ARTIFICIAL INTELLIGENCE LAB

TIC TAC TOE USING MINIMAX ALGORITHM WITH MEMOIZATION

Submitted by:

Abhinav Sharma 9915103035
Arjun Singh 9915103039

Under the supervision of
Ambalika Sarkar

Department of CSE/IT
Jaypee Institute of Information Technology University, Noida

November 2017

Statement of the problem

This is a simple Tic Tac Toe game on which the other player runs on AI technique implemented using MiniMax Algorithm with memoization.

There are two players in the game (X and O), A player wins if they claim all the fields in a row, column or diagonal. A game is over whenever a player wins or all fields are taken. A player can claim a field if it is not already taken.

The aim is to build a perfect AI so that the CPU is never defeated.

Reason/Motivation for the project:

We use Minimax Algo for our Tic Tac Toe game. The key to the Minimax algorithm is a back and forth between the two players, where the player whose "turn it is" desires to pick the move with the maximum score. In turn, the scores for each of the available moves are determined by the opposing player deciding which of its available moves has the minimum score. And the scores for the opposing players moves are again determined by the turn-taking player trying to maximize its score and so on all the way down the move tree to an end state. This algorithm is recursive, it flips back and forth between the players until a final score is found.

Objectives

- The main objective of the AI is to always win against the human player.
- The AI is designed such that it will never lose against the player it will either draw or win.
- We use 6-ply minimax tree for the AI.
- The objective for the AI is to win in the lowest possible no. of moves.
- The AI tries to win by claiming all the fields in the row, column or diagonal.
- Design a heuristic such that we can measure the relative win and loss for each player.
- Use memoization such that we do not compute the already computed branches of the tree.

Methodology

Introducing the computer as a player was challenging both from the logic and the interface perspective. We implemented the minimax recursive algorithm to allow the Computer to choose the best move, making it unbeatable. Its decision-making process is based on alternate moves, assigning a different score depending on whether the advantage is on the computer or the opponent side, so it was important to have a universal way for the game to handle player's input. Instead of calculating the value again and again we use memoization so as to use the already calculated value of the branch.

The AI will play perfectly unto its demise, is to take the "depth" or number of turns till the end of the game into account. Basically the AI should play perfectly, but prolong the game as much as possible. In order to achieve this, we will subtract the depth, that is the number of turns, or recursions, from the end game score, the more turns the lower the score, the fewer turns the higher the score.

we also took heuristics into account while preparing our tic tac toe game. If player X wins the game in that node the value of the heuristic decreases by 100, if we pass O and it wins it we increase the heuristic value by 100.

Software requirements:

- Any modern browser (eg. chrome)

References

- **https://developer.mozilla.org/enUS/docs/Learn/JavaScript/Objects**
- **javascriptissexy.com/oop-in-javascript-what-you-need-to-know/**
- **neverstopbuilding.com/minimax**
- **www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/**
- **https://www.interviewcake.com/concept/java/memoization**