DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRINCIPLES OF AI LAB

2K22 AFI Batch

LAB FILE

SUBMITTED TO:
Gull Kaur Ma'am

SUBMITTED BY:
Shikhar Asthana
(Roll No. 2K22/AFI/24)

EXPERIMENT 6

AIM: Consider the Tic-Tac-Toe game. Implement a solution for the game using the MiniMax Algorithm.

THEORY: The Tic-Tac-Toe game is a 2-player game where each player takes turn marking an empty 3X3 grid using ‘X’ and ‘O’ respectively. The aim of the game is for the player to place their marks in a horizontal, vertical or diagonal row. The first player to achieve the same is the winner.

How this game can be mapped to AI:

- The game can be created on a 3X3 grid where each player can mark their symbols.
- Instead of 2 players playing, we will have 1 human player and 1 AI player (bot).
- The player and bot will keep taking turns until either one of them wins/draws.

The strategy in which the bot decides which move to play is the crux of our AI implementation. The bot can convert all possible moves from a given state which can represent our state space tree. The root of the tree will be the current state and the leaf nodes will represent termination conditions of the game (win/lose/draw). MiniMax is one of these strategies. In this, after every move of the player, the bot will try to calculate all possible moves until a termination condition is reached. Then the algorithm will assign scores to each termination condition – Win will have positive score, lose will have negative score and Draw can have a neutral score. Since while trying out all positive moves, the bot will also have to take into account the possible moves of what the human player might take, the bot will need to alternate between choosing the best move (Max – representing best bot move) and choosing the min move (Min – representing the player move). Based on these, the algorithm can reach back to the current state and choose the move which has the highest score. This will ensure that no matter what the scenario is, the bot will always take the best possible moves. If implemented correctly, the bot can either win or draw but can never lose.

MiniMax algorithm for Tic-Tac-Toe is manageable due to the relatively smaller size of possible moves compared to the computational power available in current computers. However, it is an exhaustive approach and might create performance issues when applied on larger problems/games.

ALGORITHM:

```
1. Minimax(board,isMaximising)
    a. Check for terminal conditions based on passed state (board):
        i. If Player won:
            1. Return a negative score
        ii. If AI won:
            1. Return a positive score
        iii. If Draw:
            1. Return a neutral score
    b. If isMaximising is True:
        i. Find best possible move from the remaining spaces of the board.
        ii. Best score ← -800 (any large non negative value)
        iii. For empty spaces in board:
            1. Score ← Minimax(board,False)
            2. If Score >Best Score:
                a. Best Score = Score
        iv. Return Best Score
    c. If isMaximising is False:
        i. Find worst possible move from the remaining spaces of the board.
        ii. Best score ← +800 (any large positive value)
        iii. For empty spaces in board:
            1. Score ← Minimax(board,False)
            2. If Score < Best Score:
                a. Best Score = Score
        iv. Return Best Score
2. BotMove()
    a. This will use the above function to decide the best possible move.
    b. Best Score = -800 (any large non negative value)
    c. For empty spaces in board:
        i. Score = Minimax(board, False)
        ii. If Score >Best Score:
            1. Best Score = Score
        iii. Play the move associated with best score.
```

CODE: Implemented code is submitted over the google classroom assignment.

OUTPUT:

