Project Title: HexAI – AI Strategy for Hex Game

Submitted By: Filza Salman (22k-5011), [Yasaal Khan] (22k-6004)

Course: AI

Instructor: Talha Shahid

Submission Date: 4/20/2025

1. Project Overview

Project Topic:

HexAI is an artificial intelligence project centered on the strategic two-player board game Hex. The innovation involves building a smart AI using a combination of Minimax with Alpha-Beta Pruning, Monte Carlo Tree Search (MCTS), and reinforcement learning techniques to simulate advanced gameplay and evolve strategies through self-play.

Objective:

The main objective is to develop a capable AI agent for Hex that can learn, adapt, and compete with human or AI players. The AI aims to exhibit optimal decision-making, strategic depth, and adaptive behavior through learning mechanisms.

2. Game Description

Original Game Background:

Hex is a connection-based game played on a hexagonal grid. Two players attempt to connect their respective opposite sides of the board using continuous chains of their pieces. Due to the Hex Theorem, a win is guaranteed for one of the players, ensuring the game has no draw condition.

Innovations Introduced:

- Implementation of **Minimax with Alpha-Beta Pruning** to minimize unnecessary exploration.
- Use of **Monte Carlo Tree Search (MCTS)** with tuning for intelligent probabilistic decision-making.
- Application of **Dijkstra's algorithm-based heuristics** for shortest-path evaluation.
- Use of **Transposition Tables** for reducing repeated state evaluation.
- Difficulty scaling and adaptive strategies based on player performance.

These changes aim to make the AI more flexible, intelligent, and scalable across different skill levels.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm:** Optimized for two-player zero-sum gameplay.
- **Alpha-Beta Pruning:** Reduces unnecessary branches, improving efficiency.
- **Monte Carlo Tree Search (MCTS):** Simulates multiple outcomes for better move prediction.
- **Reinforcement Learning:** AI will learn strategies via self-play and policy/value estimation.

Heuristic Design:

- Influence mapping and control zones over the board.
- Path evaluation toward the winning edge using graph algorithms like Dijkstra.
- Blocking potential of opponent moves.
- State evaluation functions tuned via reinforcement learning experience.

Complexity Analysis:

- Minimax complexity: O(b^d), reduced via Alpha-Beta.
- MCTS: Depends on simulations per move (generally large but manageable).
- Reinforcement Learning: Complexity varies by training episodes and convergence time.

4. Game Rules and Mechanics

Modified Rules:

- Standard Hex rules are used.
- AI supports various difficulty levels and optionally uses different AI strategies.

Winning Conditions:

- A player wins by forming a continuous chain of their colored pieces between their two designated board edges.

Turn Sequence:

- Players take turns alternately.
- AI will choose moves based on the selected algorithm (Minimax, MCTS, or learned policy).

5. Implementation Plan

Programming Language: Python

Libraries and Tools:

- **Pygame** – for visual interface (optional).
- **NumPy** – for board state representation and calculations.
- **Scikit-learn** – for classic machine learning (optional).
- **TensorFlow/PyTorch** – for reinforcement learning module.

Milestones and Timeline:

- **Week 1-2:** Implement base game mechanics and UI.
- **Week 3-4:** Add AI logic using Minimax and heuristics.
- **Week 5-6:** Integrate MCTS and reinforcement learning models.
- **Week 7:** Test all components, optimize AI.
- **Week 8:** Final tests, UI polish, and report writing.

6. References

- Wikipedia: Hex (board game)
- Hex Strategy Books and Research Papers
- MCTS and Reinforcement Learning resources on Medium, ArXiv, and official docs
- NetworkX and Red Blob Games for pathfinding visualization