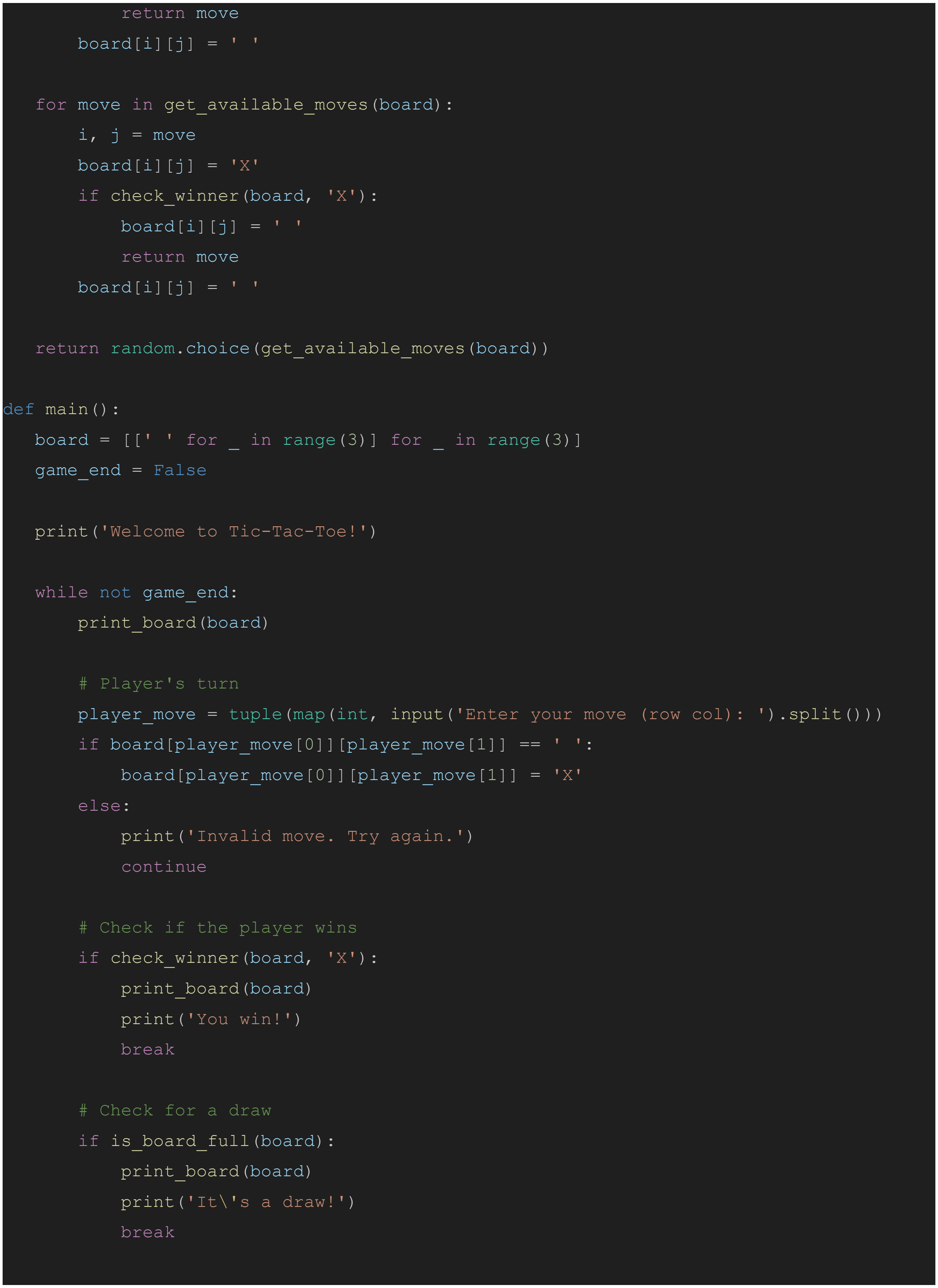
AI EXP 1.2
Vignesh RK
9551
TE COMPS-A

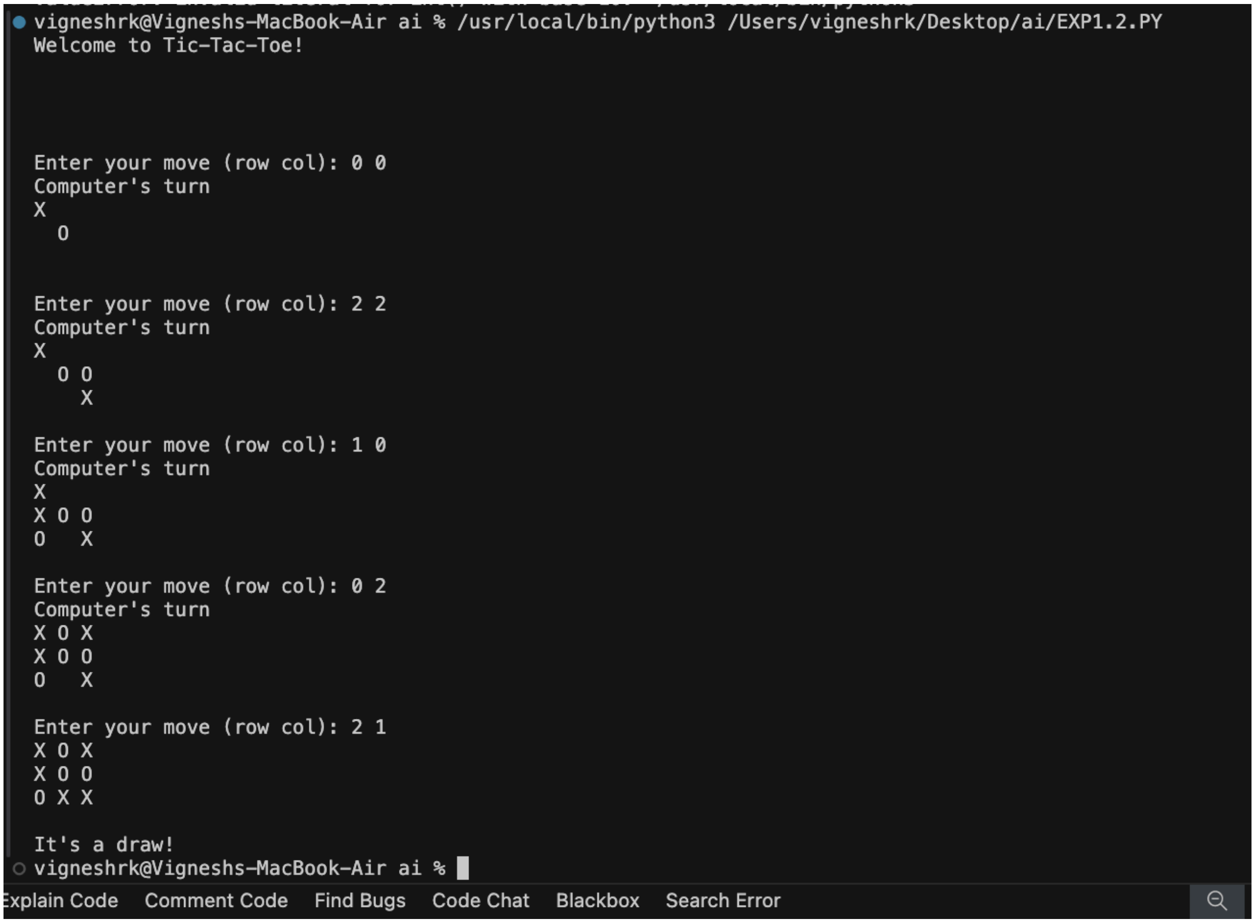
```
import random

def print_board(board):
    for row in board:
        print(' '.join(row))
    print()

def check_winner(board, player):
    for i in range(3):
        if all(board[i][j] == player for j in range(3)) or all(board[j][i] == player for j in range(3)):
            return True
    if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i] == player for i in range(3)):
        return True
    return False

def evaluate(board):
    if check_winner(board, 'X'):
        return -1   # Player X wins
    elif check_winner(board, 'O'):
        return 1    # Player O wins
    else:
        return 0    # It's a draw

def is_board_full(board):
    return all(board[i][j] != ' ' for i in range(3) for j in range(3))

def get_available_moves(board):
    return [(i, j) for i in range(3) for j in range(3) if board[i][j] == ' ']

def get_best_move(board):
    for move in get_available_moves(board):
        i, j = move
        board[i][j] = 'O'
        if check_winner(board, 'O'):
            board[i][j] = ' '
```



```
           return move
       board[i][j] = ' '

   for move in get_available_moves(board):
       i, j = move
       board[i][j] = 'X'
       if check_winner(board, 'X'):
           board[i][j] = ' '
           return move
       board[i][j] = ' '

   return random.choice(get_available_moves(board))

def main():
   board = [[' ' for _ in range(3)] for _ in range(3)]
   game_end = False

   print('Welcome to Tic-Tac-Toe!')

   while not game_end:
