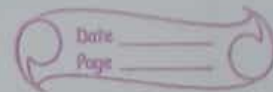


u/10/23

Lab Program - 2



Tic-Tac-Toe

```
board = { 1: ' ', 2: ' ', 3: ' ',
          4: ' ', 5: ' ', 6: ' ',
          7: ' ', 8: ' ', 9: ' ' }
```

```
def printBoard (board):
    print ( board[1] + ' | ' + board[2] + ' | ' +
            board[3] )
    print ( '- + - + - ')
    print ( board[4] + ' | ' + board[5] + ' | ' + board[6] )
    print ( '- + - + - ')
    print ( board[7] + ' | ' + board[8] + ' | ' + board[9] )
    print ( '\n' )
```

```
def spaceFree (pos):
    if (board[pos] == ' '):
        return True
    else:
        return False
```

```
def checkwin () :
    if (board[1] == board[2] and board[1] == board[3] and
        board[1] != ' '):
        return True
    elif (board[4] == board[5] and board[4] == board[6] and
        board[4] != ' '):
        return True
    elif (board[7] == board[8] and board[7] == board[9]
        and board[7] != ' '):
        return True
```

```

elif (checkwin()):
    if (letter == 'X'):
        print ('Bot wins')
    else:
        print ('You win!')
    return
else:
    print ("position Taken")
    position = int (input (" Enter new pos "))
    insertletter (letter, position)
    return

```

```

player = 'O'
bot = 'X'

```

```

def playerMove():
    position = int (input (" Enter new pos "))
    insertletter (letter, position)
    return

```

```

def compMove():
    bestScore = -1000
    bestMove = 0
    for key in board.keys():
        if (board[key] == ''):
            board[key] = bot
            score = minimax (board, False)
            board[key] = ''
            if (score > bestScore):
                bestScore = score
                bestmove = key
    insertletter (bot, bestmove)
    return

```

```

def minimax(board, isMax):
    if (checkMoveForWin(bot)):
        return 2
    elif (checkMoveForWin(player)):
        return -1
    elif (checkDraw()):
        return 0

```

```

    if isMax:
        bestScore = -1000
        for key in board.keys():
            if board[key] == ' ':
                board[key] = bot
                score = minimax(board, False)
                board[key] = ' '
                if (score > bestScore):
                    bestScore = score
        return bestScore
    else:
        bestScore = 1000

```

```

        for key in board.keys():
            if board[key] == ' ':
                board[key] = player
                score = minimax(board, True)
                board[key] = ' '
                if (score < bestScore):
                    bestScore = score
        return bestScore

```

```

while not checkWin():
    compMove()
    playerMove()

```

Q. 04.06.24