Tic_tac_toe_game

June 1, 2023

1 Tic Tac Toe

Let us create a simple Tic Tac Toe game in Python. It will help you to build up game logic and understand how to structure code.

2 Introduction :

There will be two players in a game. Two signs represent each player. The general signs used in the game are X and O. Finally, there will be a board with 9 boxes.

3 The game play will be as follows.

1)one user will place their sign in one of the available empty boxes. 2)the second user will place their sign in one of the available empty boxes. 3)The goal of the players is to place their respective signs completely row-wise or column-wise, or diagonally. 4)The game goes on until a player wins the game or it ended up in a draw by filling all boxes without a winning match.

[3]:
```
import random
```

Create a board using a 2-dimensional array and initialize each element as empty. You can represent empty using any symbol you like. Here, we are going to use a hyphen. '-'.

then Write a function to check whether the board is filled or not. Iterate over the board and return false if the board contains an empty sign or else return true.

[14]:
```
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)
#Write a function to start the game.

    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: ").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()
```

```
Player X turn
- - -
- - -
- - -
Enter row and column numbers: 1 1

Player O turn
X - -
- - -
- - -
Enter row and column numbers: 2 1

Player X turn
X - -
O - -
- - -
Enter row and column numbers: 3 2

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

Player X turn
X - O
O - -
- X -
Enter row and column numbers: 2 3
```

```
Player O turn
X - O
O - X
- X -
Enter row and column numbers: 3 3

Player X turn
X - O
O - X
- X O
Enter row and column numbers: 1 2

Player O turn
X X O
O - X
- X O
Enter row and column numbers: 2 2

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

Match Draw!

X X O
O O X
X X O
```