

# Elevator Simulation Project

## Problem Description

This program can be used to simulate the running and scheduling of a multi-cart elevator in the real world.

## Design Features

1. See <Threads> part.
2. Running of each cart is based on a stable finite state machine (FST).
3. SCAN disk scheduling algorithm is used to schedule carts.
4. Fastest response time strategy is used for system-level scheduling.
5. Similar to the elevator in reality, the input is divided into two parts: buttons inside the cart (each floor has one button in a cart) and buttons outside the cart (up and down for each floor).
6. Real time input. Output will be saved as a file, which could be used in our GUI tool. (See <Virtualization> part)

## Idealization Considerations

1. To simply our model, the acceleration and deceleration before cart starting and stopping are ignored. I considered implementing this feature in my original plan, but the time is not enough.

2. Capacity is not considered. This is because elevators in our real life generally do not consider capacity either. Even under full load, the cart will still stop at its destinations. It is hard to calculate number of passengers before enter the cart. This program will simulate the actual situation as much as possible.

3. Opening and closing doors will be considered to happen instantly. However, the state of door opening will be hold for a period of time.

4. Undo feature is not included in this version. Though it is no so hard to implement undo feature under certain conditions (see in my original plan).

5. All carts are running parallelly. In our real life, more complex strategies are always adopted. For example, 6 carts will be divided into 2 groups, each with 3 carts running parallelly. This will ensure both single-passenger efficiency and overall efficiency. (If we press buttons of both groups, there will be multiple choices.)

## Threads

Input Thread: For command input.

Output Thread: For console output and file output.

Up Going Control Thread: Schedule tasks go upwards.

Down Going control thread: Schedule tasks go downwards.

Each cart has a unique running thread for a Finite State Machine.

For input/output instructions, see <Readme.md> file.

## Virtualization (for Windows system)

I have implemented GUI tool for this simulator using Unity (a popular video game engine). If you want to try this GUI tool, please copy your "data.txt" output file generated from the simulator program into the "Elevator\_Data" folder inside the GUI program folder, then click "Elevator.exe" to start. A sample picture is shown in the next page.

I only tried this tool under 1080p (1920 \* 1080) resolution, a 16:9 resolution is recommended.

Later I will release a complete GUI version (with all clickable button input) based on Unity and C# multithreading on my Github (<https://github.com/wpn-zju/Elevator-Simulator>), as type letters in the console output window is annoying.

A Sample from the GUI tool

