AI - Lab Test - 1

Name - Divyakriti Masaun
USN - 1BM18CS029
Date - 10/11/20

Q2) Tic-Tac-Toe

(Computer vs Computer)

```
import numpy as np
import random
from time import sleep

# creating an empty board
def create_board():
    return (np.array([[0,0,0],
                      [0,0,0],
                      [0,0,0]]))

# check for empty places on board
def possibilities(board):
    l = []
    for i in range(len(board)):
        for j in range(len(board)):
            if board[i][j] == 0:
                l.append((i,j))
    return(l)
```


Divyakriti

Divyakriti Masaun
1BM18CS029

selecting a random ~~spot~~ place for the player.

```
def random_place(board, player):
    selection = possibilities(board)
    current_loc = random.choice(selection)
    board[current_loc] = player
    return(board)
```

checking whether the player has three of their marks in a horizontal row.

```
def row_win(board, player):
    for x in range(len(board)):
    win = True
        for y in range(len(board)):
            if board[x,y] != player:
                win = False
                continue
            if win == True:
                return(win)
        return(win)
```

~~# etc~~

Divyakriti

Divyakriti Masaun
1BM19CS029

checking whether the player has three of their marks in a vertical row.

```
def col_win(board, player):
    for x in range(len(board)):
        win = True

    for y in range(len(board)):
        if board[y][x] != player:
            win = False

            continue
    if win == True:
        return(win)
    return(win)
```

checks whether the player has three of their marks in a diagonal row.

```
def diag_win(board, player):
    win = True
    y = 0
    for x in range(len(board)):
        if board[x, x] != player:
            win = False

    if win:
        return win
    win = True
```

Divyakriti Masaun
1BM18CS029

```
if win:
    for x in range(len(board)):
        y = len(board) - 1 - x
        if board[x, y] != player:
            win = False
    return win
```

Evaluating whether there is a winner or it is a tie.

```
def evaluate(board):
    winner = 0

    for player in [1, 2]:
        if (row_win(board, player) or
            col_win(board, player) or
            diag_win(board, player)):

            winner = player
    if np.all(board != 0) and winner == 0:
        winner = -1
    return winner
```


Divyakriti

AI-Lab-test-1

Divyakriti Masaun
1BM18CS029

```
# main function
def play-game():
    board, winner, counter = create-board(), 0, 1
    print(board)
    sleep(2)
while winner == 0:
    for player in [1, 2]:
        board = random-place(board, player)
        print("Board after" + str(counter) + "more")
        print(board)
        sleep(2)
        counter += 1
        winner = evaluate(board)
        if winner != 0:
            break
    return(winner)

print("winner is:" + str(play-game()))
```


Divyakriti