Artificial Intelligence
Mini-Project
Tic-Tac-Toe using Min-Max

Problem Statement -

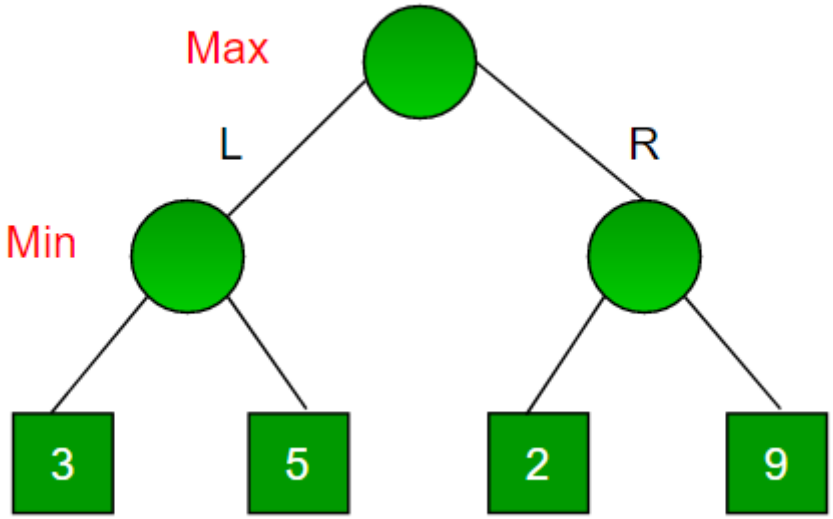
Tic-tac-toe using Min-Max algorithm (alpha beta pruning) involving two modes (two players and with the agent) using GUI.

Objective –

- To create an agent solver who scans the board and take the decision (based on min-max algorithm) to choose the best position.
- Next is two for two player the result checker which checks for the winning or losing of players.

Min – Max Algorithm -

In Minimax the two players are called maximizer and minimizer. The **maximizer** tries to get the highest score possible while the **minimizer** tries to do the opposite and get the lowest score possible.

