CSE 440 | Section 1 | Group 1 | Summer 2024

Tic-Tac-Toe AI

Group Members

- Arafat Anam Chowdhury - 203 1514 042
- Rajib Mohammed Nasim - 203 1050 042
- Khandker Shakila Rowshan Puspita - 211 1886 042

Abstract

This project presents the implementation of an AI agent that plays Tic-Tac-Toe optimally using the Minimax algorithm with alpha-beta pruning. The AI incorporates heuristic evaluations to enhance performance, strategically competing against human players. The solution integrates a graphical user interface (GUI) for intuitive interaction, delivering a robust platform to demonstrate advanced game tree search techniques in AI.

Introduction

Tic-Tac-Toe is a classic game that provides a simplified framework to demonstrate fundamental concepts in artificial intelligence (AI), particularly game tree search. This project develops an AI agent capable of playing optimally by evaluating potential moves using the Minimax algorithm and optimizing the decision-making process through alpha-beta pruning. The project also includes a user-friendly interface to engage users in interactive gameplay.

Objectives

- Implement an AI agent for Tic-Tac-Toe using the Minimax algorithm.
- Optimize AI performance with alpha-beta pruning to reduce computational complexity.
- Explore heuristic evaluation methods for effective decision-making.

Technologies and Tools

- **Programming Language:** Python
- **Libraries/Frameworks:** *Tkinter* for GUI, *math* for calculations
- **Algorithm:** Minimax with alpha-beta pruning

Methodology

- **Game Design (game.py)**
  - The game logic was structured in this file, incorporating rules for moves, determining winners, and identifying draw conditions.
- **AI Implementation (ai.py)**
  - The Minimax algorithm evaluates the game tree recursively, determining the optimal moves.
  - Alpha-beta pruning enhances the algorithm by eliminating unnecessary branches.
  - Heuristic evaluations assign scores to non-terminal game states for intermediate-depth evaluation.
- **GUI Development (main.py)**
  - The interface, built using Tkinter, provides a visual 3x3 game board. Buttons respond to user inputs, while AI decisions are displayed dynamically.

Features

- **AI Strategies:** Implements optimal gameplay through Minimax with alpha-beta pruning.
- **Heuristic Values:** Evaluate intermediate states for efficient decision-making.
- **Interactive UI:** Users play against AI via a responsive graphical interface.
- **Score Tracking:** Maintains records for wins, losses, and draws.

Results

The AI consistently performed optimally, avoiding errors in gameplay. Alpha-beta pruning reduced computation time significantly, enabling real-time interactions. The heuristic evaluation effectively limits search depth, balancing performance and computational efficiency.

Conclusion

This project successfully demonstrates the application of game tree search algorithms in AI through a simple yet effective Tic-Tac-Toe game. The AI achieves optimal gameplay by implementing Minimax with alpha-beta pruning and incorporating heuristics while providing users with an interactive platform.

References

- https://github.com/Cledersonbc/tic-tac-toe-minimax
- https://github.com/Tzachc/Tic-Tac-Toe-Artificial-intelligence-VS-Machine-learning
- https://github.com/kying18/tic-tac-toe
- https://github.com/GeorgeSeif/Tic-Tac-Toe-AI