PRACTICAL NO : 02
NAME : SHINDE ATHARVA BHAUSAHEB
ROLL NO : 62

CODE :
```python
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
```

```python
    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:
\n").split()))
            print()

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

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

            # checking whether the game is a 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 the board
            print()
            self.show_board()


# Starting the game
tic_tac_toe = TicTacToe()
tic_tac_toe.start()


OUTPUT :

/usr/bin/python3.10 /home/pvg-aids-ai/.config/JetBrains/PyCharmCE2023.2/scratches/
scratch_2.py
Player O turn
- - -
- - -
- - -
Enter row and column numbers to fix spot:
1 2
```

```
- 0 -
- - -
- - -
Player X turn
- 0 -
- - -
- - -
Enter row and column numbers to fix spot:
2 3


- 0 -
- - X
- - -
Player 0 turn
- 0 -
- - X
- - -
Enter row and column numbers to fix spot:
3 3


- 0 -
- - X
- - 0
Player X turn
- 0 -
- - X
- - 0
Enter row and column numbers to fix spot:
3 2


- 0 -
- - X
- X 0
Player 0 turn
- 0 -
- - X
- X 0
Enter row and column numbers to fix spot:
2 1


- 0 -
0 - X
- X 0
Player X turn
- 0 -
0 - X
- X 0
Enter row and column numbers to fix spot:
3 1


- 0 -
0 - X
X X 0
Player 0 turn
- 0 -
0 - X
X X 0
Enter row and column numbers to fix spot:
1 1
```

```
O O -
O - X
X X O
Player X turn
O O -
O - X
X X O
Enter row and column numbers to fix spot:
2 2


O O -
O X X
X X O
Player O turn
O O -
O X X
X X O
Enter row and column numbers to fix spot:
1 3

Player O wins the game!

Process finished with exit code 0
```