Project Report

Project Title: HexAI - AI Strategy for Hex Game

Submitted By: Bisma
Course: AI
Instructor: Alishba Subhani
Submission Date: 3/10/2025

1. Executive Summary

This project, titled HexAI, focuses on developing an intelligent agent capable of playing the strategic board game Hex. The aim is to design and implement AI agents using Minimax with Alpha-Beta Pruning, Monte Carlo Tree Search (MCTS), and Reinforcement Learning. These methods enhance decision-making and enable the AI to adapt and learn strategies over time.

2. Introduction

Background:

Hex is a two-player abstract strategy game where players aim to form a connected path across opposite sides of a hexagonal board. Traditionally, the game involves intuitive and tactical thinking. This project introduces an AI-enhanced version of Hex where the opponent can be either a human or another AI. The AI utilizes advanced decision-making algorithms to simulate strategic thinking.

Objectives of the Project:

- Develop AI agents capable of playing Hex using strategic algorithms.
- Implement Minimax with Alpha-Beta Pruning for efficient move searching.
- Integrate Monte Carlo Tree Search for probabilistic evaluation.
- Apply Reinforcement Learning to enable self-improvement over time.
- Evaluate AI performance against human players and other AI configurations.

3. Game Description

Original Game Rules:

Hex is played on an N×N hexagonal board, where each player is assigned either red or blue. Players take turns placing a stone of their color on empty cells, attempting to connect their respective sides of the board. There are no ties; one player will always win due to the Hex Theorem.

Innovations and Modifications:

- AI algorithms including Minimax with Alpha-Beta Pruning, MCTS, and Reinforcement Learning are used.
- Heuristic evaluation functions based on Dijkstra’s algorithm assess potential winning paths.
- Transposition Tables are used to avoid redundant state evaluations and optimize performance.
- Adjustable AI difficulty and strategy tuning based on game history.

4. AI Approach and Methodology

AI Techniques Used:

- Minimax Algorithm: Evaluates game tree to select the optimal move.
- Alpha-Beta Pruning: Optimizes Minimax by ignoring suboptimal branches.
- Monte Carlo Tree Search (MCTS): Uses random simulations to guide decisions in high branching factor scenarios.
- Reinforcement Learning: The AI learns strategies from repeated gameplay and adapts based on rewards.

Algorithm and Heuristic Design:

- **Heuristics**:
  - Influence mapping (control over central and border regions)
  - Path potential evaluation (proximity to victory)
  - Opponent blocking strategies
- **Evaluation Metrics**:
  - Move quality and prediction accuracy
  - Win probability based on board state
  - Learning rate over multiple iterations

AI Performance Evaluation:

- Win rate comparison against human players and other AI variants
- Average decision time per move
- Analysis of strategy improvement through reinforcement learning over multiple games

5. Game Mechanics and Rules

Modified Game Rules:

- The game follows traditional Hex rules.
- AI strategies vary depending on the selected difficulty and algorithm.
- AI players may utilize different decision strategies (Minimax, MCTS, RL).

Turn-based Mechanics:

- Two players (AI vs. AI or AI vs. Human) alternate turns.

- Each turn consists of placing a colored piece on an empty cell.
- The AI evaluates multiple future states before selecting the move.

Winning Conditions:

- A player wins by connecting their assigned edges with an unbroken chain of their color.

6. Implementation and Development

Development Process:

- Designed Hex board logic and GUI interface.
- Implemented Minimax and Alpha-Beta pruning.
- Integrated MCTS for probabilistic AI decision-making.
- Reinforcement Learning modules were trained using self-play.
- Performance evaluations were conducted through simulated matches.

Programming Languages and Tools:

- Programming Language: Python
- Libraries:
    - Pygame – GUI and board rendering
    - NumPy – Data operations and matrix handling
    - Scikit-learn – Utility functions for data handling
    - TensorFlow / PyTorch – Reinforcement learning framework
- **Tools**:
    - Jupyter Notebooks – Testing and visualization

Challenges Encountered:

- High computational cost of MCTS and RL in real-time gameplay
- Designing effective heuristic functions for Hex path evaluation
- Balancing exploration and exploitation in reinforcement learning
- Synchronizing GUI with AI decision delays

7. Team Contributions

Team Members and Responsibilities:

- **Bisma**:
    - Implemented Minimax algorithm with Alpha-Beta Pruning
    - Developed heuristic evaluation functions
    - Integrated Transposition Tables
    - Implemented Monte Carlo Tree Search and Reinforcement Learning models
    - Designed game board and GUI using Pygame
    - Conducted AI performance testing and final integration

8. Results and Discussion

AI Performance:

- Minimax with Alpha-Beta Pruning achieved high accuracy with fast decisions (~0.5s per move).
- MCTS provided better results in mid-to-late game states with ~65% win rate in simulations.
- Reinforcement Learning showed gradual performance improvement over 1000+ training games.

9. References

- Wikipedia - [Hex (board game)]
- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*