10/12/24

Lab-9

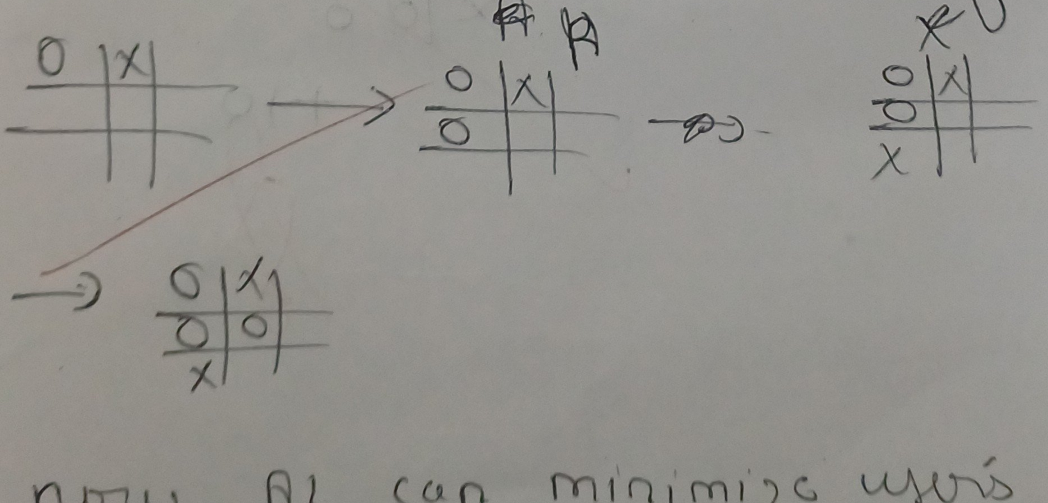
MinMax for TicTacToe

```
board = [[' ',' ',' '],[' ',' ',' '],[' ',' ',' ']]
function print board(board):
    for row in board:
        print row

function check-winners(board)
    for rows in board:
        if row[0] == row[1] == row[2]
            return row[0]

    for col in range(3):
        if board[0][col] == board[1][col
            return board[0][col]

    if board[0][0] == board[1][1] &
    return board[0][0]

    if board[0][2] == board[1][1]
    == b[2][0]
    return b[0][2]
```

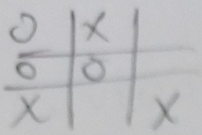
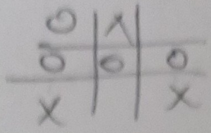
```
def is-full (board):
    for row in board:
        if ' ' in row:
            return false

    return True


def minmax (board, depth ,is-maxi):
    win = check - winner (board)
    if win == 'x':
        return 10-depth

    else if win == 'o':
        return depth-10

    else is-full (board):
        return 0:

    if is-maxi:
        best. score = float ('-inf')
        ~~for~~
        for i in range(3)
            for j in range(3)
                if board [i] [j] == 'o'
            board[i][j] = 'x'
            score = minimax(board, d+1,
                                   False)
        best_score = max(best_score, score)
        return best_score

    else:
        best_score = float(inf)
        for ~~Ai~~ i in range(3)
            for j in range(3)

    if board[i][j] == 'x'
        board[i][j] = 'O'
```

Output

user - O   AI - X



now AI can minimize user's score

 but  (user's turn)