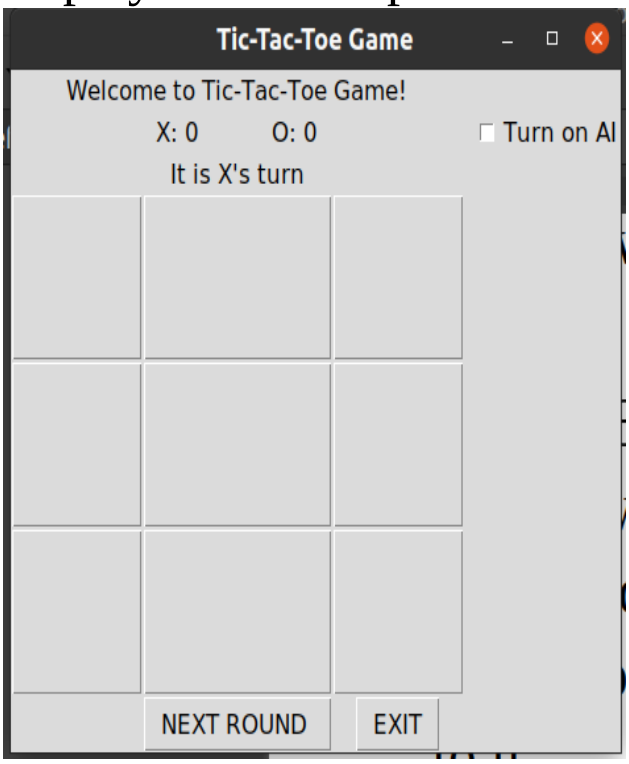


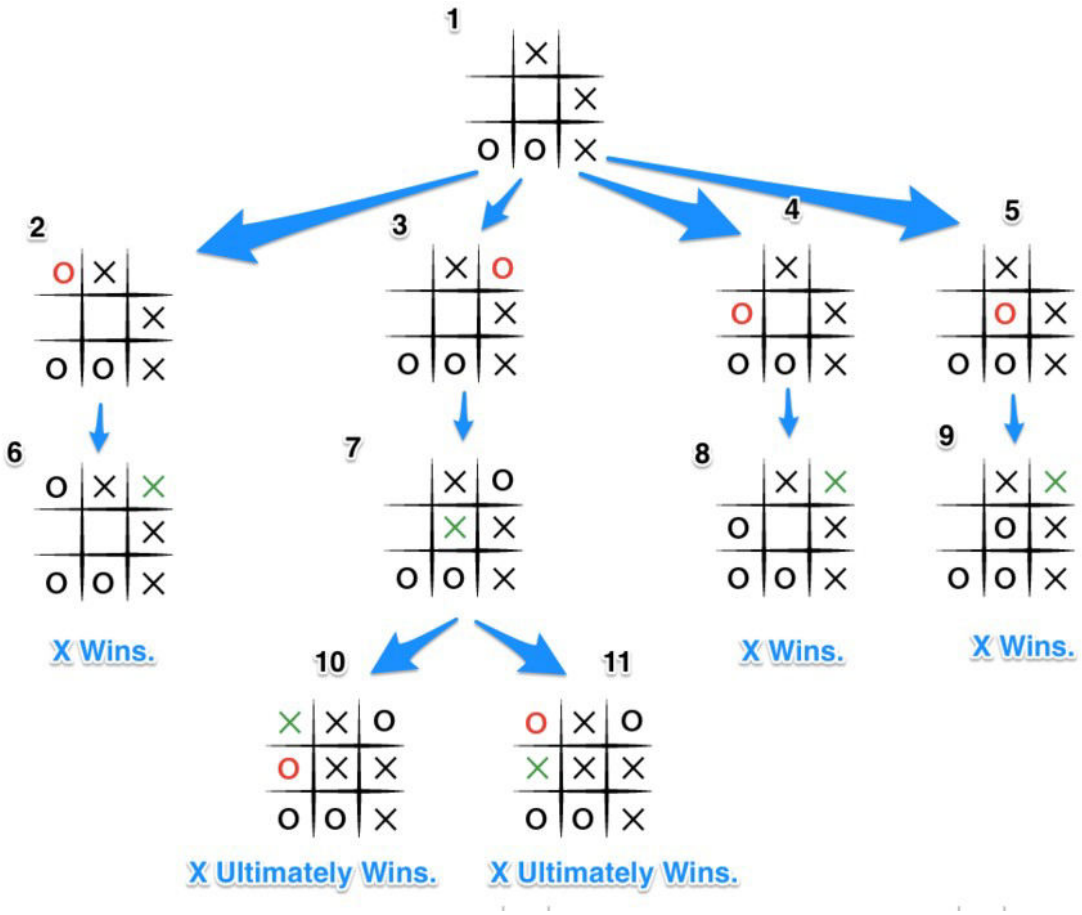
if the pattern matches – Max
else – Min

Design and Architecture-

We used python and tkinter for GUI and min-max algorithm to find the best move and another function to use for the two player mode to check who is winner.

We use 3X3 board to play with each position is like a button which has response corresponding to it.





