

Unbeatable Tic-Tac-Toe: AI Opponent Using Minimax and Pygame GUI

Project 2

Jeremy Cleland

University of Michigan Dearborn

CIS-579-002(202420)

2024 March 09

1. Introduction

Tic-Tac-Toe is a classic game known for its simplicity. It has been used as a testbed for AI research in game theory and adversarial search. The aim of this project is to explore the implementation of an AI opponent using the Minimax algorithm, which will provide hands-on experience in developing intelligent game agents. As a graduate student specializing in AI, this project serves as a foundational step in understanding the concepts and applications of adversarial search algorithms.

2. Implementation

2.1 Game Environment

The Tic-Tac-Toe game was implemented in Python using the Pygame library for the graphical interface. The game logic, encapsulated in `tictactoe.py`, handles the interaction between the player and the AI opponent, ensuring a smooth and intuitive gaming experience.

2.2 Minimax Algorithm

The AI opponent uses the Minimax algorithm to make decisions, which involves evaluating the best moves based on the game's current state. This algorithm explores all possible game outcomes by recursively evaluating the game tree, assuming that both players make the most optimal moves. The utility function used in the algorithm plays a vital role in assessing the desirability of each game state.

2.3 Graphical User Interface

The Graphic User Interface (GUI) was built using Pygame to create an immersive and easy-to-use interface. Players interact with the game through mouse clicks, and the GUI provides visual feedback on the progress and outcomes of the game. Any difficulties encountered during GUI optimization, such as enhancing responsiveness and improving visual appeal, were addressed to ensure an optimal user experience.

2.4 Alpha-Beta Pruning

Alpha-beta pruning is a search algorithm that seeks to decrease the number of nodes evaluated by the minimax algorithm in its search tree. It is an optimization technique used to increase the efficiency of the minimax algorithm.

The concept of alpha-beta pruning is that it is unnecessary to evaluate every node in the game tree to find the optimal move. If a promising move is found along a certain path, then the algorithm doesn't need to evaluate the remaining paths, which are obviously worse.

In the context of this project, alpha-beta pruning was implemented to address the efficiency issues of the minimax algorithm, particularly in terms of execution time for larger game trees. By reducing the number of nodes evaluated, the algorithm's performance was significantly improved without compromising its decision-making quality.

The implementation of alpha-beta pruning involved setting two values, alpha and beta, representing the minimum score that the maximizing player is assured of and the maximum score that the minimizing player is assured of, respectively. During the traversal of the game tree, the branches that do not have the potential to improve the game's current state for the respective player are pruned.

The application of alpha-beta pruning in this project proved to be successful, as it allowed the AI to make optimal decisions more efficiently, thereby enhancing the overall gaming experience.

3. Experimentation and Results

3.1 Methodology

Gameplay sessions were conducted to ensure the game's functionality and the AI's effectiveness. In these sessions, the AI was programmed to play as both Player X (first player) and Player O (second player) against a human opponent. A total of ten games were recorded for each configuration.

3.2 Results

Game	Human (O)	AI (X)	Tie Game
1			X
2			
3		X	
4		X	
5		X	
6		X	
7		X	
8		X	
9		X	
10		X	
Wins	0	9	1
Game	Human (X)	AI (O)	Tie
1		X	
2			X
3		X	
4		X	
5			X
6		X	
7			X
8			X
9		X	
10		X	
Wins		6	4
Total Wins	0	15	5

Table 1: Summary of game outcomes

The tabular summary of game outcomes (Table 1) concisely overviews the AI's performance against human opponents. The results demonstrate the effectiveness of the Minimax algorithm in generating optimal moves, leading to a high win rate for the AI player.

3.3 Analysis

Further analysis of the game outcomes reveals the AI's ability to consistently make rational decisions, exploiting any suboptimal moves made by the human opponent. The AI performed as expected according to the theoretical predictions of the Minimax algorithm, showcasing its ability to play optimally in a perfect information game like Tic-Tac-Toe.

4. Discussion

4.1 Interpretation of Results

The experimental results strongly support the effectiveness of the Minimax algorithm in creating an intelligent AI opponent for Tic-Tac-Toe. The AI's win rate of 15 out of 20 games and ability to force draws against human players highlight the algorithm's capability to make optimal decisions based on the current game state.

4.2 Challenges and Solutions

One of the main challenges encountered during the project was the efficiency of the Minimax algorithm, particularly in terms of execution time for larger game trees. To address this, alpha-beta pruning was implemented to reduce the number of nodes evaluated, significantly improving the algorithm's performance without compromising its decision-making quality.

4.3 Limitations and Future Work

While the current implementation successfully demonstrates the application of the Minimax algorithm in Tic-Tac-Toe, there are opportunities for further enhancement. Potential areas for future work include:

- Incorporating machine learning techniques will enable AI to adapt its strategy based on the human player's moves.
- Extending the AI to handle more complex games with larger state spaces, such as Connect Four or Chess.
- Investigating other adversarial search algorithms, such as Monte Carlo Tree Search, and comparing their performance with Minimax.

5. Conclusion

The successful implementation of an AI opponent for Tic-Tac-Toe using the Minimax algorithm yielded promising experimental results, highlighting the efficacy of adversarial search in creating intelligent game agents. This project provided an excellent opportunity to apply AI concepts and algorithms to a practical problem, reinforcing the understanding of adversarial search and its significance in AI research. The insights gained from this project serve as a solid foundation for exploring more complex AI techniques and their potential applications in various domains. This project has paved the way for further research and development in

intelligent game agents and adversarial search algorithms by overcoming the challenges and considering the limitations.