National University of Computer and Emerging Sciences

Project Proposal

Project Title:
Hexagonal Ludo – An Innovative Multi-Player Board Game Variant

Submitted By:
Haris Khan [22k-4185]

Group members:

Haris Khan [22k-4185]

Muhammad Abdul Rehman [22k-4291]

Talha Idress [22k-4292]

Course:
Artificial Intelligence

Instructor:
Miss Alina Arshad

Miss Syeda Ravia Ejaz

Submission Date:
3/22/2025

---

1. Project Overview

- **Project Topic:**
  Develop a Hexagonal Ludo variant that transforms the classic Ludo game by changing the board layout to a hexagonal grid. In this version, the board has six home bases instead of four, each player starts with three tokens rather than four, and the dice roll range is increased from 1–6 to 1–8. This game will support multi-player interactions (up to six players) with unique strategic dynamics.
- **Objective:**
  The main goal is to develop a strategic AI for the modified Hexagonal Ludo using a modified minimax algorithm tailored for multi-player settings. The project will focus on designing evaluation heuristics that assess token safety, progress toward the finish, and risk-reward tradeoffs while handling the increased variability introduced by the 8-sided dice. Reinforcement learning techniques will also be explored to further enhance AI performance through self-play simulations.

---

2. Game Description

- **Original Game Background:**
  Traditional Ludo is a board game where each player has four tokens and a square board with four home bases. Players take turns rolling a die (typically 1–6) and move their tokens accordingly, with the objective of moving all tokens from start to finish while sending opponents' tokens back to the starting area.
- **Innovations Introduced:**
  - **Board Structure:**
    The game board is reimagined as a hexagonal grid with six home bases, altering spatial dynamics and offering new routes and strategic positions.
  - **Token Count:**
    Each player starts with three tokens, changing the balance between risk and reward as players manage fewer tokens.
  - **Dice Range:**
    Dice rolls now range from 1 to 8, introducing greater variability. This change creates new tactical opportunities and challenges, such as faster token progression or higher risk of overshooting.

- **Impact on Gameplay:**
  These modifications yield a more complex and dynamic environment where players must continuously evaluate multiple paths, risk exposure, and opponent positioning. The increased number of players (up to six) further deepens strategic complexity as interactions among tokens become more unpredictable.

---

3. AI Approach and Methodology

- **AI Techniques to be Used:**
  - **Modified Minimax Algorithm:**
    Implement a minimax-based decision framework that is adapted to accommodate multi-player interactions. This may involve integrating concepts from game theory or simplifying the decision tree to focus on key tactical interactions.
  - **Alpha-Beta Pruning:**
    Utilize pruning techniques to optimize the minimax search by reducing the number of nodes that need evaluation, ensuring efficient decision-making.
  - **Reinforcement Learning (Optional):**
    Experiment with reinforcement learning to allow the AI to improve over time through self-play, discovering optimal strategies in the face of increased variability.
  - **Other Techniques:**
    Consider leveraging Monte Carlo Tree Search (MCTS) or neural network approaches to handle the probabilistic outcomes resulting from the 8-sided dice and to support broader exploration of the decision space.
- **Heuristic Design:**
  Develop evaluation functions that assess:
  - **Token Safety:** Measures the risk of a token being captured.
  - **Progress Toward the Finish:** Evaluates how close each token is to reaching its goal.
  - **Blocking Potential:** Assesses the ability to obstruct opponents' movements.

- **Risk-Reward Tradeoff:** Balances aggressive strategies against safer, more conservative plays on a board with diverse pathways.
- **Complexity Analysis:**
  - **Time Complexity:**
    The decision-making algorithm will have an exponential time complexity relative to the branching factor and search depth. With modifications for a multi-player game and an 8-sided dice, the effective branching factor increases. Optimization techniques like alpha-beta pruning and MCTS can help mitigate these effects.
  - **Implementation Challenges:**
    - Managing the larger state space due to the hexagonal board structure and increased movement options.
    - Balancing multi-player dynamics where the AI must consider the progress and potential actions of up to five opponents.
    - Efficiently integrating probabilistic outcomes from the expanded dice range into the AI's evaluation functions.

---

4. Game Rules and Mechanics

- **Modified Rules:**
  - **Board Layout:**
    The game board is arranged as a hexagonal grid with six home bases.
  - **Token Count:**
    Each player begins with three tokens.
  - **Dice Mechanics:**
    Players roll an 8-sided die (values 1–8) to determine movement.
  - **Movement:**
    Tokens move along predefined paths modeled as nodes on a graph. Players choose which token to move based on the dice outcome.

- **Capturing:**
  If a token lands on a space occupied by an opponent, that opponent's token is sent back to the starting area.

- **Winning Conditions:**
  A player wins by successfully moving all three of their tokens from the starting area to their designated home base. Additional variations may reward players with extra points for strategic plays such as blocking opponents.

- **Turn Sequence:**

  0. **Dice Roll:** The active player rolls the 8-sided die.
  1. **Token Movement:** The player selects one token to move the number of spaces indicated by the dice.
  2. **Capture/Block Check:** If the token lands on a space occupied by an opponent, capture mechanics are triggered.
  3. **Turn End:** Once the token has moved, the turn passes to the next player in the predetermined sequence or based on an initiative system.

---

5. Implementation Plan

- **Development Environment:**
  The project will be developed using a game engine that supports both visual scripting and text-based scripting for rapid prototyping and performance tuning.

- **Programming Language:**

  C++ or Maybe Python

- **Tools and Libraries:**
  The implementation will make use of:

  - Integrated development and debugging tools provided by the game engine.
  - Graph modeling utilities to represent the hexagonal board as a network of nodes.
  - AI libraries or modules that support decision-making algorithms, including minimax, alpha-beta pruning, and reinforcement learning techniques.
  - Version control systems for collaboration and code management.

- **Milestones and Timeline:**
  - **Week 1-2:**
    Finalize game design, rule modifications, and board layout. Model the hexagonal grid as a graph.
  - **Week 3-4:**
    Develop the AI strategy by implementing the modified minimax algorithm, incorporating alpha-beta pruning, and designing evaluation heuristics.
  - **Week 5-6:**
    Code and test core game mechanics including token movement, capturing, and turn sequencing. Develop UI elements to display dice rolls, token positions, and turn indicators.
  - **Week 7:**
    Integrate the AI with the game mechanics and conduct multi-player simulation tests. Refine heuristics and, if applicable, integrate reinforcement learning components.
  - **Week 8:**
    Final testing, performance optimization, playtesting sessions, and preparation of the final project report.

---

6. References

Nil..

The End