CodeBoxx

Odyssey

Week 1 - Algorithms and Problem Solving - Requirements Manual

Summary

This document is intended to document the requirements of Week 1 of the CodeBoxx "Odyssey", which is fourteen (14) weeks in duration.

At the end of each weekly theme, candidates must have been able to become familiar with and put into practice the concepts that make them operational on the various fields of expertise that follow one another.

Goal

1. Describe the scenario, context, tasks, requirements, deliverables and acceptability criteria to assess the participant's success.

Week 1: Algorithms and Problem Solving

Through Genesis, now-participating applicants have developed and deployed a static website and have done so through free market tools and have secured and exposed it in the cloud. They did so by exploring the control of Source Code. They were also able to familiarize themselves with some notions of the world of elevators. These concepts start to make sense in the context of week one.

I. Scenario

Rocket Elevators is in the process of designing a new generation of its products. It includes two solutions for lifts:

- A residential offer
- A commercial offer

The participants are in charge of writing the controller algorithms for each offer.

One of the keys to understanding this exercise is that the residential offer is simpler and should serve as a foundation for the controller of the commercial offer that will be deployed in corporate, commercial or hybrid buildings . The simple notions identified in the residential controller can be repeated and evolve in complexity.

II. Milestones, Requirements and Success Criteria

The requested algorithms must respect three criteria in order to understand the intrinsic role and definition of the algorithms:

An algorithm is a sequence of steps that solve a given task. Its abstract nature makes it independent of how it is likely to be implemented. It must meet three criteria to be considered valid:

1. It must be finite and delimited: If it never ends in trying to solve the problem in question, the algorithm is useless.
2. Its instructions must be clearly defined and each step must be precise. It must cover all cases and must be unambiguous.
3. It must be effective in solving the problem and it must be demonstrated in a simple way (on a board or on paper)

Requirements for the residential controller:

- The Residential building for which the first controller is to be programmed has only one column consisting of two elevators and 10 floors.
- Objects that need to be controlled:
    - Column
    - Elevators (Cages)
    - Call buttons
    - Doors
    - Floor request buttons
- The controller for residential building will serve intuitively for that of the corporate building that must also be programmed.

Requirements for the commercial controller:

- The Corporate building for which the second controller is to be programmed has 66 floors (**including** 6 basements) served by four columns, each consisting of 3 elevators.
- Objects that need to be controlled
    - Battery
    - Columns
    - Elevators (Cage)
    - Call buttons
    - Doors
    - Floor request buttons
    - Floor display

Additional requirements:

If candidates are familiar with the minimum requirements of this scenario, they can enhance the value of their deliverables by including the following considerations that will allow Rocket Elevators to offer superior quality products over the competition:

- Logic of prioritization of elevators
    - When calling on a moving cage, which elevator is to be sent?
    - Depending on the direction
    - Based on multiple floor calls and queries
- Logic back to the origin
    - Is it better to have a lift waiting on the ground floor?
    - Security logic
    - Routines checking system status before using them
    - Safe behavior of equipment (deadlines, decisions)
- Temporal logic
    - Provide the system with an awareness of the schedules to assign elevators to the floors most likely to be used at specific times of the day, week ...
    - Measure waiting times for calls to ensure that each user reaches their destination in a timely manner
- Logic of loading
    - Detect a full elevator
    - Correlated with the waiting time, how to ensure a service on the floors when the cages leave the origin systematically full?
- Other considerations can be suggested by the participant and validated by the client.

Deliverables:

- A private repository in the participant's Github account, named "Rocket_Elevators_Controllers" with its description containing the text "Contains the algorithm files for the elevator controllers for the New Rocket Elevator Solutions for both Residential and Commercial Offers"
- A text file named Residential_Controller.algo
- A text file named Commercial_Controller.algo
- Algorithms must be written in English
- Explaining video of 2 to 5 minutes:
    - As a key part of your portfolio, you must deliver a demonstration video of up to 5 minutes explaining the logic of your algorithms. This video demonstrates how well your algorithms work, explaining each step of the sequence when a user wants to go to a floor.