Integrative Project Assignment

pi-sem2-2024-25 - version 1.0

March 10, 2025

Abstract

This document comprises a detailed description of the Integrative Project Assignment for the practical work to be developed within ESOFT, MATCP, MDISC, PPROG, and LAPR2 courses. The work consists in developing an IT solution that allows a simplified simulation of the operation of railway networks. This document briefly introduces the business domain and sub-areas, the functional and non-functional requirements of the solution to be developed, the Integrative Project technical details, and the operating mode, i.e., the teamwork framework.

Table 1: Version register

Version	Description
1.0	First version

Contents

1 Integrative Project

In this project, students should analyze, design, and implement a computer solution that allows a simplified simulation of the operation of railroad networks.

Students must be organized into teams and develop a proof of concept that includes two components: *i)* (i) a map editing tool (including industries and cities) and scenarios (including temporal and historical context), and *ii)* (ii) a simulation tool that will allow the user to define and interact with the operation of a railway network (including lines, stations, and trains) in the context of a map/scenario. To manage the railway network, two types of tools will be considered: statistical tools to assess the network performance and tools to analyse the network connectivity and the quality of the services provided.

Following the good practices learned throughout the degree and in particular in Software Engineering (ESOFT), Programming Paradigms (PPROG), Computational Mathematics (MATCP), Discrete Mathematics (MDISC), and Laboratory-Project II (LAPR2) courses, this project implies applying an iterative and incremental development process. Therefore, the agile SCRUM approach must be used to manage teamwork in each four-week SPRINT.

The software solution to be developed must be composed of a set of Java applications that must meet the requirements. To increase the solution maintainability and respect good software development practices, the implementation must follow a Test-Driven Development (TDD) approach.

1.1 Team operation framework

Each team with LAPR2 students must nominate a Scrum Master for each Sprint (rotative role) and create a Scrum Board in the Organization *Departamento de Engenharia Informática*, following these guidelines:

- Only one team member (the assigned scrum master) creates the Project Board with the title ”sem2-pi-24.25-gnnn-board in which nnn must be replaced by the team number;
- The board privacy must be ”private”;
- Access (writer) must be given to TP and PL teachers assigned to the team's class;
- The project board must include the following columns: Backlog, ToDo, In Process, Testing, Done, Logbook.

All teams involved in the integrative project must create the project repository on GitHub in the Organization *Departamento de Engenharia Informática*, following these guidelines:

- Only one team member creates the repository, with the title sem2-pi-24.25-gnnn-repo based on the provided **template**;
- All TP, PL and OT teachers from the involved class courses supervising the project must be added to the repository with the "Reader" role.

2 Problem Statement: Description

The railway is an efficient, safe, and environmentally friendly way of moving people and goods. Given that the proper installation and maintenance of a railway structure involves very significant investments on a national scale (and transnationally in the European context), it is obvious that there is a need to manage the installed resources effectively. In addition to the structural and economic aspects of railways, trains and railways are a recurring theme in computer games, allowing experimentation from a playful and entertainment perspective, involving some of the complexity inherent to the subject.

In the context of this project, the teams will have to develop a software solution that supports some of the basic functionalities of managing a railway system, inspired by a classic simulation game (RailRoad Tycoon). Two tools will have to be developed: a map and scenario editor and a simulation tool. In the context of the simulator, functionalities will be created that allow statistical analysis of the functioning of the railway network, including its multiple components, as well as tools that explore the topological aspects of the network.

2.1 Edition of Maps and Scenarios

The Map Editor allows you to create rectangular maps and place static elements on the map. Elements can be cities or industries.

