DEPARTMENT OF COMPUTER SCIENCE

COMP338 - Artificial Intelligence

Course Project #2: Games and Adversary Search

Tic Tac Toe Game

St#1 "Bader Manasra : 1193186"

St#2 "Yazan Zidan : 1193162"

First: what Tic Tac Toe game?

Tic-tac-toe (or X/O) is a two player adversary game where each player marks X or O on a 3X3 grid. Players take turns marking a square at each step and each player tries to mark a line of Xs or Os, the winner is the player that marks either a horizontal, vertical, or diagonal row.

Second: What the Minimax Algorithm?

- Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory. It provides an optimal move for the player assuming that opponent is also playing optimally.
- Mini-Max algorithm uses recursion to search through the game-tree.
- Min-Max algorithm is mostly used for game playing in AI. Such as Chess, Checkers, tic-tac-toe, go, and various tow-players game. This Algorithm computes the minimax decision for the current state.
- The minimax algorithm performs a depth-first search algorithm for the exploration of the complete game tree.

Constructing a Game Tree:

- Initial State: Start with the initial state of the game at the root of the tree.
- Expand Nodes: For each possible move, create a child node. Continue this for each player alternately until a terminal state (win, lose, or draw) is reached.
- Assigning Values: Assign a value to each terminal state. For example, +2 for a win, -2 for a loss, and 0 for a draw.
- Backtracking: Once the terminal states are reached, backtrack to calculate the value of the parent nodes. This is where the minimax decision is made. For a maximizer node, select the child with the maximum value; for a minimizer node, select the child with the minimum value.
- Optimal Move: The value at the root of the tree will give the best possible move considering the opponent also plays optimally.

Third: The Tic-Tac-Toe game and its rules:

Tic-Tac-Toe, also known as Noughts and Crosses or Xs and Os, is a simple, two-player game that is typically played on a 3x3 grid. The game is popular due to its

straightforward rules and the fact that it requires no special equipment. Here are the basic rules and gameplay:

Objective:

The main objective in Tic-Tac-Toe is to be the first player to get three of their own marks (either an X or an O) in a row, column, or diagonal on the 3x3 grid.

Setup:

- The game is played on a square grid of 3x3 spaces.
- There are two players: one uses X and the other uses O.

Rules and Gameplay:

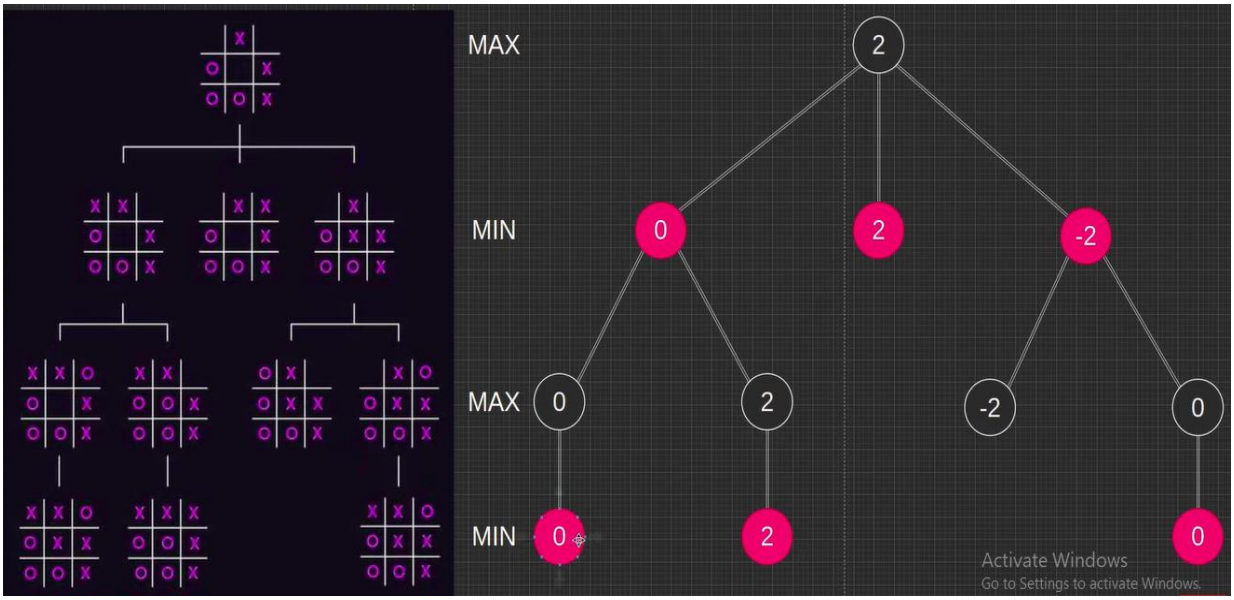
1. Starting the Game: Players decide who goes first. Traditionally, one player is X and the other is O. X usually goes first.
2. Taking Turns: Players take turns placing their mark (X or O) in an empty square.
3. Winning the Game: A player wins if they succeed in placing three of their marks in a horizontal, vertical, or diagonal row.
4. Draw or Tie: If all 9 squares are filled and neither player has 3 marks in a row, the game is a tie.
5. No Overwriting: Once a mark has been placed in a square, it cannot be changed or removed.
6. Game End: The game ends when one player wins by getting three in a row or when all squares are filled resulting in a tie.

