Module 5 - Exercise 1 Plan - Dan Bailey

Overview

The goal is to design a Jeopardy game program and adhere to structured programming principles. The program will simulate a version of the Jeopardy game, where a contestant answers questions phrased as responses, and the program expects answers in the form of questions. The plan outlines the components, logic, and flow to ensure clarity and avoid redundancies.

Role

I will approach this as the moderator of the game, responsible for presenting questions, validating answers, tracking scores, and managing the game flow. This perspective makes sure the program controls the game environment while allowing user interaction as the contestant.

Program Components

1. **Question Bank**: A collection of questions organized by categories and difficulty levels, like on the show. (100, 200, 300 points). Each question includes:
    - A category
    - A point value.
    - The question text, phrased as a statement.
    - The correct answer, phrased as a question.
2. **Player Score**: Tracks the contestant's score, increasing for correct answers and decreasing for incorrect ones.
3. **Timing Constraints**: Each question has a time limit (e.g., 30 seconds) to simulate the real game's pressure.
4. **Game Flow**:
    - Display a game board with categories and point values.
    - Allow the contestant to select a category and point value.
    - Present the question and wait for the answer within the time limit.
    - Validate the answer and update the score.
    - Mark the question as answered to prevent reuse.
    - Continue until all questions are answered or the player chooses to end the game.

Logical Steps

1. **Initialize the Game**:

    - Load the question bank.
    - Set the initial score to 0.
    - Initialize a timer.
2. **Display Game Board**:
    - Show available categories and point values in a grid format.
    - Indicate which questions have been answered.
3. **Handle Player Input**:
    - Prompt the contestant to select a category and point value.
    - Validate the selection (ensure it's valid and not already answered).
4. **Output Question**:
    - Display the question text.
    - Start a 30-second timer.
    - Accept the contestant's answer as input.
5. **Validate Answer**:
    - Compare the input to the correct answer.
    - Award points for correct answers or deduct points for incorrect ones.
6. **Update Game State**:
    - Mark the question as answered.
    - Update the score.
    - Refresh the game board.
7. **End Game**:
    - Check if all questions are answered or if the player wants to quit.
    - Display the final score and end the program.