2.1.1 Cities

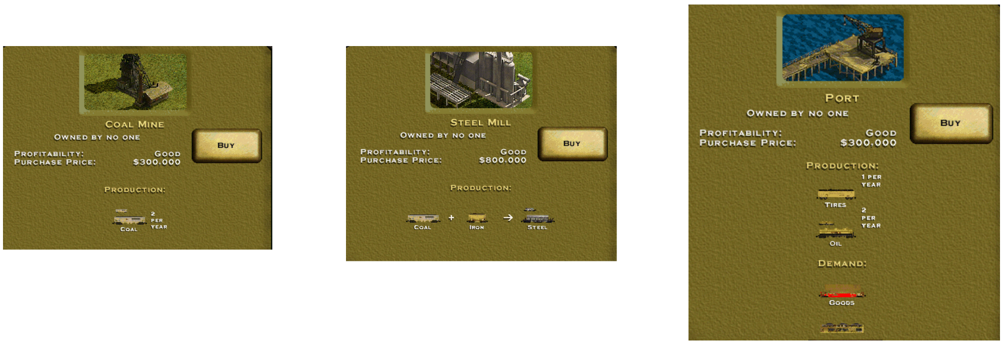
A city bears a name, a location (with XY coordinates on the map), and a set of blocks that represent housing. In the simulator's context, cities generate and consume passengers and mail, but they also consume final products (e.g., food, textiles, cars). The image in figure 1 refers to the city of Madrid.

Figure 1: City of Madrid

2.1.2 Industries

As for industries[1], there are three different types:

- **Primary sector industries** - which generate resources, such as mines (*e.g.,* iron, coal, bauxite, or nitrate mines) and farms (vegetables, cereals, wool, coffee, rubber, cattle). The resources generated by these units can, if transported by train in the context of the simulator, be consumed/processed by other types of industries;



Coal Mine
Owned by no one
Profitability: Good
Purchase Price: $300,000
Production:
Coal 2 per year

Steel Mill
Owned by no one
Profitability: Good
Purchase Price: $800,000
Production:
Coal + Iron → Steel

Port
Owned by no one
Profitability: Good
Purchase Price: $300,000
Production:
Tires 1 per year
Oil 2 per year
Demand:
Goods

Figure 2: Coal Mine, Steel Mill and Port

- **Transforming industries** - which transform the resources provided by the Primary Sector Industries into new products. This industry includes bakeries, textiles, automobiles, and steel mills. For example, a steel mill transforms coal and iron ore into steel. The steel produced can then be transported to an automobile factory and used in the production of automobiles.

[1] For a full list of industries available in the game, check railroad-tycoon.fandom

- **Mixed industries** - Ports potentially have the capacity, depending on what is defined in the scenario, to import/export any cargo.

In the images in figure 2, it is possible to see, next to the city of Madrid, the existence of a steel mill, a nitrate plant, and a wool production farm.

The Scenario Editor must allow the user to define the restrictions and conditions for using a given map, namely:

- Time Restrictions - the period in which the simulation will take place;
- Technological Restrictions - for example, which locomotives or industries are available in the scenario;
- Historical Restrictions - historical events that make sense in the scenario and change the operating conditions of the simulator. For instance, a war increases the steel demand, and a vaccination campaign or a schooling program encourages mail and/or passenger traffic. In this context, the goods that the Ports import and export are also defined.

2.2 Simulator

The simulator will enable the manager (or in the playful perspective, the player) to create a railway network composed of stations that serve cities and industries, lines connecting those stations, and trains that transport cargo between stations, according to a route defined by the manager. It is crucial to highlight that cities and industries only generate or transform cargo if they have a station; when the cargo is generated, it is then available to be collected at the station.

2.2.1 Stations

There are three types of stations as represented in figure 3: *Depots* (50k, 3x3 radius), *Stations* (100k, 4x4 radius), and *Terminals* (200k, 5x5 radius). Their cost and economic radius distinguish the stations.

All industrial and population sites within the radius of a station send (supply) and receive (demand) business through the station. By adding the supply and demand for cargoes from the industry and population in a range, the supply and demand for the station are determined. For example, if it is assumed that each coal mine creates an average supply of two coal carloads early, a station with three coal mines in its economic radius generates a supply of about six carloads of coal per year.