Fourth: The complete game trees, there will be more than one tree as the game progresses, after every move taken by the player, the AI should construct a new tree,

Here in image below : On the left side, there is a visual representation of a game tree for Tic-Tac-Toe. The tree starts with a root node at the top and branches out into possible game states at each level. Each node represents a state of the Tic-Tac-Toe board, with 'X' and 'O' marks placed in different configurations. The branching of nodes represents potential moves by each player. The tree structure demonstrates the exhaustive possibilities of player moves and outcomes in a sequence of turns. It's a complete or nearly complete tree, with some paths

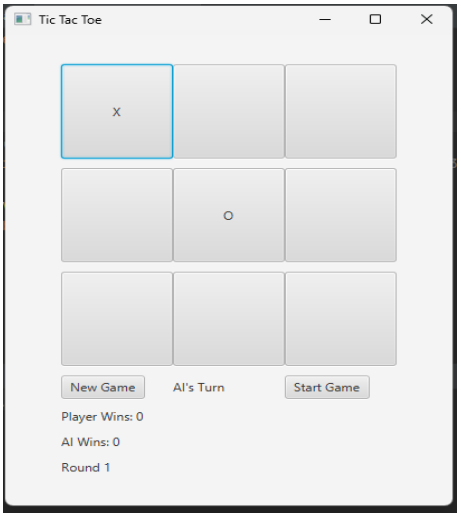
leading to wins (presumably marked with "X" for the maximizing player), losses, or draws. The paths are marked with horizontal lines connecting the nodes, suggesting a logical progression from one move to the next.

On the right side, there is a simplified version of a game tree with numerical annotations. This section of the image is labeled with "MAX" and "MIN" to represent the points in the game where the maximizing player (usually the one who is trying to win) and the minimizing player (trying to prevent the other from winning) make their moves, respectively. This portion of the image is showing a smaller, more abstracted portion of the game tree, where the final outcomes of the game are represented with numbers (2, 0, -2), indicating the utility values for the maximizing player: "2" for a win, "0" for a draw, and "-2" for a loss. The nodes are colored pink, and the tree is layered showing the decision-making process at each depth level, where the best move is chosen based on the minimax algorithm.

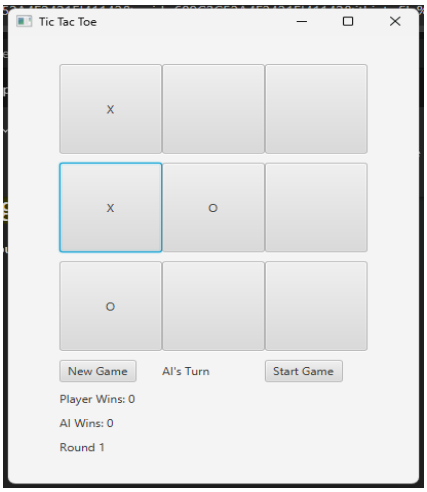


Present an example of game play :

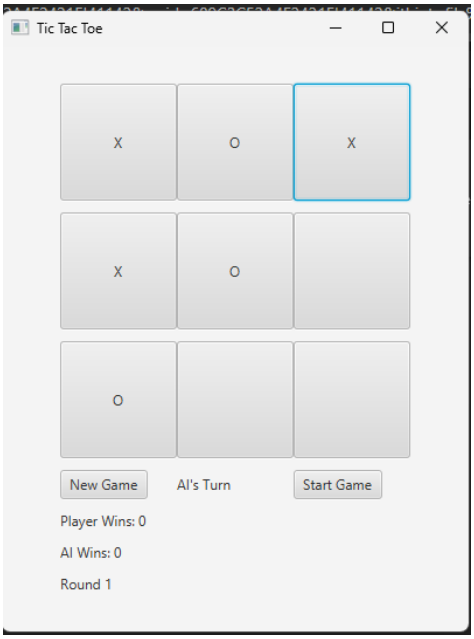
Here suppose Player play first on the top left button as (1,1) on board. Then Ai click on button (2,2) ,



