جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering and Information Technology

Computer Science Department

COMP338 Project #2 Rep

Students name and id :

Ashraf Mtor 1183389
Iman Salameh 1201786

Instructure name : Dr.Mohammed Hellal

Date :15/01/2024

A background

The algorithm depends on many points which are:

- 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.
- In this algorithm two players play the game, one is called MAX and other is called MIN.
- Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit.
- Both Players of the game are opponent of each other, where MAX will select the maximized value and MIN will select the minimized value.
- The minimax algorithm performs a depth-first search algorithm for the exploration of the complete game tree.
- The minimax algorithm proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion.

When constructing the game tree we consider that the opponent also play in an optimal way.

Constructing the game tree:

Each game can be represented as a tree and at first the head of the tree represents the Max decision and each level down changes the state from Max to Min and vice versa

To construct the tree, for each state of the player we construct all possible moves for the opponent and in the lower level, for each state of the opponent we construct the all possible moves, we do this until we reach the leaves of the tree which contain the result of the game, we can represent results with numbers like:

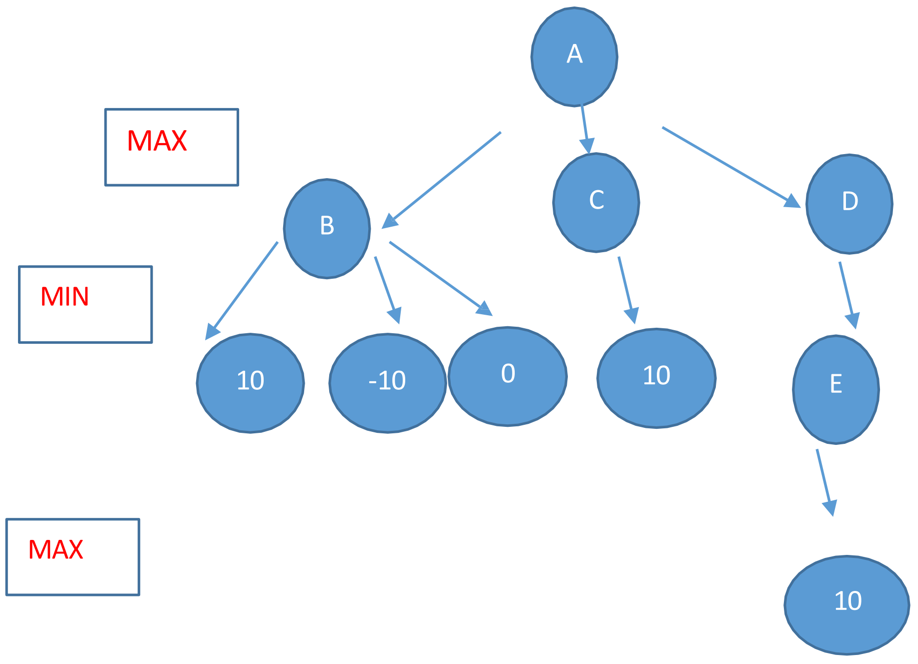
+10 for winning.

-10 for loosing (the opponent

wins). 0 for the state of draw.

The opponent always take the branch which lead to minimum result, and the player choose the one with max result.

For example to make decision in such state.



In the previous state to determine the next move we start from the

lowest level: E choose the MAX of its branches and become 10

The upper level

B choose the MIN and become -10

C become 10

D become 10

A choose the MAX and could choose either C or D in this state.

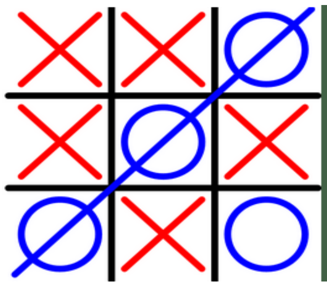
NOTE:
Some modification could be implemented by choosinf the nearest winning chance This could be applied by implementing a factor for depth and each down level gives
-1 to the state.

Tic Tac Toe game:

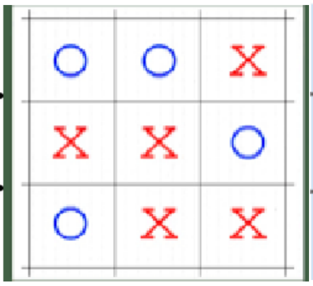
The most common Tic Tac toe is 3*3 grid.