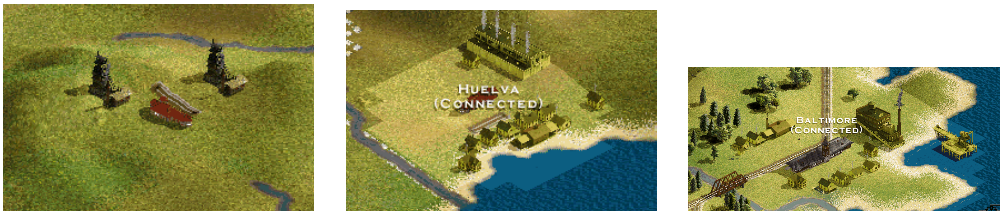


Figure 3: Depot, Station and Terminal

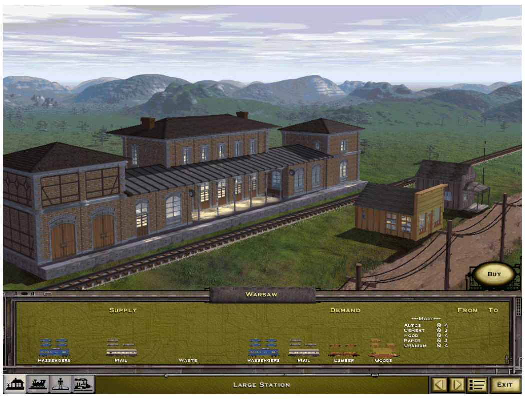


Figure 4: Warsaw Terminal

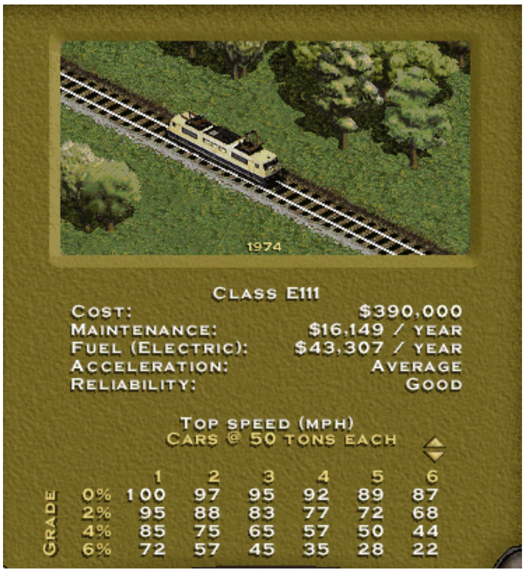
When built, a station consists of just one building, which can be upgraded with other buildings that can increase the station's performance, improve train operation, facilitate train manoeuvrers, and increase the lifespan of cargo stored at the station. Some of the buildings that can be used to upgrade stations are: telegraph (later telephone), café (small or large), customs, post office, hotel (small or large), silo, liquid storage. When viewing a station, in addition to the buildings present, the cargo that is ready for collection and the cargo that needs to be delivered (as in the example) should also be displayed (Figure 4).

2.2.2 Railway lines

Railway Lines allow the connection of two stations. In the context of this project, it is not intended to make a detailed graphical edition of the route, but only a topological representation of the connection between two stations. A line has a certain length, can be single or double, and can be electrified (or not).

2.2.3 Trains, Locomotives and Carriages

Trains are composed of locomotives and carriages. The locomotives provide traction to the composition (train) and can be of different types, namely, run by steam, diesel, and electricity. Note that electric locomotives require electrified lines. The locomotives are characterized by several technical aspects, such as power, acceleration, top speed, start year of operation, and acquisition price. The carriages[2] are distinguished by the type of cargo they can transport (*e.g.,* passengers, mail, coal, iron ore, steel, cars) and, in the context of the simulator, they have no acquisition cost.



Class E111

Cost:	$390,000
Maintenance:	$16,149 / year
Fuel (Electric):	$43,307 / year
Acceleration:	Average
Reliability:	Good

Top speed (mph)
Cars @ 50 tons each

Grade	1	2	3	4	5	6
0%	100	97	95	92	89	87
2%	95	88	83	77	72	68
4%	85	75	65	57	50	44
6%	72	57	45	35	28	22

Figure 5: E111 Electric Train

2.2.4 Routes and Cargos

In the simulator, the player/user acquires trains within the available budget and can put the train into service on a specific route. A route is a list of stations where the train passes, along which it loads (carriages) cargo at each station.

