VIT Vidyalankar Institute of Technology Accredited A+ by NAAC	**Department of Electronics Engineering (SBL Mini Project)**

Semester	S.E. Semester IV– Electronics Engineering
Subject	Skill Lab: Python Programming
Subject Professor In-charge	Prof. Manoj Suryawanshi
Lab Teacher	Prof. Manoj Suryawanshi
Laboratory	Online

1: Student Name	Pratyush Raut	
Roll Number	20103A0071	
2: Student Name	Ronak Nawal	
Roll Number	20103A0080	
Grade and Subject Teacher's Signature		

Problem Title	Flipping Tiles
Introduction	Flipping tiles game can be played to test our memory. In this, we have a certain even number of tiles, in which each number/figure has a pair. The tiles are facing downwards, and we have to flip them to see them. In a turn, one flips 2 tiles, if the tiles match then they are removed. If not then they are flipped and placed back in the position. We keep on doing this until all the tiles have been matched and removed.
Objective of the Project	It will sharpen the mind.
Google Drive link	

Program:

```
import random


class TicTacToe:

    def __init__(self):
        self.board = []

    def create_board(self):
        for i in range(3):
            row = []
            for j in range(3):
                row.append('-')
            self.board.append(row)

    def get_random_first_player(self):
        return random.randint(0, 1)

    def fix_spot(self, row, col, player):
        self.board[row][col] = player

    def is_player_win(self, player):
        win = None

        n = len(self.board)

        # checking rows
```

```
        for i in range(n):
            win = True
            for j in range(n):
                if self.board[i][j] != player:
                    win = False
                    break
            if win:
                return win

        # checking columns
        for i in range(n):
            win = True
            for j in range(n):
                if self.board[j][i] != player:
                    win = False
                    break
            if win:
                return win

        # checking diagonals
        win = True
        for i in range(n):
            if self.board[i][i] != player:
                win = False
                break
        if win:
            return win

        win = True
        for i in range(n):
            if self.board[i][n - 1 - i] != player:
                win = False
                break
        if win:
            return win
        return False

        for row in self.board:
            for item in row:
                if item == '-':
                    return False
        return True

    def is_board_filled(self):
        for row in self.board:
            for item in row:
                if item == '-':
                    return False
        return True

    def swap_player_turn(self, player):
        return 'X' if player == 'O' else 'O'

    def show_board(self):
```

```
        for row in self.board:
            for item in row:
                print(item, end=" ")
            print()

    def start(self):
        self.create_board()

        player = 'X' if self.get_random_first_player() == 1 else 'O'
        while True:
            print(f"Player {player} turn")

            self.show_board()

            # taking user input
            row, col = list(
                map(int, input("Enter row and column numbers to fix spot: ").split()))
            print()

            # fixing the spot
            self.fix_spot(row - 1, col - 1, player)

            # checking whether current player is won or not
            if self.is_player_win(player):
                print(f"Player {player} wins the game!")
                break

            # checking whether the game is draw or not
            if self.is_board_filled():
                print("Match Draw!")
                break

            # swapping the turn
            player = self.swap_player_turn(player)

        # showing the final view of board
        print()
        self.show_board()


# starting the game
tic_tac_toe = TicTacToe()
tic_tac_toe.start()
```

Results: (Attach screen shots of the output results)