Project Report

Title: Ludo Mix: AI-Enhanced Board Game with Safe Zone Mechanics

Submitted By:

1. Syed Zain Haider [22k-4281]  
2. Ali Jaffry [22k-4430]  
3. Areeb Ahmed [22k-4623]

Course: Artificial Inteliigence

Instructor: Sir Shafique ur Rehman

Date of Submission: 11 May 2025

# 1. Executive Summary

This project presents a modified version of the traditional Ludo game, developed in Python. The core innovation lies in integrating a rule-based AI system that allows the computer to play strategically, leveraging safe zones, base exits, and token prioritization. The game features dynamic board rendering, multiple player support (2–4), and a power-up mechanism for landing on safe squares.

# 2. Introduction

Background:  
Ludo is a classic board game played by 2–4 players, each trying to move their tokens from base to home. This project introduces digital gameplay, allowing both human and computer-controlled players, and enhances the experience with added game mechanics like power-ups and safe zones.

Objectives of the Project:  
- Digitally simulate the Ludo board game using Python.  
- Implement an AI player with basic strategic decision-making.  
- Introduce new game rules like safe square bonuses.  
- Create a visually understandable board display in a text-based interface.

# 3. Game Description

Original Game Rules:  
- Each player starts with four tokens in the base.  
- Tokens move based on a dice roll (1–6).  
- A roll of 6 allows a token to leave the base.  
- Tokens must reach the home square with exact roll value.

Innovations and Modifications:  
- Added safe squares where tokens cannot be captured.  
- Landing on a safe square rewards an extra turn (“Ludo Mix Powerup”).  
- AI players use a strategy to select tokens intelligently.  
- Enhanced CLI-based visual board display.

# 4. AI Approach and Methodology

AI Techniques Used:  
The computer player uses a rule-based decision engine. While not using machine learning, the AI prioritizes moves based on logical conditions such as escape from base, safety, distance covered, and goal reachability.

Algorithm and Heuristic Design:  
- Priority 1: Exit base if possible with a roll of 6.  
- Priority 2: Move token directly into the home if possible.  
- Priority 3: Land on safe squares when possible.  
- Priority 4: Advance the furthest moved token.  
- Fallback: Choose a token randomly.

AI Performance Evaluation:  
The AI demonstrates reasonable decision-making, often outperforming random play strategies. Its strength lies in survival and steady progress rather than aggressive tactics.

# 5. Game Mechanics and Rules

Modified Game Rules:  
- Safe squares include positions like 1, 9, 14, etc.  
- Landing on a safe square rewards an extra turn.  
- Tokens must reach the home with exact movement.  
- Captures are disabled on safe squares.

Turn-based Mechanics:  
Players take turns rolling a die and choosing a token to move. AI decisions are automated. A turn may repeat if certain bonus conditions are met.

Winning Conditions:  
A player wins when all their tokens reach the home square. The first to do so is declared the winner.

# 6. Implementation and Development

Development Process:  
The game logic was developed iteratively in Python. Functions were created for display, dice rolling, move validation, AI logic, and user interaction.

Programming Languages and Tools:  
- Language: Python  
- Libraries: random, time, sys, os  
- Tool: Visual Studio Code

Challenges Encountered:  
- Board visualization using characters in a terminal was complex.  
- Ensuring valid token movement and home constraints required detailed condition checks.  
- Managing AI decisions in edge cases (e.g., blocked paths) needed fine-tuning.

# 7. Team Contributions

- Syed Zain Haider: Handled game mechanics including turn handling, token movement rules, and win condition checks.  
- Ali Jaffry: Developed the board layout system and worked on the user input/interaction flow.  
- Areeb Ahmed: Implemented the AI decision logic and designed the safe square mechanism.

# 8. Results and Discussion

AI Performance:  
- The AI avoided risky moves and used safe zones effectively.  
- Win rate vs. humans averaged around 60% over multiple sessions.  
- Average decision time for AI: ~0.5 seconds per move.

# 9. References

- Python Official Documentation: https://docs.python.org  
- Ludo Rules – Wikipedia: https://en.wikipedia.org/wiki/Ludo\_(board\_game)  
- Stack Overflow Discussions on random and struct modules