       print_board(board)

       # Player's turn
       player_move = tuple(map(int, input('Enter your move (row col): ').split()))
       if board[player_move[0]][player_move[1]] == ' ':
           board[player_move[0]][player_move[1]] = 'X'
       else:
           print('Invalid move. Try again.')
           continue

       # Check if the player wins
       if check_winner(board, 'X'):
           print_board(board)
           print('You win!')
           break

       # Check for a draw
       if is_board_full(board):
           print_board(board)
           print('It\'s a draw!')
           break
```

```python
        # Computer's turn
        print('Computer\'s turn')
        computer_move = get_best_move(board)
        board[computer_move[0]][computer_move[1]] = 'O'

        # Check if the computer wins
        if check_winner(board, 'O'):
            print_board(board)
            print('Computer wins!')
            break

        # Check for a draw again
        if is_board_full(board):
            print_board(board)
            print('It\'s a draw!')
            break

if __name__ == "__main__":
    main()
```

Output:



```
vigneshrk@Vigneshs-MacBook-Air ai % /usr/local/bin/python3 /Users/vigneshrk/Desktop/ai/EXP1.2.PY
Welcome to Tic-Tac-Toe!




Enter your move (row col): 0 0
Computer's turn
X
  O


Enter your move (row col): 2 2
Computer's turn
X
  O O
    X

Enter your move (row col): 1 0
Computer's turn
X
X O O
O   X

Enter your move (row col): 0 2
Computer's turn
X O X
X O O
O   X

Enter your move (row col): 2 1
X O X
X O O
O X X

It's a draw!
vigneshrk@Vigneshs-MacBook-Air ai %
```