In a 3-by-3 grid game, the player who is playing "X" goes first. Players alternate placing Xs and Os on the board until either player has three in a row, horizontally, vertically, or diagonally or until all squares on the grid are filled. If a player is able to draw three Xs or three Os in a row, then that player wins. If all squares are filled and neither player has made a complete row of Xs or Os, then the game is a draw.

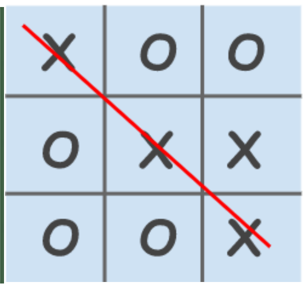
Here is some cases for winning and drawing:



O wins



Draw case



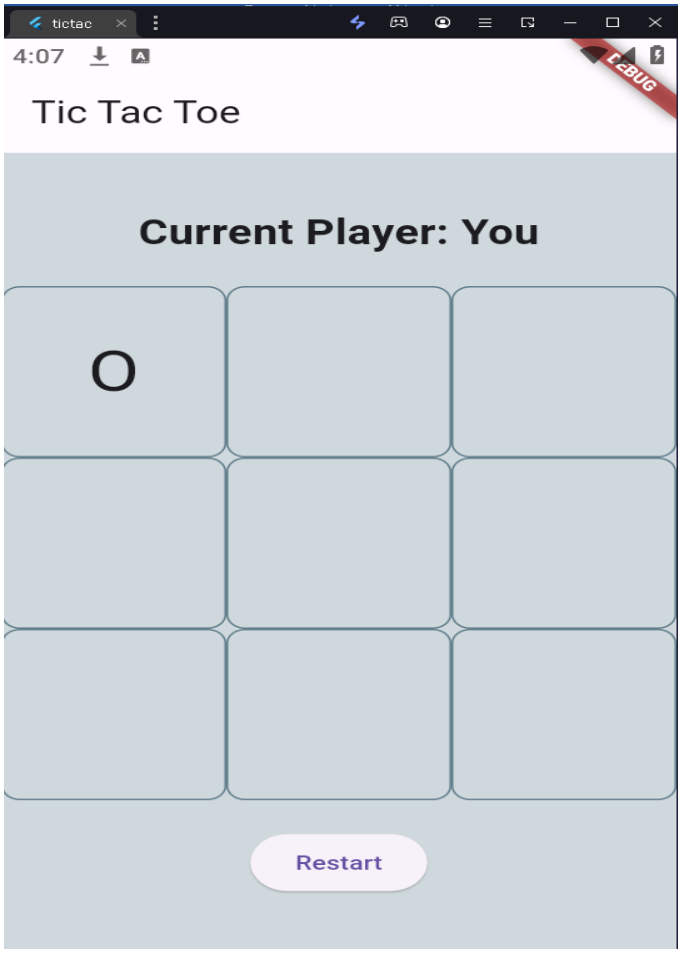
X wins

There are many strategies that players do to win the game.

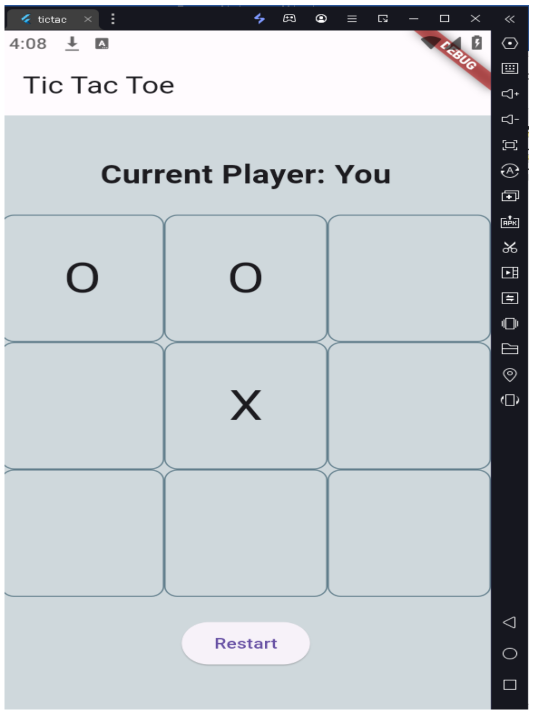
In this game the best strategy is the one which always leads to draw or win and prevent the opponent from winning.

In the next part I will discuss how to implement the Minimax algorithm in order to obtain an optimal solution for this game.

