### **Project: Tic Tac Toe Game**

#### **Overview**

This project is a browser-based Tic Tac Toe game designed for two players. The interface allows users to input their names and compete on a 3x3 grid. The game keeps track of the players' wins, and tied games, and provides options for resetting, rematching, or starting a new game with different players. The project is built using HTML for structure, CSS for styling, and JavaScript (with jQuery) for game logic and interaction.

#### **Technologies Used**

* **HTML**: Provides the basic structure of the web page, defining the game layout and input forms.
* **CSS**: Adds visual styles, including layout, colors, fonts, and animations, to enhance the user experience.
* **JavaScript**: Implements the core game logic, including turn management, win detection, and user interactions.
* **jQuery**: Simplifies DOM manipulation, event handling, and animations.

#### **File Descriptions**

1. **index.html**
   * This is the main HTML file that sets up the structure of the Tic Tac Toe game.
   * Contains two sections: the initial player input form and the game board that appears after players start the game.
   * Includes references to external CSS (style1.css) and JavaScript (script.js) files for styling and functionality.
2. **style1.css**
   * Handles all the visual styling of the game.
   * Applies a background image, fonts, colors, and button styles.
   * Uses CSS transitions and transforms for smooth animations when interacting with buttons.
   * Ensures the game is responsive and centered on the screen regardless of the device size.
3. **script.js**
   * Contains the game logic and handles user interactions.
   * Manages the game flow, including turn-taking, win detection, and score tracking.
   * Uses jQuery to manipulate the DOM, such as updating the display text, enabling/disabling buttons, and showing/hiding elements.

#### **HTML Structure (index.html)**

1. **Head Section**
   * Includes meta tags for character set and viewport settings, ensuring the page is responsive and correctly displays text.
   * Links to the external CSS (style1.css) and jQuery library.
   * Contains the page title "Tic Tac".
2. **Body Section**
   * **Player Input Form**: A div with id open where players enter their names.
     + Two input fields for the players to enter their names.
     + A "Play" button to start the game and a "Reset" button to clear the input fields.
   * **Game Container**: A div with class container that contains the game board and controls.
     + **Heading (#heading)**: Displays the current player's turn or the game result.
     + **Game Board (.game)**: A 3x3 grid of buttons (.box) where players click to make their moves.
     + **Win Count (#win\_count)**: Displays the score, showing how many games each player has won and how many games have been tied.
     + **Control Buttons**: "Re-Match" and "New Game" buttons to start a new round with the same or different players, respectively.

#### **CSS Styling (style1.css)**

1. **Global Styles**
   * The body is styled with a full-screen background image and centered content using flexbox.
   * The font family is set to a monospaced font for a classic, game-like feel.
2. **Player Input Form (#open)**
   * Background is a gradient overlay with a background image.
   * Large, easy-to-read text with text shadow for a glowing effect.
   * Input fields and buttons are styled with smooth transitions and slight hover effects to enhance interactivity.
3. **Game Board and Controls**
   * **Heading (#heading)**: Styled with large, centered text, a background color, and rounded corners. Changes color when a player wins or a game is tied.
   * **Win Count (#win\_count)**: Similar styling to the heading, displaying the scores of both players and tied games.
   * **Game Board (.game)**: A flexbox grid that holds the buttons representing the Tic Tac Toe grid. Buttons are spaced evenly with a gap between them.
   * **Buttons (.box)**: Each box is styled with a background color, large font for the marks ("O" and "X"), and hover effects that shrink and dim the button slightly.
   * **Control Buttons (#rematch, #new)**: Styled similarly to the input buttons with added hover effects, changing background color and text shadow on hover.

#### **JavaScript Functionality (script.js)**

1. **Game Variables**
   * **boxes**: A jQuery selector that references all the buttons in the game grid.
   * **turn**: A boolean that tracks the current player's turn. true represents Player 1 (O) and false represents Player 2 (X).
   * **count**: A counter that tracks the number of moves made during the game.
   * **win1, win2, tied**: Variables that track the number of wins for Player 1, Player 2, and tied games, respectively.
   * **win**: An array containing the possible winning combinations on the Tic Tac Toe grid.
2. **Document Ready Function**
   * Initializes the game when the document is fully loaded.
   * Hides the game board initially and waits for the players to input their names.
3. **Start Game (#play Button)**
   * On clicking the "Play" button, the game checks if the player names are valid (not empty and not the same).
   * If valid, the input form fades out, and the game board fades in, with the first player's turn displayed.
4. **Winner Detection**
   * After each move, the winner() function checks all possible winning combinations.
   * If a winning combination is found, the game declares the winner by updating the heading text and disabling all grid buttons.
   * If all grid spaces are filled without a winner, the game declares a tie.
   * The score is updated and displayed after each game.
5. **Move Handling (.box Click)**
   * When a player clicks on an empty grid space, their mark ("O" or "X") is placed.
   * The turn is switched to the other player, and the game checks for a winner.
   * The grid space is disabled after a mark is placed to prevent further clicks.
6. **New Game (#new Button)**
   * Resets the game by clearing the grid, resetting the scores, and returning to the player input form.
   * The game board fades out, and the input form fades in.
7. **Rematch (#rematch Button)**
   * Clears the grid while keeping the current scores.
   * The game is reset to start with the same players.
8. **Reset Input Fields (#reset Button)**
   * Clears the player name input fields, allowing players to enter new names.

#### **Features**

* **Interactive UI**: Players are guided through the game with visual and textual cues.
* **Responsive Design**: The game is designed to work on various screen sizes, maintaining its layout and usability.
* **Dynamic Turn Management**: The current player’s turn is clearly indicated, and the turn alternates automatically after each move.
* **Score Tracking**: The game keeps track of the number of wins for each player and the number of tied games.
* **Game Reset and Rematch**: Players can reset the game to start fresh or rematch with the same players, keeping the current score.
* **Input Validation**: The game ensures that both players have different names before starting.

#### **How to Run**

1. Clone or download the project files to your local machine.
2. Open the index.html file in any modern web browser.
3. Enter the names of both players and click "Play" to start the game.
4. Players take turns clicking on the grid to place their mark. The game automatically detects a win or a tie.
5. Use the "Re-Match" button to play another round with the same players or the "New Game" button to start over with new names.
6. If you need to clear the input fields before starting, use the "Reset" button.

#### **Future Enhancements**

* **Single Player Mode**: Add an AI opponent for a single-player experience.
* **Animations**: Introduce more dynamic animations for moves and winning sequences.
* **Leaderboard**: Implement a leaderboard to track the scores across multiple games.
* **Sound Effects**: Add sound effects to enhance the game experience.