Nama: Moh.Rangga Wijaya Fadilah
NPM: 227006416144
Mata Kuliah: Pemrograman Berorientasi Objek

Tic-Tac-Toe (3x3 Grid) Game with Human vs. Computer

Project Overview

This project is a Python console game that simulates the classic **Tic-Tac-Toe (3x3 Grid)** game using **Object-Oriented Programming (OOP)** concepts. The game is built with clear class separation (Move, Player, Board) to provide a structured and modular architecture.

A human player competes against a computer player. The first player to complete a full row, column, or diagonal with their marker wins. If the board is completely filled without a winner, the game ends in a tie.

Players can play multiple rounds in a single session, with the option to continue after each completed game.

General Requirements

- A 3x3 grid is displayed with numbered positions from 1 to 9.
- Two players:
  - **Human Player** (manual input)
  - **Computer Player** (random choice)
- Each player takes turns to place their marker ('X' for human, 'O' for computer).
- If a player selects an already occupied position, the turn is skipped.
- The game checks for:
- Win conditions: 3 identical markers in a row, column, or diagonal.
- Tie condition: board is full and no winner.
- The player is prompted to play again or exit after the game concludes.

****

Welcome to Tic-Tac-Toe

****

Positions:

| 1 | 2 | 3 |

| 4 | 5 | 6 |

| 7 | 8 | 9 |

Board:

| | | |

| | | |

| | | |

Please enter your move (1-9):

Class Descriptions

Class: Move

Represents a move on the board.

- **Attributes:**
    - position: An integer from 1–9 indicating where to place the marker.
    - marker: 'X' or 'O', depending on the player.

Class: Player

Handles player attributes and behavior.

- **Attributes:**
    - name: Display name (e.g., "You" or "Computer").
    - marker: 'X' or 'O'.

- is_human: Boolean indicating if the player is a human.

- **Methods:**
  - **make_move(board):**
    ➤ If human: prompts for input and validates it.
    ➤ If computer: randomly selects an available spot.

Class: Board

Manages the state and display of the game board.

- **Attributes:**
  - cells: A list of 9 strings representing the 3x3 board cells.
- **Methods:**
  - display_positions(): Displays the number guide (1–9).
  - display(): Shows the current board with markers.
  - update(move): Updates a board cell based on a valid move.
  - is_position_free(position): Checks if a cell is unoccupied.
  - get_free_positions(): Returns list of available cell indices.
  - check_winner(marker): Returns True if the player wins.
  - is_full(): Checks if the board is full.

Game Flow

1. Display welcome screen and position guide.
2. Alternate turns between players.
3. Players place markers in available positions.
4. After each move:

   Check for winner.

   Check for tie.

   Display updated board.

5. End the game with:

   Win message: "🏆 [Player] wins the game!"

   Tie message: "🤝 It's a tie!"

6. Ask if the player wants to play again.