Experiment No. 3

Name : Sudhanshu Narsude

UID : 2019130044

Class : TE COMPS Batch C

Subject : AIML

Aim : Tic – tac – toe using A* algorithm .

Program :

```
myBoard = [
    [' ', ' ', ' '],
    [' ', ' ', ' '],
    [' ', ' ', ' '],
]

counterLimit = 5


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)],
]

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 myBoard[k[0]][k[1]] == ' ':
                    empty += 1
                if myBoard[k[0]][k[1]] == 'O':
                    dot += 1
                if myBoard[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 myBoard[i][j] == ' ':
                myBoard[i][j] = 'O'
                fvalueList.append(calculate_F_Value(i, j))
                myBoard[i][j] = ' '
    position = max(fvalueList, key=lambda x: x[2])
    myBoard[position[0]][position[1]] = 'O'

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

    if counter == 9:
        flagH = "Draw"

    for location in goalStates:
        if myBoard[location[0][0]][location[0][1]] == 'X' and
myBoard[location[1][0]][location[1][1]] == 'X' and
myBoard[location[2][0]][location[2][1]] == 'X':
            flagH = True
            break
        elif myBoard[location[0][0]][location[0][1]] == 'O' and
myBoard[location[1][0]][location[1][1]] == 'O' and
myBoard[location[2][0]][location[2][1]] == 'O':
            flagH = False
            break
    return flagH


# Function to print Tic Tac Toe
def print_tic_tac_toe():
    print("\n")
    print("\t     |     |")
    print("\t  {}  |  {}  |  {}".format(myBoard[0][0], myBoard[0][1],
myBoard[0][2]))
    print('\t_____|_____|_____')
```

```
    print("\t     |     |")
    print("\t  {}  |  {}  |  {}".format(myBoard[1][0], myBoard[1][1],
myBoard[1][2]))
    print('\t_____|_____|_____')

    print("\t     |     |")

    print("\t  {}  |  {}  |  {}".format(myBoard[2][0], myBoard[2][1],
myBoard[2][2]))
    print("\t     |     |")
    print("\n")

endFlag = False

print_tic_tac_toe()

while True:
    humanLocation = list(map(int, input("Enter your next move location:
").strip().split()))
    #humanLocation = [humanLocation[0] - 1, humanLocation[1] - 1]

    if myBoard[humanLocation[0]][humanLocation[1]] != ' ':
        print("Watch out!!\nIt's not an empty cell")
        continue

    myBoard[humanLocation[0]][humanLocation[1]] = 'X'
    print_tic_tac_toe()

    gameStatus = checkWin()
    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_tic_tac_toe()

    gameStatus = checkWin()
```

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

.Output:

```
PS C:\Users\sudha\Ring_Algorithm> python -u "c:\Users\sudha\game.py"

     |     |
     |     |
_____|_____|_____
     |     |
     |     |
_____|_____|_____
     |     |
     |     |
     |     |

Enter your next move location: 0 0

     |     |
  X  |     |
_____|_____|_____
     |     |
     |     |
_____|_____|_____
     |     |
     |     |
     |     |



     |     |
  X  |  O  |
_____|_____|_____
     |     |
     |     |
_____|_____|_____
     |     |
     |     |
```

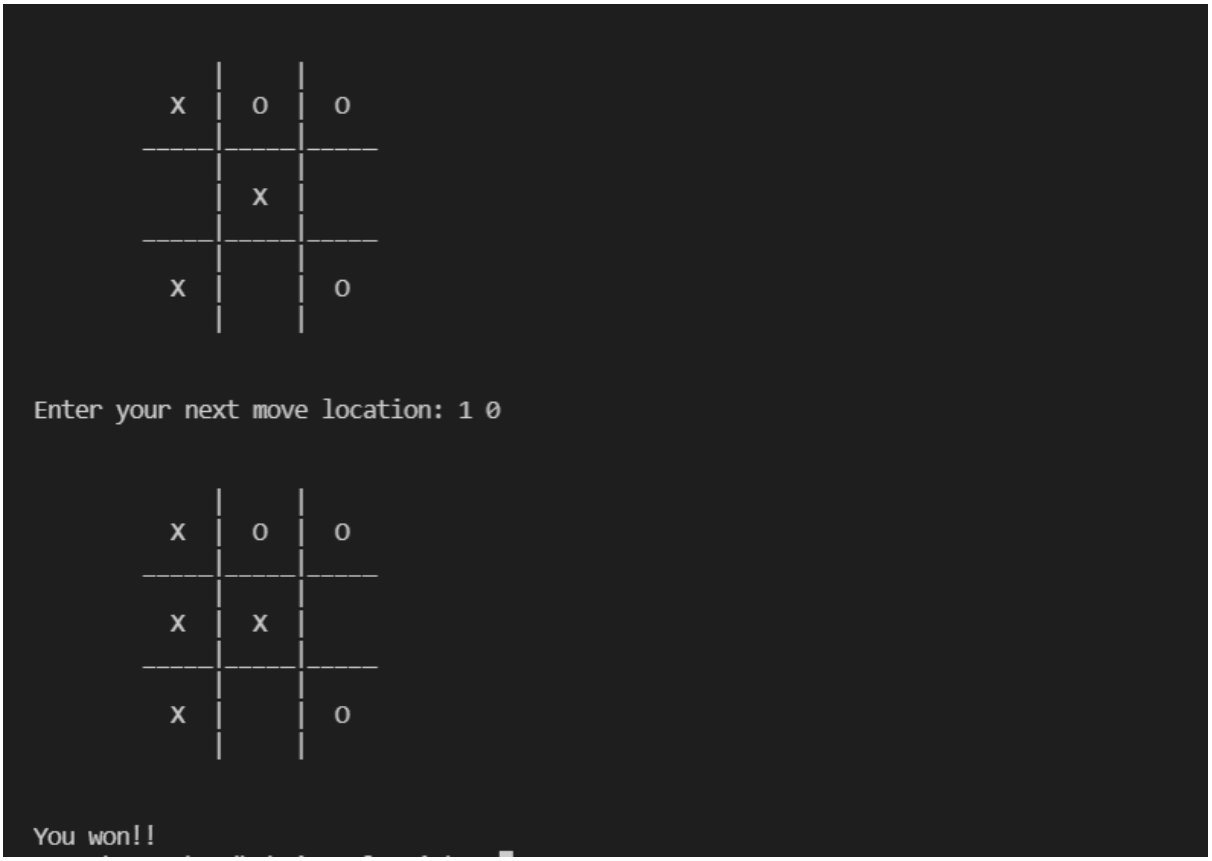
```
Enter your next move location: 1 1

     |     |
  X  |  O  |
_____|_____|_____
     |     |
     |  X  |
_____|_____|_____
     |     |
     |     |
     |     |


     |     |
  X  |  O  |
_____|_____|_____
     |     |
     |  X  |
_____|_____|_____
     |     |
     |     |  O
     |     |

Enter your next move location: 2 0

     |     |
  X  |  O  |
_____|_____|_____
     |     |
     |  X  |
_____|_____|_____
     |     |
  X  |     |  O
     |     |
```



Conclusion :

In this experiment I implemented Tic-tac-toe using A* algorithm. Here I kept ‘X’ as a human player and ‘O’ as a Computer player. Agent Computer uses the number of crosses as heuristic value and selects the move among the available moves which will break the streak of opponent .