

Elevator Challenge Architecture Overview

The project is a simulation of a building with multiple floors and elevators implemented using React. It consists of the following components:

1. **App Component:** The main component of the application responsible for setting up the basic parameters for the building and elevators. It renders the Building component and passes the required props such as the number of floors, number of elevators, floor height, wait time, and time for floor.
2. **Building Component:** This component represents the building structure and contains the logic for managing elevator movements and floor statuses. It initializes the state for elevator and floor statuses and handles elevator calls from floors. The Building component renders Floor and Elevator components based on the provided number of floors and elevators.
3. **Floor Component:** Each Floor component represents a floor in the building and includes a button for calling an elevator. It manages the remaining wait time for an elevator to arrive and updates the timer accordingly. The Floor component also triggers the handleElevatorCall function when the button is clicked.
4. **Elevator Component:** The Elevator component represents an elevator in the building. It displays an image of the elevator and applies CSS transitions to simulate elevator movements between floors. The position and transition effect of the elevator are determined based on its status, such as current destination and movement duration.

Main Algorithm Description

The main algorithm for managing elevator movements and handling elevator calls is implemented within the Building component. Here's an overview of the key steps:

1. **Handling Elevator Calls:** When a floor calls an elevator, the closest available elevator is selected based on the calculated wait time. The elevator's queue is updated with

the called floor, and the elevator's status is set to moving if it's not already moving or waiting.

2. **Finding Closest Elevator:** The algorithm calculates the wait time for each elevator to reach the called floor and selects the elevator with the shortest wait time as the closest elevator.
3. **Calculating Wait Time:** The wait time for an elevator to reach a specified floor is calculated based on the current position of the elevator, queued stops, and travel time between floors. The total wait time includes the time to reach each stop and the wait time between stops.
4. **Moving Elevator:** When an elevator is selected to serve a floor, it moves to the target floor with a simulated delay to mimic the travel time. Once the elevator reaches the floor, it stops, opens its doors, and waits for a specified duration before continuing to the next stop.

Overall, the algorithm efficiently manages elevator movements and ensures timely responses to floor calls, providing an effective simulation of elevator operations in a multi-floor building.