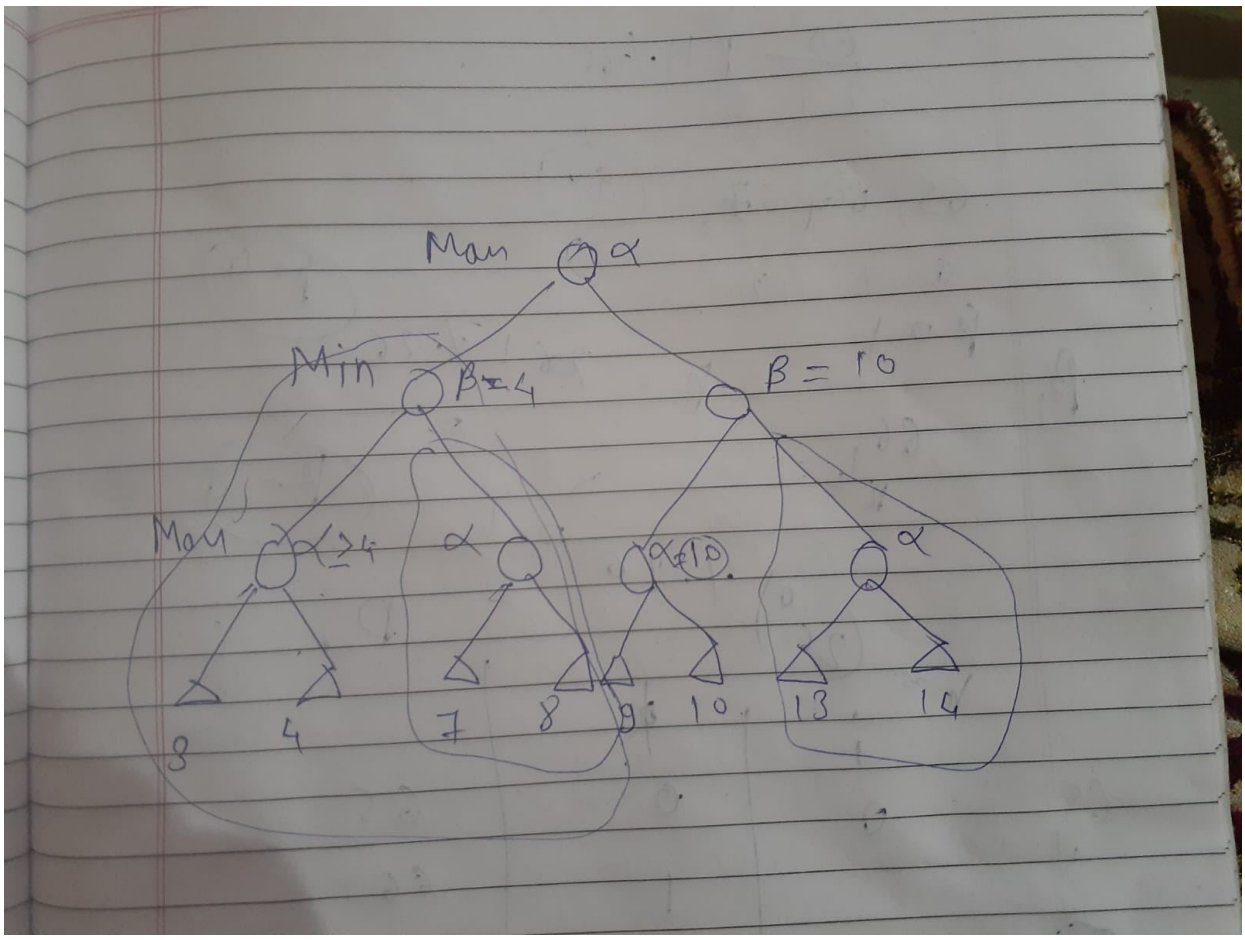
Algorithm for finding best move-

<u>Basic Algorithm to find next move-</u>

```
def findpath(board):
   for palce in board:
         if min_max(place) == bestmove:
               best_move = place
   return best_move
```

Alpha-Beta prunning -

- alpha and beta are just two values for each level in the game tree(for MIN – beta and MAX – Alpha)
- alpha is set to INT_MIN and beta is set to INT_MAX.
- whenever we got the max value we set beta >= max_value and skip the other part to explore
- whenever we got the min value we set beta >= min_value and skip the other part to explore
- thus as top the max will try to explore only that part of the game tree which will give the maximum profit.
- Thus time complexity is reduced from O(b**d) to about O(b**d/2)



-

Min-Max Algorithm-

```
def min_max(board,place,alpha,beta):
  for position in range(0,9):
        if check_win(board,place):
              return win
        if player == computer:
              best_move = INT_MIN
        else:
               best_move = INT_MAX
        if player == computer:
                best_move = max(best_move)
                alpha = min(alpha,best_move)
        else:
                best_move = min(best_move)
                beta = max(beta,best_move)
        if      beta < alpha:
               return best_move
```

Outputs -

Two player mode -

With an Agent -


Tic-Tac-Toe Game
Welcome to Tic-Tac-Toe Game!
X: 0
O: 1
Turn on AI
O wins!!!
O
O
X
O
X
X
O
X
NEXT ROUND
EXIT