Description:

This code is a simple implementation of a Tic-Tac-Toe game using Pygame. Here’s a breakdown of how it works:

# Imports

- pygame: The library used for creating the game window and handling graphics.

- sys: Used for exiting the program.

- time: Used for adding delays.

- tictactoe as ttt: Importing a custom Tic-Tac-Toe module that presumably contains functions and constants used in the game.

# Initialization

- pygame.init(): Initializes all imported Pygame modules.

- size defines the dimensions of the game window (600x400).

- black and white: RGB tuples representing colors used in the game.

- screen: Sets up the display window.

- mediumFont, largeFont, moveFont: Fonts of different sizes for rendering text.

# Game State Variables

- user: Keeps track of which player the user has chosen (X or O).

- board: Initializes the game board using a function from the ttt module.

- ai_turn: A boolean flag to manage turns between the user and AI.

# Main Game Loop

The main loop handles events and game logic:

1. Event Handling: Checks for user inputs and exits the game if the window is closed.

2. Screen Setup: Clears the screen and fills it with black color.

3. User Player Selection:

   - If user is None, the game prompts the user to choose between playing as X or O. This is done by rendering buttons and checking for mouse clicks.

4. Game Board Display:

   - If the user has selected a player, the game board is drawn.

   - Each tile is rendered as a rectangle, and any existing moves (X or O) are displayed in the corresponding tiles.

5. Game Status:

- Checks if the game is over and displays the appropriate message (winner or tie).

- If the AI's turn is next, it calculates the move using a minimax algorithm from the ttt module.

6. User Moves:

- If it's the user's turn, it checks for mouse clicks on the board. If the user clicks an empty tile, the board is updated.

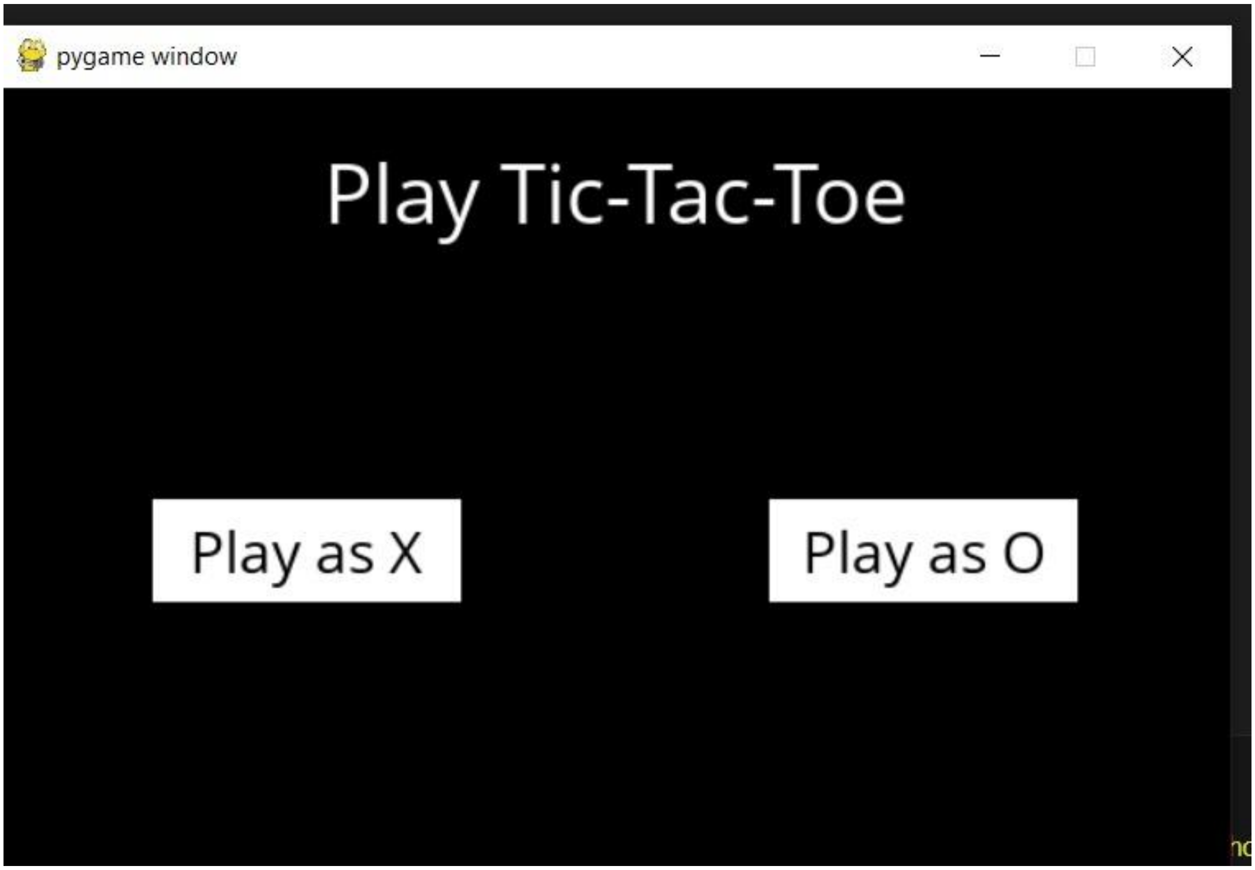
7. Restart Option:

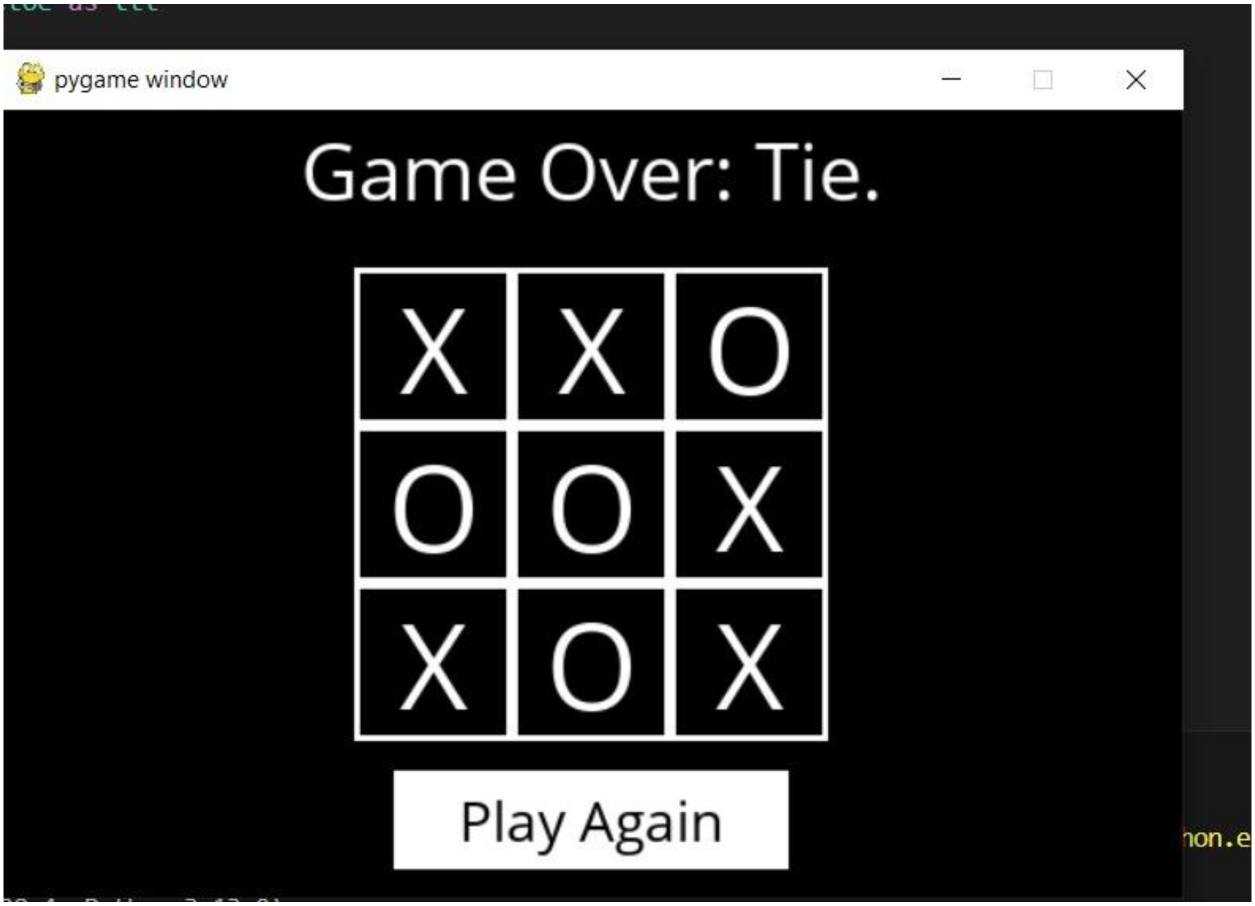
- When the game ends, a "Play Again" button is displayed. If clicked, it resets the game state.

8. Update Display: The screen is updated with pygame.display.flip() at the end of each loop iteration.

Overall, this code sets up a playable Tic-Tac-Toe game where a user can play against an AI. It handles player selection, drawing the game board, checking for game status, and allowing users to restart the game after it concludes.

Output:





Game Over: Tie.

X	X	O
O	O	X
X	O	X