Project(Tic Tac Toe)

Date:- 13.03.2023

Team:- 2

Team Members

Divya Bhupatbhai Lakhani

Nirav Arvindbhai Patel

Nikhil

Le Nhat Minh Quan

Kushani Lakshika Gimhani Abeykoon Mayandige

Description of Code

This code creates a game program called Tic-Tac-Toe.

Step1 - Import Libraries

In this step, we import the necessary libraries that we will use in our Tic-Tac-Toe game. We are importing the random and itertools libraries.

In [ ]:
```
import random
import itertools
```

Step2 - Create Board

In this step, we create a board for our Tic-Tac-Toe game. We are using a list of numbers from 1 to 9 to represent our game board. Each number on the board corresponds to a position on the board where players can make their moves.

In [ ]:
```
board = list(range(1, 10))
```

Step3 - Create function to print board

In this step, we define a function called "show_board" that takes a list (our game board) as an argument and prints it to the console. The function displays the current state of the game board.

In [ ]:
```
def show_board(board):
    print(str(board[0]) + ' | ' + str(board[1]) + ' | ' + str(board[2]))
    print('----------')
    print(str(board[3]) + ' | ' + str(board[4]) + ' | ' + str(board[5]))
    print('----------')
    print(str(board[6]) + ' | ' + str(board[7]) + ' | ' + str(board[8]))
```

Step4 - Create function to determine winner

In this step, we define a function called "find_winner" that takes a list (our game board) as an argument and determines if there is a winner. The function checks all possible combinations of winning moves and returns the winning player's token ("X" or "O") or None if there is no winner.

In [ ]:
```
def find_winner(board):
    combos = ([(0,1,2),(3,4,5),(6,7,8),(0,3,6),(1,4,7),(2,5,8),(0,4,8),(2,4,6)])
    for player in ['X','O']:
        positions = [i for i, x in enumerate(board) if x == player]
        for combo in combos:
            win = True
            for pos in combo:
                if pos not in positions:
                    win = False
```

```
            if win == True:
                return player
    return None
```

Step5 - Create function to take input from players

In this step, we define a function called "take_input" that takes the current player's token as an argument and prompts the player to enter their move. The function checks if the move is valid and updates the game board accordingly.

In [ ]:
```
def take_input(player_token):
    valid = False
    while not valid:
        player_answer = input('Player %s, please enter your move: (1-9) ' % player_token)
        try:
            player_answer = int(player_answer)
        except:
            print('Invalid input, please try again.')
            continue
        if player_answer >= 1 and player_answer <= 9:
            if (str(board[player_answer-1]) not in 'XO'):
                board[player_answer-1] = player_token
                valid = True
            else:
                print("Invalid move, please try again")
        else:
            print("Invalid move, please try again")
```

Step6 - Create function to switch player

In this step, we define a function called "switch_player" that takes the current player's token as an argument and returns the other player's token. This function is used to switch between the players after each move.

In [ ]:
```
def switch_player(player_token):
    if player_token == 'X':
        return 'O'
    else:
        return 'X'
```

Step7 - Create main function to run the game

In this step, we define the main function called "play_game" that runs the Tic-Tac-Toe game. The function uses a while loop to keep the game running until there is a winner or a tie. It calls the "take_input", "show_board", and "find_winner" functions and switches between the players after each move.

In [ ]:
```
def play_game():
    show_board(board)
    player_token = 'X'
    winner = None
    while not winner:
        take_input(player_token)
        show_board(board)
        winner = find_winner(board)
        player_token = switch_player(player_token)
    if winner == 'X' or winner == 'O':
        print('Player %s wins!' % winner)
    else:
        print('Tie Game!')
```

Step8 - Run the game

In this step, we call the "play_game" function to start the Tic-Tac-Toe game. This is the entry point of our program, and it starts the game loop.

In [ ]:
```
play_game()
```

CODE :

In [1]:
```
## # Tic-Tac-Toe

# Step1 - Import Libraries
import random
import itertools
```

```
# Step2 - Create Board
board = list(range(1, 10))

# Step3 - Create function to print board
def show_board(board):
    print(str(board[0]) + ' | ' + str(board[1]) + ' | ' + str(board[2]))
    print('----------')
    print(str(board[3]) + ' | ' + str(board[4]) + ' | ' + str(board[5]))
    print('----------')
    print(str(board[6]) + ' | ' + str(board[7]) + ' | ' + str(board[8]))

# Step4 - Create function to determine winner
def find_winner(board):
    combos = ([(0,1,2),(3,4,5),(6,7,8),(0,3,6),(1,4,7),(2,5,8),(0,4,8),(2,4,6)])
    for player in ['X','O']:
        positions = [i for i, x in enumerate(board) if x == player]
        for combo in combos:
            win = True
            for pos in combo:
                if pos not in positions:
                    win = False
            if win == True:
                return player
    return None

# Step5 - Create function to take input from players
def take_input(player_token):
    valid = False
    while not valid:
        player_answer = input('Player %s, please enter your move: (1-9) ' % player_token)
        try:
            player_answer = int(player_answer)
        except:
            print('Invalid input, please try again.')
            continue
        if player_answer >= 1 and player_answer <= 9:
            if (str(board[player_answer-1]) not in 'XO'):
                board[player_answer-1] = player_token
                valid = True
            else:
                print("Invalid move, please try again")
        else:
            print("Invalid move, please try again")

# Step6 - Create function to switch player
def switch_player(player_token):
    if player_token == 'X':
        return 'O'
    else:
        return 'X'

# Step7 - Create main function to run the game
def play_game():
    show_board(board)
    player_token = 'X'
    winner = None
    while not winner:
        take_input(player_token)
        show_board(board)
        winner = find_winner(board)
        player_token = switch_player(player_token)
        if all(str(x) in 'XO' for x in board):
            break
    if winner == 'X' or winner == 'O':
        print('Player %s wins!' % winner)
    else:
        print('Tie Game!')

# Step8 - Run the game
play_game()
```

Link to this Project's GitHub Repository:

https://github.com/NikhilNikhil844/Team-2-Nikhil-CS-Project-Tic-tac-toe.git

```
1 | 2 | 3
----------
4 | 5 | 6
----------
7 | 8 | 9
Player X, please enter your move: (1-9) 1
X | 2 | 3
----------
4 | 5 | 6
----------
7 | 8 | 9
Player O, please enter your move: (1-9) 1
Invalid move, please try again
Player O, please enter your move: (1-9) 2
X | O | 3
----------
4 | 5 | 6
----------
7 | 8 | 9
Player X, please enter your move: (1-9) 3
X | O | X
----------
4 | 5 | 6
----------
7 | 8 | 9
Player O, please enter your move: (1-9) 4
X | O | X
----------
O | 5 | 6
----------
7 | 8 | 9
Player X, please enter your move: (1-9) 5
X | O | X
----------
O | X | 6
----------
7 | 8 | 9
Player O, please enter your move: (1-9) 5
Invalid move, please try again
Player O, please enter your move: (1-9)
Invalid input, please try again.
Player O, please enter your move: (1-9) 5
Invalid move, please try again
Player O, please enter your move: (1-9) i
Invalid input, please try again.
Player O, please enter your move: (1-9) o
Invalid input, please try again.
Player O, please enter your move: (1-9) *
Invalid input, please try again.
Player O, please enter your move: (1-9) 6
X | O | X
----------
O | X | O
----------
7 | 8 | 9
Player X, please enter your move: (1-9) 7
X | O | X
----------
O | X | O
----------
X | 8 | 9
Player X wins!
```