

Elevator Simulation in C++

Project Overview

This project simulates an elevator system in a multi-floor building using C++ and SFML. The simulation models the behavior of elevators, handling user requests, managing elevator movement, and providing a real-time graphical visualization with sound effects. The main goal is to practice object-oriented programming (OOP), system design, and interactive graphical user interfaces.

How to Compile and Run the Simulation

Prerequisites

- C++ compiler (g++, clang++, etc.)
- CMake (for building the project)
- SFML library (for graphics, audio, and window management)

Compilation Steps

1. Clone the repository or download the project files.
2. Navigate to the project directory:

```
cd elevator-simulation
```

3. Create a build directory:

```
mkdir build && cd build
```

4. Run CMake to configure the project:

```
cmake ..
```

5. Compile the project using:

```
make
```

6. Run the simulation:

```
./elevator_simulation [options]
```

Command-Line Options

- **-e, --elevators <num>**: Set the number of elevators (1-5, default: 4)
- **-f, --floors <num>**: Set the number of floors (2-10, default: 10)
- **-h, --help**: Display help message

Examples:

```
./elevator_simulation -e 3 -f 8 # Run with 3 elevators and 8 floors
./elevator_simulation --elevators 5 --floors 10 # Run with 5 elevators and 10 floors
```

Features

All features required in Project requirements have been successfully implemented, including bonus features of GUI, Sound and proximity algorithm.

Core Features

- Configurable number of floors and elevators via command-line arguments
- Elevator movement between floors based on user requests
- Different elevator states (idle, moving, doors open)
- Graphical visualization of building, floors, and elevators
- Sound effects for elevator actions (door opening/closing, arrival)
- Realistic time-based simulation for elevator behavior

User Interaction

- Request elevators to specific floors using keyboard number keys or by clicking floor buttons
- Direct control of specific elevators using function keys
- Click on elevators to set their destination floors
- When a floor button is pressed and an elevator is already at that floor, the doors will open
- Visual feedback for elevator states (blue for idle, yellow for moving, green for doors open)

User Interface

- Adjustable UI scale using '+' and '-' keys
- Dynamic window resizing with responsive UI elements
- Interactive elevator selection via mouse clicks
- Visual input modal for entering destination floors
- On-screen instructions and status information
- Clickable floor buttons for calling elevators

Controls

- **Keys or floor button**: Request an elevator to that floor
- **F8-F12 Keys**: Add internal requests to specific elevators (F8 for first elevator, F9 for second, etc.)
- **Mouse Click**:

- Click on a stationary elevator to set its destination floor
- Click on a floor button to call an elevator to that floor
- **+/- Keys:** Adjust the UI scale factor
- **Escape Key:** Exit the simulation

Architecture

The project uses a modular object-oriented design with the following key components:

- **Building:** Manages the collection of floors and elevators
- **Elevator:** Handles elevator state, movement, and request processing
- **Request:** Represents floor requests with direction
- **GUI:** Manages all graphical rendering and user interaction

TEAM

The project is developed by Shamsher Singh. Class: CIS278 Professor: Kamran Iftekhari

Demo Video

A short demo video showcasing the simulation in action is provided.