2.2.5 Simulator Operation

In addition to creating the dynamic elements already described, the player/user can play or pause the simulator. During the simulator's operation, the evo-

[2] For a full list of carriages available in the game check railroad-tycoon.fandom

lution of events should be displayed, namely the trains' journeys (start and end).

2.2.6 Project Exclusions

The RailRoad Tycoon (II and later) game/simulator is particularly complete, covering the areas of gameplay, economy and engineering. Keeping in mind that this game is an inspiration for the current project, there are many topics covered by the game that will not be focused on in the project, in particular:

- Collision detection
- Timetable generation/management
- Calculation of train (de)accelerations
- Editing of line details (graphical mode), connections will be topological

2.3 System Users

This system may be used by different users, namely:

- **Editor** - a person who can create maps and scenarios that can be provided in a bundle with the game or sold separately.
- **Player** – a person who wants to play a game; in advanced versions, a scenario can be played by multiple players simultaneously.

3 Minimal Viable Product (MVP)

The purpose of this project is to develop a Minimal Viable Product (MVP) iteratively and incrementally; therefore, the work is divided into three Sprints:

- **Sprint 1** – Weeks 3 to 6 – from 10/March to 6/April
- **Sprint 2** – Weeks 7 to 10 – from 7/April to 18/May
- **Sprint 3** – Weeks 11 to 14 – from 19/May to 15/June

A description of the MVP is provided for each sprint. Teams must follow the user stories (US) provided and consider their interconnections and respective dependencies. At the end of each Sprint, each team must be able to meet the specified requirements. Teams must add USs to the backlog, size them appropriately, and distribute them among team members[3].

[3] These will also be supervised in TP LAPR2 classes

3.1 Sprint 1

3.1.1 Maps, Scenarios and Simulator

This Sprint aims at developing the following User Stories (US) (Requirement Engineering and Analysis):

- US01 - As an Editor, I want to create a map with a size and a name.
- US02 - As an Editor, I want to add an industry in a position XY of the selected map.
- US03 - As an Editor, I want to add a city in a position XY of the selected map, with a name and a positive number of house blocks. AC1: A city name cannot have special characters or digits. AC2: The house blocks can be assigned manually or automatically (randomly around the city tag position)
- US04 - As an Editor, I want to create a scenario for a selected map.
- US05 - As a Player, I want to build a station. AC1: Overbuilding is not possible
- US06 - As a Player, I want to upgrade a selected station with a building
- US07 - As a Player, I want to list all the stations to select one to see its details, including the existing building and the demand/supply cargoes.
- US08 - As a Player, I want to build a railway line between two stations. AC1: The player should choose the stations from a list of registered stations.
- US09 - As a Player, I want to buy a train. AC1: The player should choose the locomotive from a list of available locomotives for the scenario as well as a current date.
- US10 - As a Player, I want to assign a selected train to a route with a valid station and the respective list of cargoes to be picked up
- US11 - As a Player, I want to list all trains. AC1: The display of trains must include the locomotive and the current cargoes. AC2: The trains should be grouped by locomotive type and then listed alphabetically by name.

- US12 - As a Product Owner, I want to create a simulator that generates cargoes at current stations, automatically, considering the cities and industries that the railway network serves. AC1: This simulator should provide options for start/pause.

3.2 Non-functional requirements

This section describes some non-functional requirements that must be considered when implementing the project.

- Validation of business rules must be respected when recording and updating data.
- The class structure must be designed to allow easy maintenance and the addition of new features, following the best Object-Oriented (OO) practices.
- The application must be developed in Java language.
- All those who wish to use the application must be authenticated with a password of seven alphanumeric characters, including three capital letters and two digits.
- The application documentation must be in English.
- During system development, the team must: (i) adopt best practices for identifying requirements and for OO software analysis and design; (ii) adopt recognized coding standards (e.g., CamelCase); (iii) use Javadoc to generate useful documentation for Java code.
- The development team must implement unit tests for all methods, except for the methods that implement Input/Output operations. Unit tests should be implemented using the JUnit 5 framework. The JaCoCo plugin should be used to generate the coverage report.
- All the images/figures produced during the software development process should be recorded in SVG format.
- The application ought to employ object serialization to guarantee the persistence of the data in two successive runs.