Monte Carlo Tree Search Final Project Report

Team Members: Shuyan Bian, Tianrui Li, Jianan Ni

INFO 6205 Spring 2025

Team Members: Shuyan Bian, Tianrui Li, Jianan Ni

Date: April 20, 2025

Executive Summary

This project implements a complete Monte Carlo Tree Search (MCTS) framework capable of playing two turn-based games: the classic Tic-Tac-Toe and Connect Four (screen-style four-in-a-row). The primary objective was to design and validate an extensible MCTS algorithm under a common interface, supporting easy game integration. The project delivers two playable AI agents (via CLI) and a suite of reproducible unit tests.

Game 1: Tic-Tac-Toe

Features Implemented

- Full implementation of TicTacToe with internal TicTacToeState and TicTacToeMove.

- MCTS agent supports:

- Node expansion

- UCT-based child selection

- Random or "smart" simulation

- Win/draw-based backpropagation

Test Coverage

- TicTacToeTest.java: Validates state transitions and move generation.

- TicTacToeNodeTest.java: Verifies expansion and child logic.

- MCTSTest.java: Confirms the best move selection from root.

Observations

- MCTS reliably selects the center as first move.

- Early versions failed to block the opponent; enhanced simulation logic fixed that.

- Deterministic wins were confirmed using predefined board states.

Game 2: Connect Four

Game Description

Connect Four is a two-player turn-based game played on a 67 grid. Players alternate dropping discs into columns. The first to connect four in a rowhorizontally, vertically, or diagonallywins.

Implementation Highlights

- ConnectFour.java: Core state and game rules.

- ConnectFourNode.java: Tracks MCTS tree with statistics.

- MCTS.java: Shared logic used across both games.

Special Handling

- Simulation policy enhanced with:

- Center bias (column 3 preferred early)

- Defensive move selection (blocks opponent's win)

- Backpropagation based on outcome of simulation (2 = win, 1 = draw)

Test Coverage

- ConnectFourTest.java: Includes

- Column availability

- Vertical, horizontal, diagonal win detection

- Player alternation

- All tests now pass with properly controlled move sequences

Performance and Search Quality

Game	Simulation Count	Response Quality
Tic-Tac-Toe	1000-10000	Near-optimal
Connect Four	5000-20000	Solid for 4-5 plies

Run-time grows with iteration count, but Connect Four remains tractable due to smart simulation heuristics.

How to Run (via Maven)

```
mvn clean compile

mvn exec:java -Dexec.mainClass="tictactoe.TicTacToeHumanVsMCTS"

mvn exec:java -Dexec.mainClass="connectfour.ConnectFourHumanVsMCTS"
```