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

# 2.Background

# 3. Details

# 4. Functionality

# 1. Abstract

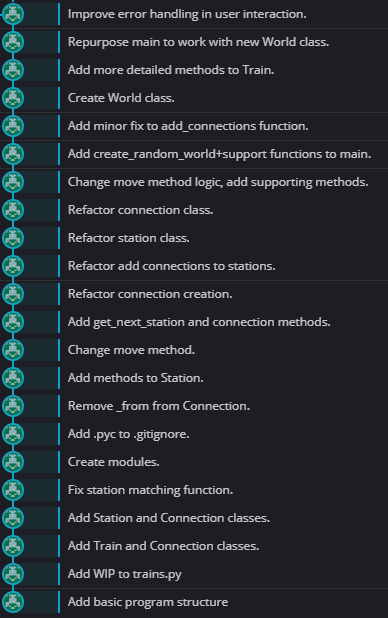
For this project, I have decided to go with the Trains topic. Based on the fact that, it is focused on data structures, a topic that I would like to improve my knowledge on.

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 on 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 1). 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*](https://github.com/Stat-sid/Project---Trains) *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 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.

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.

# 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 [3] to exit the simulation.