**Trivia Game Documentation**

**Project Overview**

This project aims to create a competitive two-player trivia game that fetches questions from The Trivia API. Players will input their names, choose categories together, and answer questions of varying difficulty. The game will maintain a score system and declare a winner at the end. It features a simple yet engaging interface with question rounds, score tracking, and dynamic category selection.

**Technical Stack**

1. **HTML**: Used for the structure and layout of the game interface.
2. **CSS**: Provides the basic styling and responsiveness for the game.
3. **JavaScript**: Handles the game logic, DOM manipulation, and interactions between the game and the user.
4. **The Trivia API**: The API is used to fetch questions based on selected categories and difficulty levels.

**Data Structure for the Game**

**1. Player Object**

Each player has the following attributes:

* **Name**: The player's name.
* **Score**: The player's accumulated score throughout the game.

**2. Question Object**

Each question has the following attributes:

* **Category**: The category of the question (e.g., Science, History, etc.).
* **Difficulty**: The difficulty level of the question (easy, medium, hard).
* **Question**: The actual question text.
* **Correct Answer**: The correct answer to the question.
* **Incorrect Answers**: A list of incorrect answers to the question.

**3. Game Start Status Object**

This object holds the status of the game and the game state:

* **Players**: An array of player objects.
* **Current Player**: An index indicating the current player (0 for Player 1, 1 for Player 2).
* **Current Category**: The category selected by the players.
* **Questions**: The list of questions fetched for the current category.
* **Current Question**: The index of the question being asked from the list of questions.
* **Used Categories**: An array that keeps track of categories that have already been used.

**Game Flow**

**Game Initialization**

1. **Get Player Names**:
   * Players enter their names via text input fields on the initial screen.
   * The names are saved to the player objects.
2. **Initialize Game Start Status**:
   * A status object is initialized to store player data, current game state, and progress.
3. **Fetch Available Categories**:
   * Categories are fetched from The Trivia API to allow players to choose from a list of available options.

**Category Selection**

1. **Display Available Categories**:
   * Categories fetched from the API are displayed as buttons for selection.
2. **Allow Players to Select a Category**:
   * Players can select a category by clicking a button associated with each category.
3. **Fetch Questions for Selected Category**:
   * Once a category is selected, questions related to that category are fetched from the API.

**Question Round**

1. **Display Question**:
   * The current player's turn is displayed.
   * The current question is shown with possible answer options.
2. **Accept and Validate Player’s Answer**:
   * Players select an answer from multiple options.
   * The selected answer is validated against the correct answer.
3. **Update Score Based on Correctness**:
   * If the answer is correct, the player’s score is updated based on the difficulty of the question.
     + Easy questions add 10 points.
     + Medium questions add 15 points.
     + Hard questions add 20 points.
4. **Switch to Next Player and Question**:
   * After the current player answers, the game switches to the other player, and the next question is displayed.

**After Category Completion**

1. **Display Category Completion Message**:
   * After all questions for a category have been answered (6 questions), a message is displayed indicating that the category has been completed.
2. **Prompt to Choose New Category or End Game**:
   * Players can either select a new category to continue or end the game.
3. **If New Category Selected**:
   * The game returns to the category selection screen to choose another category.
4. **If End Game Selected**:
   * The game proceeds to the game over screen to show the final results.

**Game Over**

1. **Calculate and Display Final Scores**:
   * The game calculates each player's total score and displays it.
2. **Declare Winner**:
   * The player with the highest score is declared the winner.
   * If both players have the same score, the game results in a tie.
3. **Display Option to Start a New Game**:
   * An option to start a new game is provided on the game over screen.