UML Class Diagram Breakdown

1. Game (Main Class for Game Logic)
   * Attributes:
     + chosenWord: String - A random 5-letter word selected from the list.
     + attemptsLeft: int - The number of remaining attempts (6 in total).
     + currentAttempt: int - The current attempt number (from 1 to 6).
     + feedback: List<String> - List to store feedback for each guess.
   * Methods:
     + startGame(): Initializes the game, selects a word, and resets attempts.
     + submitGuess(guess: String): Handles the logic when the user submits a guess.
     + checkGuess(guess: String): Checks if the guess matches the target word and provides feedback.
     + provideFeedback(guess: String): Returns feedback (correct, wrong-position, or incorrect).
     + endGame(won: boolean): Ends the game either when the word is guessed or attempts run out.
2. UIManager (Manages User Interface interactions)
   * Attributes:
     + board: HTMLDivElement - The HTML element that displays the board.
     + inputBox: HTMLInputElement - The HTML input field where the user types their guess.
     + submitButton: HTMLButtonElement - The button to submit guesses.
     + messageBox: HTMLElement - The element showing messages like win, loss, and instructions.
   * Methods:
     + renderBoard(): Displays the current state of the board after each guess.
     + updateCellFeedback(guess: String, attempt: int): Updates individual cells with the correct/incorrect feedback.
     + showMessage(message: String): Displays messages like "Correct!", "Game Over", etc.
3. WordList (Stores a list of possible 5-letter words)
   * Attributes:
     + words: Array<String> - List of 5-letter words to choose from.
   * Methods:
     + getRandomWord(): Randomly selects a word from the list.
4. Feedback (Handles feedback for each guess)
   * Attributes:
     + letter: String - The letter guessed.
     + status: String - The feedback for that letter (e.g., "correct", "wrong-position", "incorrect").
   * Methods:
     + generateFeedback(guess: String): Generates feedback based on comparing the guess to the chosen word.

How These Components Interact:

* Game:
  + Initializes the game by selecting a random word from the WordList.
  + The user’s guesses are submitted to the UIManager and checked with Game logic (via checkGuess()).
  + Feedback is provided to the UIManager, which updates the board with color feedback for each guess.
  + If the user guesses the word correctly or exhausts all attempts, the game ends and a message is displayed.
* UIManager:
  + Handles user input and displays the board after each guess.
  + Calls renderBoard() and updateCellFeedback() to update the UI dynamically.
* WordList:
  + Provides the list of possible words for the game to pick from.
* Feedback:
  + Provides the color-based feedback for each letter of the guess.