Notes on Generating Random Numbers in JavaScript

Overview

This document explains how to generate random integers in JavaScript to simulate a die throw (producing numbers 1 through 6), using the `Math.random()` method and additional operations to achieve the desired range.

Key Points

1. **Objective**:
    - Simulate a die throw by generating a random integer from 1 to 6.
    - Use JavaScript's `Math.random()` to produce pseudo-random numbers and transform them into the target range.
2. **The `Math.random()` Method**:
    - Generates a pseudo-random decimal number between 0 (inclusive) and 1 (exclusive), typically with 16 decimal places.
    - Example:

      ```
      var randomNumber = Math.random(); // e.g.,
         0.7474887706339359
      ```

    - **Range**: From 0.0000000000000000 to 0.9999999999999999.
3. **Scaling the Random Number**:
    - To generate numbers for a die (1–6), scale the random decimal by multiplying by 6:

      ```
      var bigDecimal = Math.random(); // 0 to 0.999...
      var scaled = bigDecimal * 6; // 0 to 5.999...
      ```

    - **Resulting Range**: 0.0000000000000000 to 5.9999999999999999.
    - **Issue with Rounding**: Direct rounding (e.g., `Math.round(scaled)`) is problematic because:
        - Numbers < 0.5 round to 0 (never occurs).
        - Numbers ≥ 5.5 round to 6 (rare).
        - Middle numbers (e.g., 1.5 to 2.5) are more likely, skewing probabilities.

4. **Generating a Die Throw (1–6)**:

   - To ensure uniform distribution, scale, shift, and floor the number:

     ```
     var bigDecimal = Math.random();
     var improvedNum = (bigDecimal * 6) + 1;
     var numberOfStars = Math.floor(improvedNum);
     ```

   - **How it works**:
     (a) `bigDecimal = Math.random()`: Produces 0 to 0.999....
     (b) `bigDecimal * 6`: Scales to 0 to 5.999....
     (c) `(bigDecimal * 6) + 1`: Shifts to 1.000... to 6.999....
     (d) `Math.floor(improvedNum)`: Rounds down to integers 1, 2, 3, 4, 5, or 6.
   - **Result**: Each number (1–6) has an equal probability (∼16.67%).

5. **Additional Notes**:

   - `Math.random()` is pseudo-random, sufficient for most applications but not cryptographically secure.
   - The variable name `numberOfStars` in the original code is misleading; it should reflect the context (e.g., `dieRoll`).
   - General formula for random integers in range `[min, max]`:

     ```
     Math.floor(Math.random() * (max - min + 1)) + min;
     ```

   - For a die: `Math.floor(Math.random() * 6) + 1`.

Observations

- **Errors in Original Document**:

  - The variable `numberOfStars` is contextually inappropriate for a die throw.
  - The mathematical explanation incorrectly suggests multiplying by 100 quadrillion as a general approach, which is unnecessary for this case.
  - Typo in the LaTeX equation: `0.999999999999999 * 10000000000000000` should have 17 zeros to match "hundred quadrillion."

- **Improvements**:

  - Use a more descriptive variable name like `dieRoll`.
  - Combine operations into a single line for clarity:

    ```
    var dieRoll = Math.floor(Math.random() * 6) + 1;
    ```

  - Clarify that `Math.random()` never returns exactly 1.0.

- **Edge Cases**:

  - The code assumes `Math.random()` behaves consistently across environments, which is generally true for modern JavaScript engines.
  - No input validation is needed since `Math.random()` always returns a valid number.