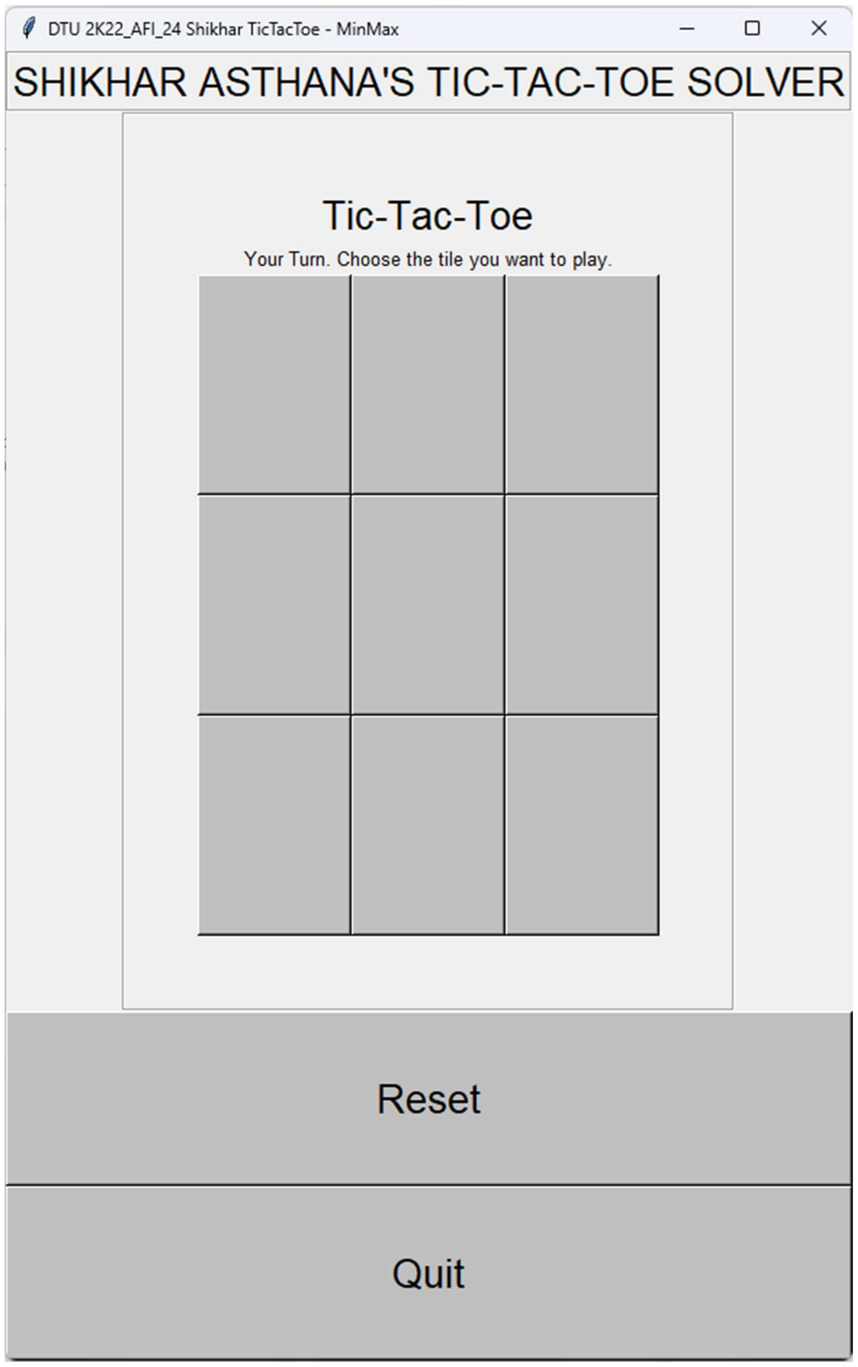
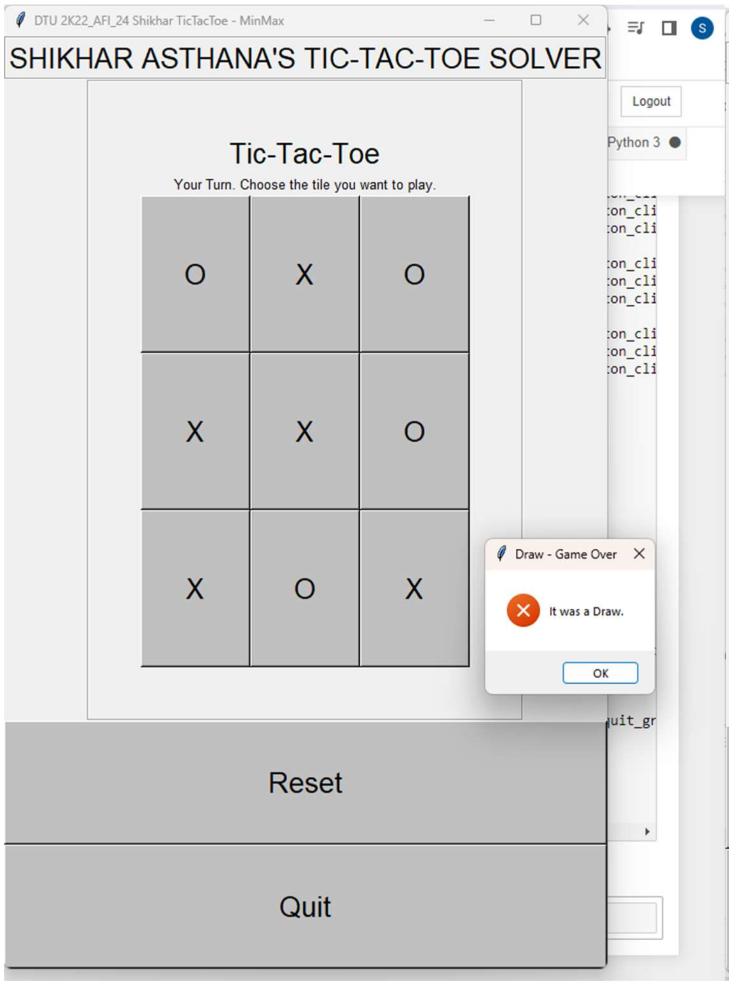

