Jaiwin Shah
2019130058
TE Comps
Batch – C

Experiment-3

Aim: To implement the game, Tic-Tac-Toe using the A* algorithm

Code:

```
brd = [
    ['_', '_', '_'],
    ['_', '_', '_'],
    ['_', '_', '_'],
]

finalstate = [
    [(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 finalstate:
        empty = 0
        dot = 0
        cross = 0
        if (i, j) in _:
            for k in _:
                if brd[k[0]][k[1]] == '_':
                    empty += 1
                if brd[k[0]][k[1]] == 'o':
                    dot += 1
                if brd[k[0]][k[1]] == 'x':
                    cross += 1
        if maxElement[2] < cross:
            maxElement = [i, j, cross]

    return maxElement


def algo():
    fvalueList = []
    for i in range(3):
        for j in range(3):
            if brd[i][j] == '_':
                brd[i][j] = 'o'
```

```
                fvalueList.append(calculate_F_Value(i, j))
                brd[i][j] = '_'
    position = max(fvalueList, key=lambda x: x[2])
    brd[position[0]][position[1]] = 'o'

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

    # when it's a draw
    if counter == 9:
        flagH = "Draw"

    for location in finalstate:

        if brd[location[0][0]][location[0][1]] == 'x' and
brd[location[1][0]][location[1][1]] == 'x' and
brd[location[2][0]][location[2][1]] == 'x':
            flagH = True
            break
        elif brd[location[0][0]][location[0][1]] == 'o' and
brd[location[1][0]][location[1][1]] == 'o' and
brd[location[2][0]][location[2][1]] == 'o':
            flagH = False
            break
    return flagH

# function for printing the board
def printboard():
    print(brd[0][0], "|", brd[0][1], "|", brd[0][2])
    print(brd[1][0], "|", brd[1][1], "|", brd[1][2])
    print(brd[2][0], "|", brd[2][1], "|", brd[2][2])
    print("\n\n")

end = False
#Print empty board
print("Let's play Tic Tac Toe!!!")

print("You are X !!!")

while True:
    print("Enter your move:")
    row=int(input("Enter the row number"))
    column=int(input("Enter the column number"))
    user = [row,column]
    user = [user[0] - 1, user[1] - 1]

    if brd[user[0]][user[1]] != '_':
        print("Play again. The spot is already occupied!!")
```

```
        continue

    brd[user[0]][user[1]] = 'x'

    print("Your move")
    printboard()

    Status = checkifWon()
    if Status == True:
        print("You won the game!!")
        end= True
        break
    elif Status == False:
        print("You lost the game!!")
        end= True
        break
    elif Status == "Draw":
        print("It's a Draw!!")
        end= True
        break

    if not end: algo()

    print("Computer's move")
    printboard()
    Status = checkifWon()
    if Status == True:
        print("You won the game!!")
        break
    elif Status == False:
        print("You lost the game!")
        break
    elif Status == "Draw":
        print("It's a Draw!")
```

Output:

```
Let's play Tic Tac Toe!!!
You are X !!!
Enter your move:
Enter the row number2
Enter the column number2
Your move
_ | _ | _
_ | x | _
_ | _ | _


Computer's move
o | _ | _
_ | x | _
_ | _ | _


Enter your move:
Enter the row number3
Enter the column number1
Your move
o | _ | _
_ | x | _
x | _ | _


Computer's move
o | _ | o
_ | x | _
x | _ | _


Enter your move:
Enter the row number1
Enter the column number2
Your move
o | x | o
_ | x | _
x | _ | _
```

```
Computer's move
o | x | o
_ | x | _
x | o | _


Enter your move:
Enter the row number2
Enter the column number1
Your move
o | x | o
x | x | _
x | o | _


Computer's move
o | x | o
x | x | o
x | o | _


Enter your move:
Enter the row number3
Enter the column number3
Your move
o | x | o
x | x | o
x | o | x


It's a Draw!!
```

Conclusion:

In this experiment, I have implemented the Tic-Tac-Toe game in python using A* search strategy. The code can demonstrate a user vs computer. Since A* is used for the computer's move, it is always the optimal move. The Agent uses number of crosses as heuristic value. The agent checks for all the possible available move and chooses one. The user is assigned "X". the computer plays with "O". A grid is displayed on the screen, indicating the move played by the user and then by the computer. To play a move by the user, the user needs to enter the location (i.e, the coordinate where he wants to place the x). To do so, the user needs to enter the row number and then the column number. Play the game till the end. The result of the game is then evaluated and displayed on the screen.