

Phase 2 Mid-Term Status Report

Project Title: Developing a Tic Tac Toe AI Agent Using Minimax with Alpha-Beta Pruning.

Course: Introduction to Artificial Intelligence

Team Members: Priyanka Singh, Heli Sureja

Instructor: Prof. Shivanjali Khare

1. Research Question

In Tic Tac Toe, is an artificial intelligence agent routinely able to surpass random strategies or human players utilizing Minimax with Alpha-Beta pruning?

2. Introduction

Often used to illustrate fundamental AI ideas, Tic Tac Toe is a straightforward, deterministic, turn-based strategic game. The aim is to create an artificial intelligence agent capable of efficiently fighting against human or random opponent and that which performs optimum, therefore guaranteeing it never loses. Implementing the Minimax method with Alpha-Beta pruning helps the agent improve computational efficiency and replicate adversarial decision-making.

3. Related Work

1. **Minimax Algorithm in Theory of Games:** The Minimax page on Wikipedia offers a basic knowledge of how two-player, zero-sum games' decision-making could be modeled. It clarifies the theoretical underpinnings of the method, which seeks to reduce the potential loss under a worst-case scenario. <https://en.wikipedia.org/wiki/Minimax>
2. **Alpha-Beta Pruning (Geeks For Geeks):** This page provides a detailed definition of the Alpha-Beta pruning method, which removes branches in the game tree that have no bearing on the ultimate choice therefore optimizing the Minimax algorithm. This greatly lowers the evaluation count of nodes, therefore enhancing the efficiency of the method.
3. **Creating Tic Tac Toe with AI:** This practical implementation tutorial shows how to create a Tic Tac Toe AI using Minimax (with optional Alpha-Beta pruning). It provides a good guide for turning abstract ideas into functioning code.

Difference: Unlike many beginning-level projects, this one use response time analysis to measure the efficiency of the AI by comparing its performance against random players, therefore adopting a more scientific approach.

Development Outcomes:

- Python implements whole game logic and artificial intelligence decision-making.
- Pygame develops a graphical user interface (GUI), which permits interactive games between a human player and the artificial intelligence.
- The AI maximizes decision-making time by using the Minimax method improved with Alpha-Beta pruning.
- Additionally used as a basis for performance comparison is a random-move bot.

5. Challenges and Solutions

- The first Minimax algorithm implementation clearly caused delays during the AI's turn.
So, Alpha-Beta pruning will be integrated to drastically lower the evaluated node count, hence improving response times
- Using Pygame's rendering features will be added to make sure the display was only changed after legitimate moves.

6. Future work

- To Track and examine response times and win rates to create performance benchmarks and graphic displays.
- To Set up an experimental option for automated artificial intelligence against random bot matches.
- To Make a video demonstration and gather final reports with thorough analytics and conclusions.