Name: Aniket Baburam Narbariya

UID:2019130043

TE COMPS, BATCH C

Exp NO:3

Code:

```
goalStates = [
   [(0, 0), (0, 1), (0, 2)],
   [(1, 0), (1, 1), (1, 2)],
   [(2, 0), (2, 1), (2, 2)],
   [(0, 0), (1, 0), (2, 0)],
   [(0, 1), (1, 1), (2, 1)],
   [(0, 2), (1, 2), (2, 2)],
   [(0, 0), (1, 1), (2, 2)],
   [(0, 2), (1, 1), (2, 0)],
]

#define the empty board
board = [
   ['_', '_', '_'],
   ['_', '_', '_'],
   ['_', '_', '_'],
]

counterLimit = 5

def calculate_F_Value(i , j):
   maxElement = [-1, -1, -1]
   for _ in goalStates:
      empty = 0
      dot = 0
      cross = 0
      if (i, j) in _:
         for k in _:
            if board[k[0]][k[1]] == '_':
               empty += 1
            if board[k[0]][k[1]] == 'o':
               dot += 1
            if board[k[0]][k[1]] == 'x':
               cross += 1
      if maxElement[2] < cross:
         maxElement = [i, j, cross]

   return maxElement
```

```
def playAI():
    fvalueList = []
    for i in range(3):
        for j in range(3):
            if board[i][j] == '_':
                board[i][j] = 'o'
                fvalueList.append(calculate_F_Value(i, j))
                board[i][j] = '_'
    position = max(fvalueList, key=lambda x: x[2])
    board[position[0]][position[1]] = 'o'

def checkStatus():
    flagH = None
    counter = 0
    for i in range(3):
        for j in range(3):
            if board[i][j] != '_':
                counter += 1

    if counter == 9:
        flagH = "Draw"

    for location in goalStates:
        if board[location[0][0]][location[0][1]] == 'x' and board[location[1][0]][location[1][1]] == 'x' and
board[location[2][0]][location[2][1]] == 'x':
            flagH = True
            break
        elif board[location[0][0]][location[0][1]] == 'o' and board[location[1][0]][location[1][1]] == 'o' and
board[location[2][0]][location[2][1]] == 'o':
            flagH = False
            break
    return flagH

def print_board():
    print(board[0][0], "|", board[0][1], "|", board[0][2])
    print(board[1][0], "|", board[1][1], "|", board[1][2])
    print(board[2][0], "|", board[2][1], "|", board[2][2])
    print("\n\n")

endFlag = False

print_board()

while True:
    #take the current location input
    humanLocation = list(map(int, input("Enter your next move location: ").strip().split()))
```

```
    humanLocation = [humanLocation[0] - 1, humanLocation[1] - 1]

    if board[humanLocation[0]][humanLocation[1]] != '_':
        print("!!It's not an empty cell!!")
        continue

    board[humanLocation[0]][humanLocation[1]] = 'x'
    print_board()

    gameStatus = checkStatus()
    if gameStatus == True:
        print("You won!!")
        endFlag = True
        break
    elif gameStatus == False:
        print("You lost!!")
        endFlag = True
        break
    elif gameStatus == "Draw":
        print("Match Draw!!")
        endFlag = True
        break

    if not endFlag: playAI()

    print_board()
```

```
    gameStatus = checkStatus()
    if gameStatus == True:
        print("You won!!")
        break
    elif gameStatus == False:
        print("You lost!!")
        break
    elif gameStatus == "Draw":
        print("Match Draw!!")
```

Conclusion:

Hence, I have implemented tic-tac-toe agent using A* algorithm.
The agent selects the move among available moves to break the streak of opponent.
Also, here, the agent uses the number of crosses(human input) as the heuristic value so,as to play the respective next move in the optimal way.