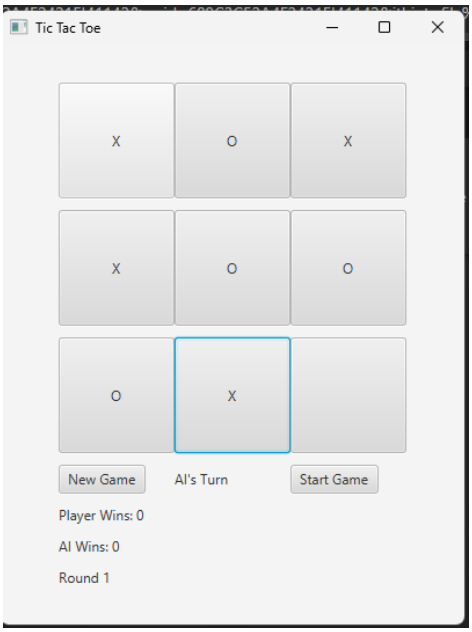
Then when player click on (2,1) ,AI click on (3,1)



Here when player click on (1,3) ,AI click on (1,2)

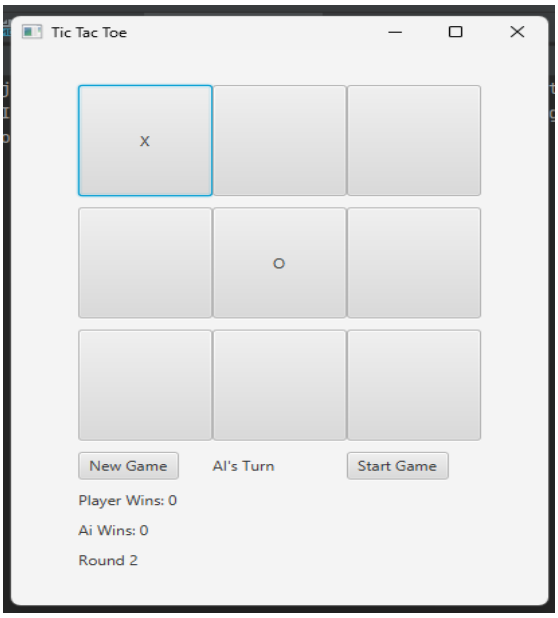


Here when player click on (3,2) ,Ai click on (2,3) that mean the game finished with A draw between the two opponents.

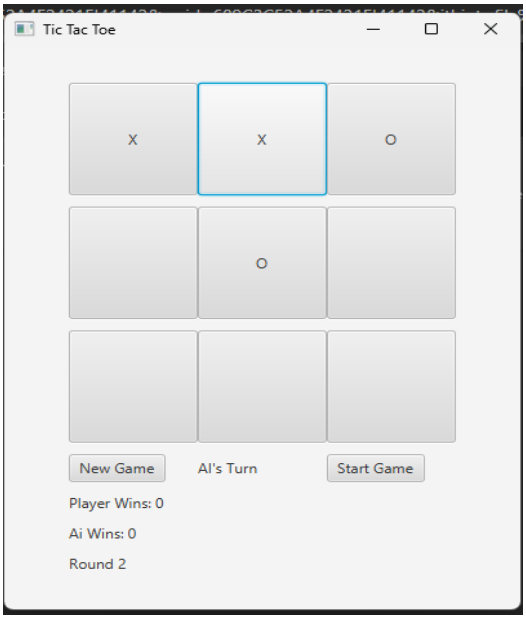


Example #2 :

Here when player click on (1,1) ,Ai click on (2,2).



Then here when player click on (1,2) , AI click on (1,3). Then game finished with Ai player wins .



Conclusion:

In conclusion, the game tree is a powerful tool for visualizing and understanding the possible outcomes of a game like Tic-Tac-Toe. By mapping out each potential move in a structured diagram, players and AI algorithms can predict and strategize future moves, looking several steps ahead to determine the best course of action.

The example provided illustrates how the minimax algorithm, used by an AI in a game of Tic-Tac-Toe, can lead to a win against a human player. The AI anticipates potential moves by the human player and strategically places its own markers to block and eventually succeed by completing a line of three Os. This not only showcases the effectiveness of the minimax algorithm in simple games but also underscores the importance of forward-thinking and planning in strategy games.

Furthermore, the abstract game tree visualizations serve as an educational tool, breaking down complex decision-making processes into easier-to-understand visuals. These abstract representations can be particularly useful for explaining the concepts of game theory and AI in academic or instructional settings.