Next Level Abstraction

TL;DR

Istvan Nagy

Introduction

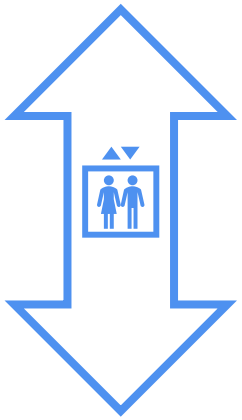
- Lead Software Engineer at EPAM
- 12+ years of Java experience
- Member of the EPAM Debrecen Java Community

He <u>thinks</u> he is funny

 **25%**

 **55%**

Abstract



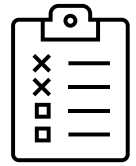
Detailed

Level of abstraction

> The amount of complexity by which a system is viewed or programmed. The higher the level, the less detail. The lower the level, the more detail.
> The highest level of abstraction is the entire system. The next level would be a handful of components, and so on, while the lowest level could be millions of objects.

PC Mag Encyclopedia

Abstraction for People



Simple

- Next destination

Self-conscious

- Next destination
- Position (floor index)
- Waiting since

Unique

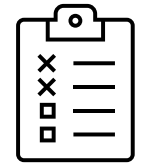
- Next destination
- Position (floor index)
- Waiting since
- Physical size/weight
- Preferences
- Behaviour

Special

- Next destination
- Position (floor index)
- Waiting since
- Physical size/weight
- Preferences
- Behaviour
- Can carry items (parcel/equipment)

Less abstract

Abstraction for elevators



Simplistic

- Position (floor index)
- Movement direction
- Floors covered
- Program

Brute force

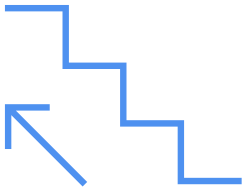
- Position (floor index)
- Movement direction
- Floors covered
- Program
- Speed

Democratic

- Position (floor index)
- Movement direction
- Floors covered
- Program
- Speed
- Buttons pressed
- Elevators called

Smart

- Position (floor index)
- Movement direction
- Floors covered
- Program
- Speed
- Buttons pressed
- Elevators called
- Load factor and people waiting
- Synch. program



Less abstract

The requirements (an example)

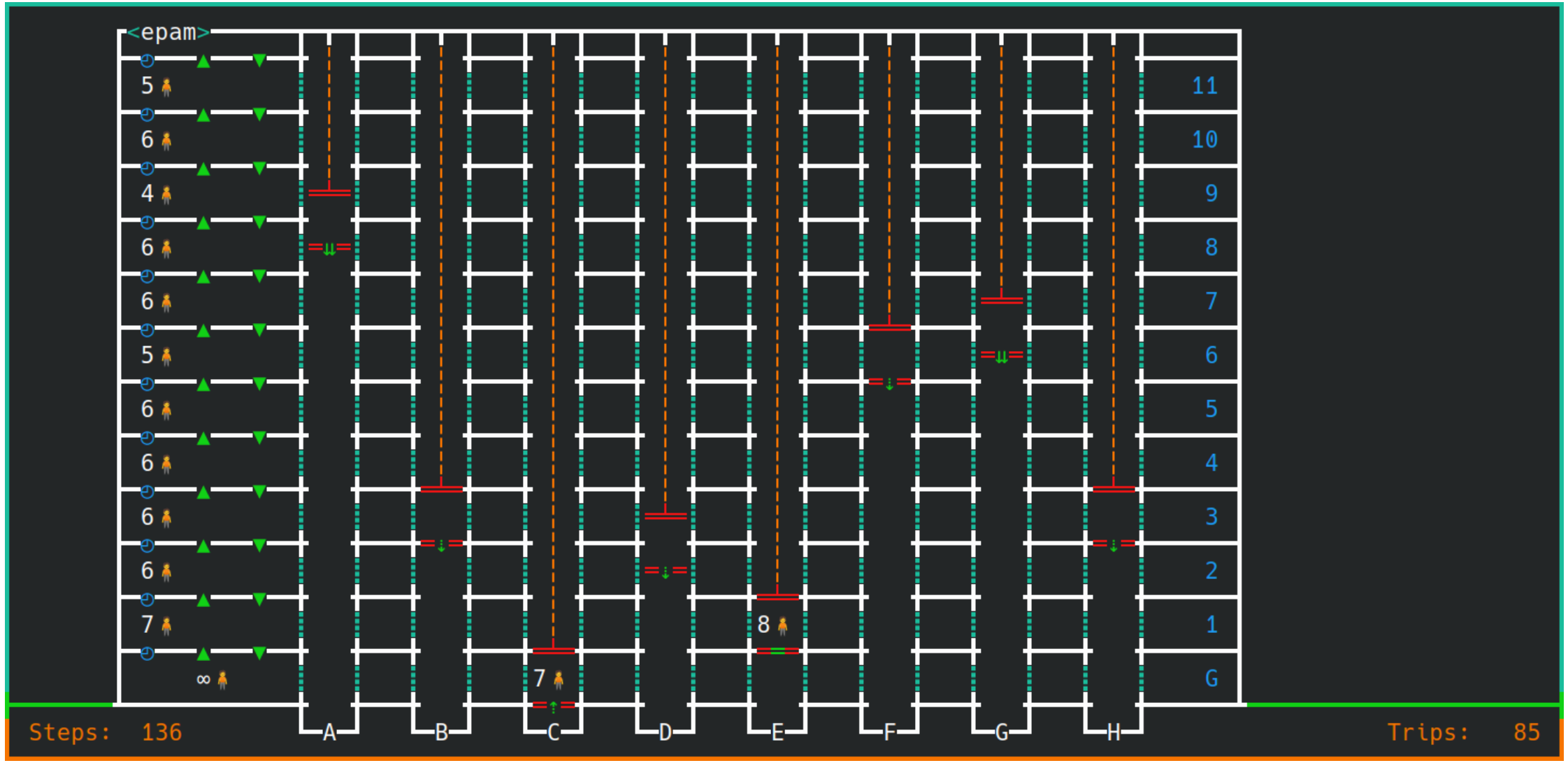
User story

As the elevator operator, I want the elevators installed in the Forest Offices to *minimize the average time* spent on *waiting* for the elevator *and travelling* in the elevator while people are using them, as *measured on a typical day*.

Acceptance criteria

Elevator controller implementation is changed to:

1. only stop elevators on a floor if:
    1. the button of that floor in the elevator car is pressed; or
    2. the caller button on the floor is pressed, and the elevator is:
        1. idle (no buttons pressed in the car); or
        2. moving in the direction of the floor where the button is pressed, and can stop safely on the floor
2. select the idle elevator that can arrive to a floor under the least amount of time when more than one elevators are idle
3. avoid sending more elevators to a floor than necessary in order to transport waiting persons



The “modern UI” I decided to use, appears to be older than I am, but it is from 2021.

About the project

Project home

- https://github.com/nagyesta/next-level-abstraction

Limitations

- Heavily uses UTF-8 characters (Windows terminal will show '?'-s)
- Needs your terminal to understand ANSI control sequences to move to the top left corner

Thank you!

Istvan Nagy

Website:

https://nagyesta.github.io