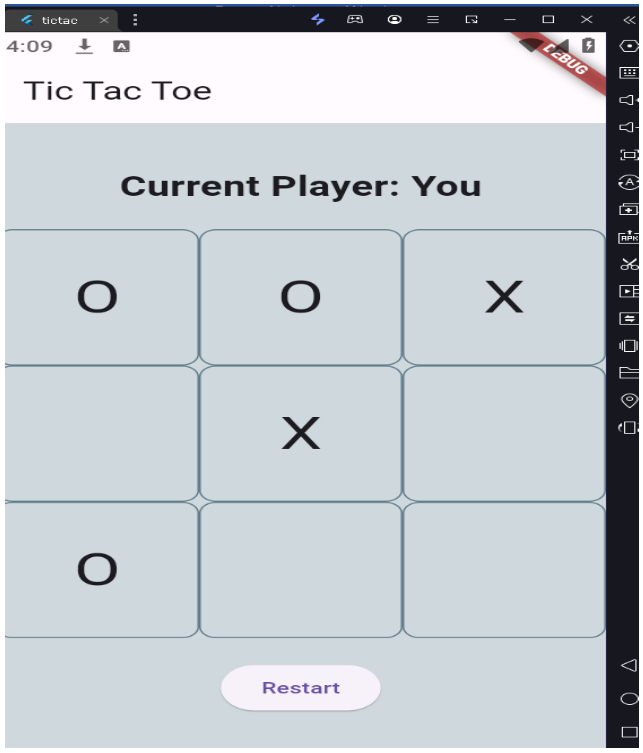
Now to construct the game tree and explain how AI make decision, first this turn start with the Ai and choose the following state:



I added ‘x’ then the AI did his turn:



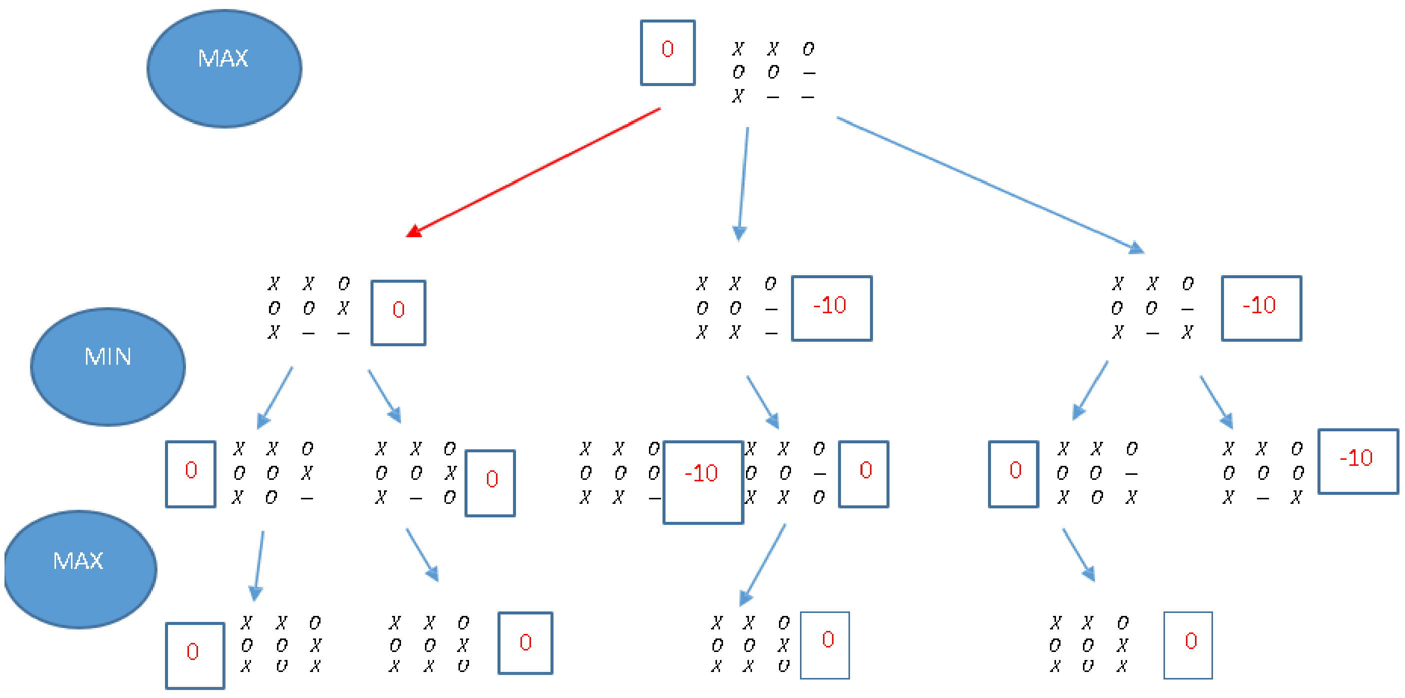
After this I added ‘x and the AI put the ‘o