DTU 2K22_AFI_24 Shikhar TicTacToe - MinMax
SHIKHAR ASTHANA'S TIC-TAC-TOE SOLVER
Tic-Tac-Toe
Your Turn. Choose the tile you want to play.
Reset
Quit

Fig 6.1 Sample UI

The above image shows the initial state of the UI. The user can play their turn by clicking on the appropriate tile to make a move. The computer will then apply MiniMax and play its move automatically. Post which, the player will again get a chance to play their move and so on until someone wins or it's a draw.

Reset button will empty all marks on the grid and bring UI back to initial state so that the player can play another game. Quit button will quit the UI window.


DTU 2K22_AFI_24 Shikhar TicTacToe - MinMax

SHIKHAR ASTHANA'S TIC-TAC-TOE SOLVER

Tic-Tac-Toe

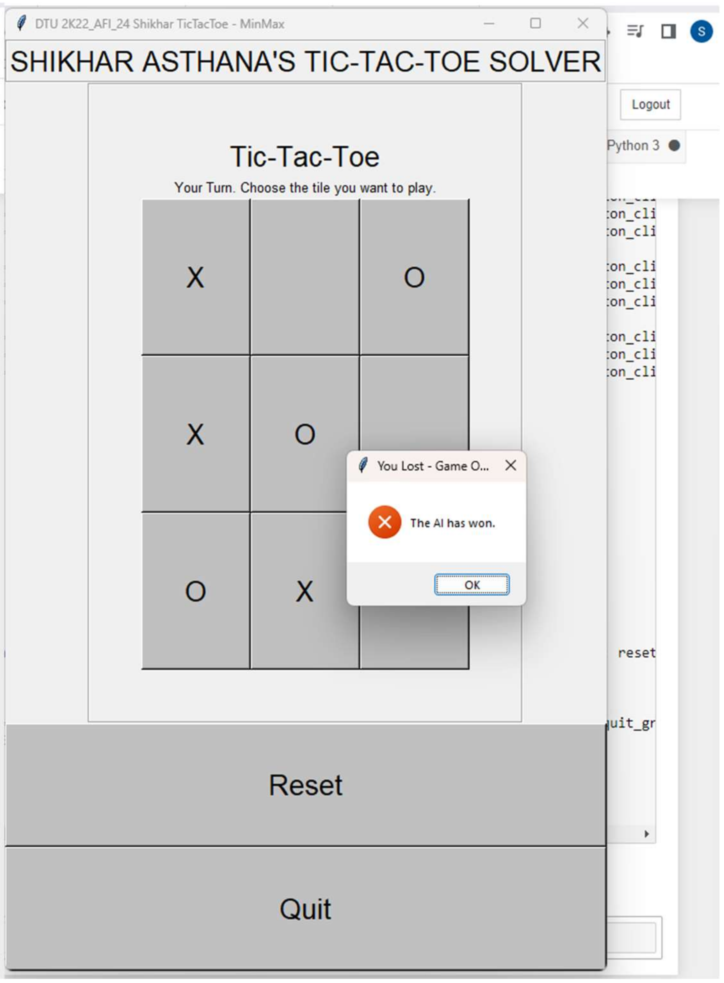
Your Turn. Choose the tile you want to play.

O	X	O
X	X	O
X	O	X

Draw - Game Over

It was a Draw.

Fig 6.2 Draw


DTU 2K22_AFI_24 Shikhar TicTacToe - MinMax

SHIKHAR ASTHANA'S TIC-TAC-TOE SOLVER

Tic-Tac-Toe

Your Turn. Choose the tile you want to play.

X		O
X	O	
O	X	

You Lost - Game O...

The AI has won.

Fig 6.3 AI Wins

The above 2 output images showcase the only two possible cases which can be reached. In the first case, we are able to reach a terminal state of Draw. In the second case, the AI wins.

As mentioned earlier, only these two cases are possible because our bot is using the MiniMax algorithm to exhaustively search all possible moves and choose the best from the available ones. This always ensures that the bot will win or in the worst case, will reach draw.