Calculator App: Architecture and Design

This document outlines the architecture and design decisions behind the development of a comprehensive calculator application. The application leverages React's state management capabilities and incorporates a range of features, including scientific functions, memory operations, and a confetti animation triggered by specific user interactions.

State Management

The calculator's functionality relies heavily on state management to track user input, display values, and manage various application states. React's useState hook is employed to define and manage the following state variables:

- displayValue: Stores the value displayed on the calculator screen.
- memory: Holds the value stored in memory for later use.
- confetti: Controls the visibility of the confetti animation.
- isRadians: Indicates whether the calculator is in radians or degrees mode.
- expression: Stores the mathematical expression entered by the user.
- switchbtn: Toggles the visibility of scientific function buttons.

Confetti Animation

The confetti animation is triggered when the user enters a mathematical expression containing both the digits 2 and 6. The useEffect hook is used to manage the animation's display duration:

- The useEffect hook is triggered whenever the confetti state changes.
- If confetti is true, a timeout is set to automatically hide the confetti after 3 seconds.

Digit and Operator Handling

The handleDigit function appends digits to the display and expression, while the handleOperator function appends operators to both. These functions ensure that user input is correctly reflected in both the display and the underlying mathematical expression.

- The handleDigit function checks if the displayValue is currently "0". If so, it replaces it with the new digit. Otherwise, it appends the digit to the existing displayValue.
- Both functions update the expression state to maintain a record of the user's input.

Calculation and Result Display

The calculateResult function evaluates the mathematical expression stored in the expression state. It utilizes the evaluate function (not shown in the provided code) to perform the actual calculation. The result is then displayed on the calculator screen.

- The calculateResult function first checks if the expression is empty. If so, it returns null.
- It then attempts to evaluate the expression using the evaluate function. If successful, it returns the result. Otherwise, it returns null, indicating an error.
- The handleEquals function triggers the confetti animation based on the expression, calculates the result, and updates the displayValue and expression states.

Memory Operations

The calculator provides a set of memory operations that allow users to store and manipulate values. These operations include:

- handleMemoryClear: Clears the memory.
- handleMemoryAdd: Adds the current display value to the memory.
- handleMemorySubtract: Subtracts the current display value from the memory.
- handleMemoryRecall: Recalls the memory value to the display.

Scientific Functions

The calculator offers a wide range of scientific functions, including trigonometric, logarithmic, and exponential operations. The handleScientificFunction function handles the execution of these functions based on the button clicked.

- The function first retrieves the current display value as a floating-point number.
- It then uses a switch statement to determine the appropriate mathematical operation based on the function argument.
- The result of the operation is then displayed on the calculator screen.

Radians and Degrees Toggle

The handleRadiansToggle function allows users to switch between radians and degrees mode. This toggle affects the behavior of trigonometric functions, ensuring that calculations are performed using the correct units.

- The function simply toggles the isRadians state variable, which is used to determine the appropriate angle unit for trigonometric functions.