Documentation: Who Wants To Be A Millionaire

Here is the documentation of my project. This is my 1st ever game in Python and 3rd game created in Programming world. Hence, I am glad to share this with you. I have used Chatbot for this documentation to make it more readable and quicker, but I made sure that the text curated by the chatbot is correct as I provided. Please let me know if there is any issue with this so that I can redo this step. Thank you😃

**Topic**: "Who Wants to Be a Millionaire" Game

**Description**: The project is a Python-based implementation of the popular quiz game "Who Wants to Be a Millionaire". Players answer multiple-choice questions to earn points, and they can use lifelines like "50/50" and "Expert's Opinion" to help them answer difficult questions. I had two options in this project; create a textual result that is based on the black and white screen, create a textual application, create a graphical application. Since I had the practice in creating simple applications, I opted for that and due to no practice with graphics in programming, I avoided that.

**The Solution Principle of the Project**

The project is built using Python's tkinter library to create a graphical user interface (GUI). The main principle is to present a series of questions to the player, track their progress, and provide interactive elements like buttons for selecting answers and using lifelines. The player's performance is recorded and displayed at the end of the game.

**The Structure of the Project: How It Is Divided into Functions**

* **start\_game**: Starts the game and displays the first question.
* **next\_question**: Displays the next question and updates the interface.
* **check\_answer**: Checks if the player's answer is correct and updates the score.
* **end\_game**: Ends the game, displays the final score, and writes the results to a file.
* **fifty\_fifty**: Implements the "50/50" lifeline by disabling two incorrect options.
* **expert\_opinion**: Provides a hint for the current question as part of the "Expert's Opinion" lifeline.

**The Functions Used and Their Interrelationships**

* **start\_game:** Kicks off the game and calls next\_question.
* **next\_question:** Displays the current question and options, and sets up the buttons to call check\_answer.
* **check\_answer:** Validates the player's answer, updates the score, and either calls next\_question for the next question or end\_game if the game is over.
* **end\_game:** Finalizes the game, saves the results, and displays the end screen.
* **fifty\_fifty:** Disables two incorrect answers, providing a hint.
* **expert\_opinion:** Displays a hint related to the current question.

These functions interact closely with each other to manage the game flow and user interactions. For example, next\_question relies on check\_answer to determine if the answer was correct before moving to the next question.

**The Use of Possible External Libraries**

Although, no external libraries are needed for the basic functionality of the game, but I will mention the general libraries I used:

* **tkinter**: Used for creating the GUI elements such as labels, buttons, and layout management.
* **random**: Used for generating random numbers to select questions and manage the "50/50" lifeline.
* **os**: Used for file operations, such as saving the game results to a text file.