ludo movement mechanics.
- **Week 5-6:** Develop AI strategy using Minimax and heuristic functions.
- **Week 7:** Integrate AI into gameplay, testing interactions.
- **Week 8:** Final bug fixes, performance optimizations, and report preparation.