AI Text-based TicTacToe

Fady Ibrahim Mahmoud Mossad

Mostafa Hatem Said Mostafa

Yulius Mohareb

January 2022

0.1 Abstract

0.2 Introduction

Nowadays many people suffer from loneliness and rely on digital games to fill their time with something entertaining and sometimes even useful. In the past 10 years, online mini games have developed greatly and are visited by millions of users throughout the day. Examples of such games which have gone viral are Flappy Bird and Among Us.

Artificial intelligence has been increasingly integrated in the past few years into more and more mini games to simulate the presence of a human opponent. This has been demonstrated in more and more mini games such as the very own windows chess game.

Therefore we have developed a proof of concept Tic Tac Toe game to demonstrate the utilization of the Minimax algorithm integrated in a text-based implementation of the simple Tic Tac Toe game.

0.3 Problem

Console games at first glance may look basic, but what is behind may be wrapping a powerful intelligent algorithm. Lately an increasingly number of visually undeveloped games have attracted a huge number of people because of their immersive experience.

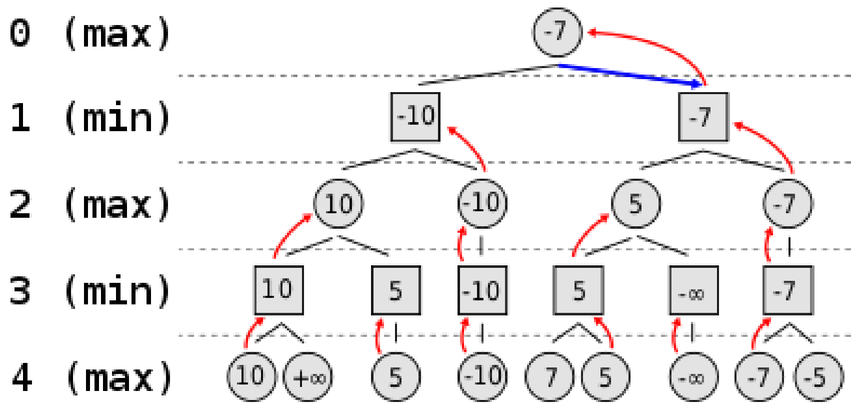
This demonstration shows how a simple command line game can be immersive and challenging for users to win.

0.4 Methodology

In an n-player game, usually a two-player game, a minimax algorithm is a recursive algorithm for determining the next move. Each position or state of the game has a monetary value. A position evaluation function is used to calculate this value, which reflects how good it would be for a player to attain that position. The player then makes a move that maximizes the position's minimal value as a result of the opponent's probable next moves. When it's A's turn to move, A assigns a numerical value to each of their legal moves.

A proposed allocation strategy is to assign a specific win to A as +1 and a specific win to B as 1. This gives rise to John Horton Conway's combinatorial game theory. If the result of a move is an immediate win for A, it is assigned positive infinity; if it is an immediate win for B, it is assigned negative infinity. Any other move's value to A is the sum of the values coming from each of B's conceivable responses. As a result, A is referred to as the maximizing player, whereas B is referred to as the minimizing player, thus the term minimax algorithm.

The algorithm can be thought of as a game tree's nodes being explored. The average number of children of each node is the tree's effective branching factor (i.e., the average number of legal moves in a position). The number of nodes to be explored usually grows inexorably as the number of plies increases (it is less than exponential if evaluating forced moves or repeated



positions). As a result, the number of nodes to be explored for a game analysis is about the branching factor increased to the power of the number of plies. As a result, employing the minimax method to fully analyze games like chess is impracticable.

0.5 Approach

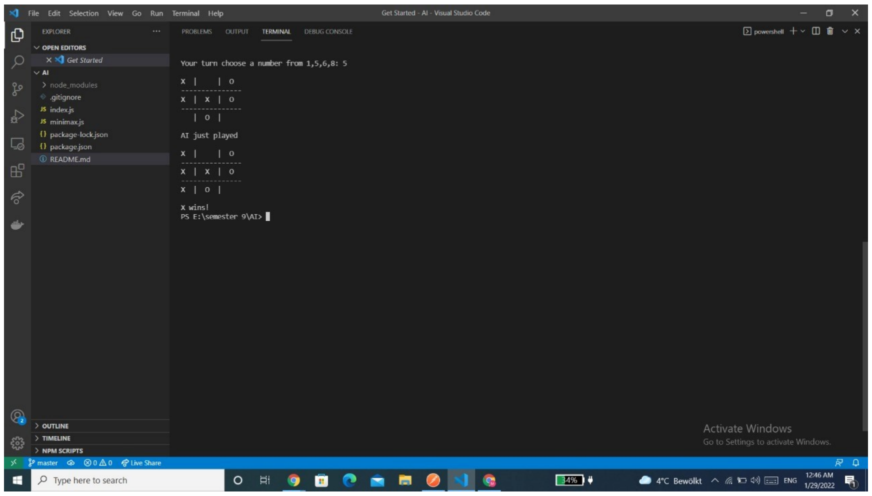
We used the Minimax algorithm over the standard Tic Tac Toe game so that the nodes are the possible game states, starting with an empty board. The minimizing game movement is always considered to be inflicted by the human player, which the maximizing game movement is considered to be inflicted by the AI agent.