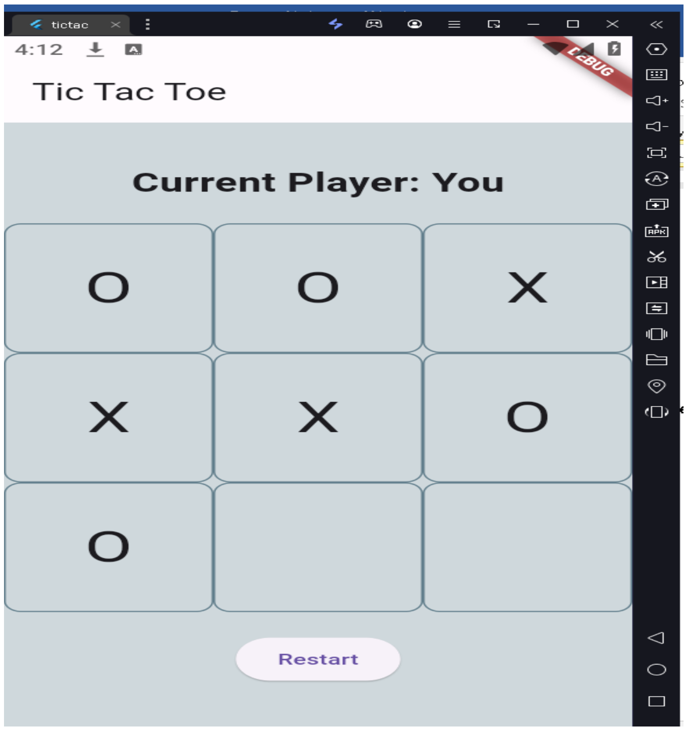
Now I put ‘x’ in the second row of the first column and the game state is:

```
O  O  X
X  X  –
O  –  –
```

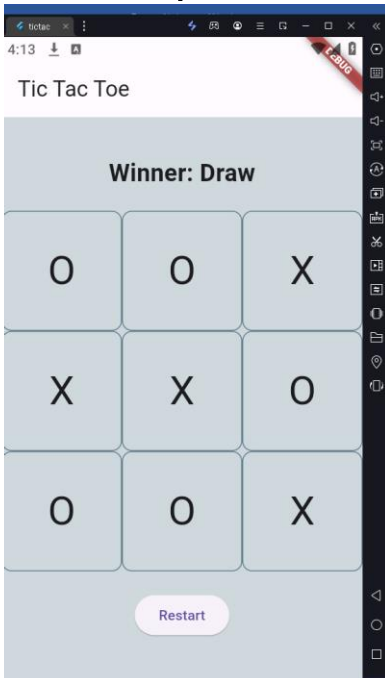
Now the AI construct the tree to make the decision:note(the graph Ai used x and me used o)



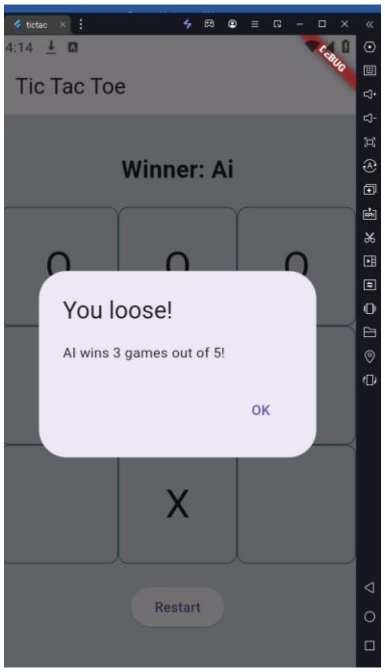
It is clear that the AI will choose the state on the left (to put x in the third column of the second row) and that what happened



After this I put 'X' on the corner and we draw



And after I lose 3 out of 5 games



Winner: Ai

You loose!

AI wins 3 games out of 5!