# Individual Project Report

## Analysis of the project design and implementation

By: Efstathios Sidiropoulos

Teacher: Evan Cavallo

Course: Programming Techniques – DA2005

Date 20.03.2021

# Table of Contents

# 1. Abstract

For this project, I have decided to go with the Trains topic. Based on the fact that it is focused on data structures, a topic that I would like to improve my knowledge of.

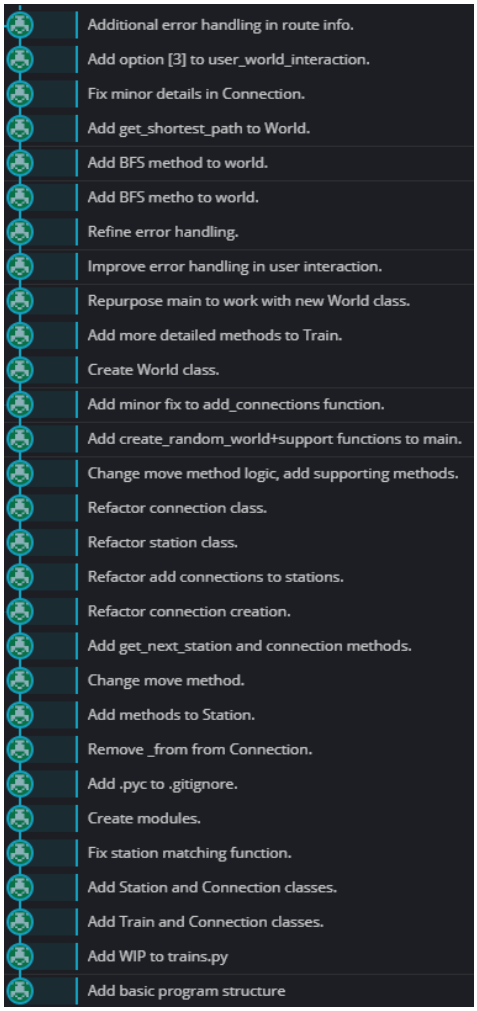
The program gives the user the ability to choose a number of trains to simulate on a fictional railway, that is created based on information provided in input files. The user is then given options on how to continue with the simulation.

# 2. Background

At the beginning of the project, I had no earlier experience of work with data structures and limited knowledge of Object-Oriented Programming (OOP), so a substantial portion of time was allocated to investigating and deciding the way forward.

The project has gone through many different iterations before it reached its final form (can be seen in picture 1a).

The first idea was to implement the simulation based on dictionaries with line identities as keys and lists of stations as values and have the trains traverse the lists back and forth. Ultimately, the idea was scrapped, in favor of a more object-oriented approach, due to the complexity of the requirements.




*1a. GitHub depository commits.*

# 3. Details

The program is structured based on 4 distinct classes – Train, Station, Connection and World. Train and Station are the main classes working the simulation. The network is structured in a graph, where the vertices are represented by Station objects and the adjacency lists are stored within Connection objects.

The different classes are stored in different .py files – modules for clarity. Connection is a support class storing information and being used as an argument for the Station class, using the add_connection method. The World class is the final class we use to store information on our train and station objects, which can also be considered our graph, but it also includes Train objects.

In the first iterations of the program, where fewer classes were used, a lot of supporting functions were used (as it can be seen in the GitHub commits – picture 1a) that were later scrapped in favor of class methods, in order to avoid bugs and make it easier to trace potential errors.

For task 2, a combination of breadth-first search and Guido van Rossum's "find shortest path" is used. Eryk Kopczyński's improved version of the algorithm was not used in order to avoid importing the collections library.

# 4. Functionality

The program is started using the main() function, where the user is being asked to specify the input files and choose the number of trains to simulate. Next, the user is asked how they want to interact with the simulation, given 3 different options, between [1] Moving the simulation one time unit ahead ($t + 1$), [2] asking the user which of the simulated trains they would like to receive information on and showing the information on the selected train for time unit $t$ and [q] to exit the simulation.

At a later stage, a 4$^{th}$ choice ([3]) was added to the program. Selecting that option, the user is able to choose between 2 stations (α, β) and time steps (t) and receive information on whether station β is reachable from station α with time t.

The user can repeatedly make selections until they decide to exit the program (option [q]). All encountered exceptions have been handled and the program does not crash due to invalid inputs.

The modules Train.py, Station.py Connection.py and World.py need to be downloaded alongside the main.py file for the program to run.

# 5. Closing Remarks

The project was a particularly challenging and time-intensive task, especially while also working a full-time job. However, I do feel like I have learned a lot from working on it and it was overall a very fulfilling experience, as I feel I have achieved my goal of getting some hands-on experience on OOP.

There are several improvements that can be made to the code, especially in structure and readability. These are things that I am planning to do in my own time after the completion of the course.