After each player movement the tree is expanded further to the next levels to explore the best move option. The agent assumes the worst case scenario of the player doing the best move in their interest to win.

As easy as this algorithm might appear with a maximum of 9 movements on the Tic Tac Toe board, it could get much more complicated with larger boards, hence a larger branching factor and more possible states.

The game has been implemented as a command-line-based game for simplicity, however it can be modified into a full GUI game with a proper UI with a few lines of code.

We used JavaScript for the implementation, with a main class for the game and an auxiliary class for the Minimax algorithm. HTML and CSS can further be used to give the game a proper UI but this was out of scope of our goal, which is to demonstrate how AI can be used to develop a single player intriguing mini game.

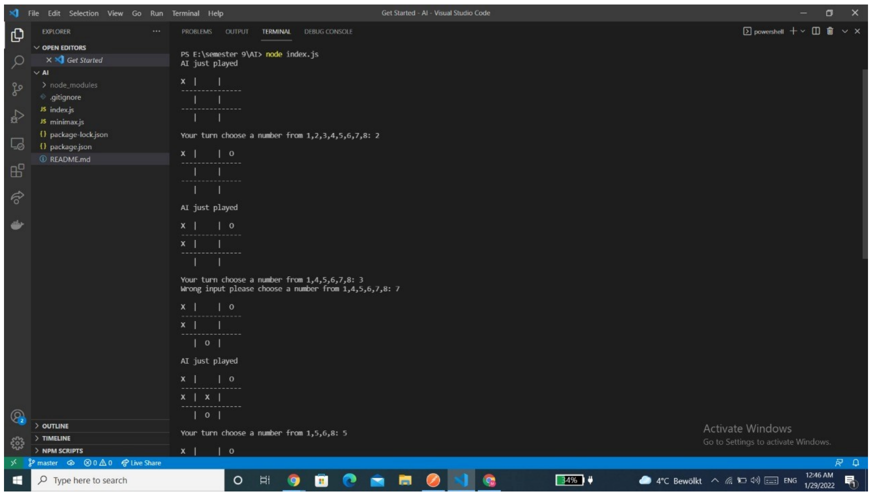


```
Your turn choose a number from 1,5,6,8: 5

X |   | O
---------------
X | X | O
---------------
  | O |

AI just played

X |   | O
---------------
X | X | O
---------------
X | O |

X wins!
PS E:\semester 9\AI>
```

0.6 Results

After running the game several times, results have showed a nearly-human intelligent behaviour from the agent in a computer win or a draw situation. User wins were rare, which we believe would be challenging for the user to repeat the game several times until their win.

An error factor can later be introduced to allow the AI agent to make mistakes at a given rate, to allow the user to have the chance to win more.



```
PS E:\semester 9\AI> node index.js
AI just played

X  |     |
---------------
   |     |
---------------
   |     |

Your turn choose a number from 1,2,3,4,5,6,7,8: 2

X  |     |  O
---------------
   |     |
---------------
   |     |

AI just played

X  |     |  O
---------------
X  |     |
---------------
   |     |

Your turn choose a number from 1,4,5,6,7,8: 3
Wrong input please choose a number from 1,4,5,6,7,8: 7

X  |     |  O
---------------
X  |     |
---------------
   |  O  |

AI just played

X  |     |  O
---------------
X  |  X  |
---------------
   |  O  |

Your turn choose a number from 1,5,6,8: 5

X  |     |  O
```

0.7 Conclusion

In conclusion, the explored prototype of an AI text-based TicTacToe mini-game using the Minimax algorithm demonstrated the power of AI agents in challenging, basic, yet entertaining, mini-games

0.8 References

Here is a list of references used throughout the development of the project, and the developed project on github:

[1] https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/

[2] https://www.youtube.com/watch?v=trKjYdBASyQ

[3] https://github.com/mostafahatem20/AI_Project