Mauricio de Souza Hilpert, Elias Heß, Yiyuan Ma, Salaheddine Mabchour, Laura Gradwohl

Real-Time Traffic Simulation with Java

Milestone 2:

Milestone Goal

The goal for the second milestone is a Live SUMO connection and the ability to add vehicles and control the traffic lights with visualisation in our own GUI and easy-to-understand code documentation, as well as having an User Guide.

The **complete overview to our Milestone** progress for the second milestone can be found under the following link: Overview_Milestone_2

The Link guides to a Markdown file that contains the technical details, the written user guide with screenshots for a better visualisation as well as the roles and our next steps.

Key Features

Real-time Visualization: View the traffic map and vehicles moving in real-time.
Traffic Light Control: Select any junction and manually switch phases (Green/Yellow/Red) safely using synchronized logic.
Vehicle Injection: Add specific car types (Ferrari, Bugatti, F1, etc.) on demand.
Stress Testing: One-click scenario to inject 100+ vehicles to test network capacity.
Robust Logging: Uses Log4j 2 for detailed simulation tracking and error handling. Crash **Prevention:** Implements smart phase switching (Modulo arithmetic) to prevent SUMO connection errors.

Technologies Used

Programming Language: Java 21 (Compatible with Java 8+)
GUI Framework: Java Swing (JFrame, JPanel)
Simulation Engine: Eclipse SUMO (https://eclipse.dev/sumo/) (Simulation of Urban MObility)
Middleware: TraCI4J / TraaS (Traffic Control Interface for Java)
Logging: Apache Log4j 2 (for robust error tracking and event logging)
Architecture: MVC (Model-View-Controller) Design Pattern

Mauricio de Souza Hilpert, Elias Heß, Yiyuan Ma, Salaheddine Mabchour, Laura Gradwohl

User Guide

For our user guide we have a short video tutorial in our project repository. It is available under the following link: User_Guide_Videotutorial

We also have a detailed written description:

1. Introduction:

Welcome to our Traffic Simulation tool. This tool allows you to simulate and analyse traffic flows in real time. You can control the vehicles and the signals.

2. Interface overview

The control panel contains buttons located at the bottom of the screen.
The following elements:

- Start
- Step
- Vehicle Type
- Add Car
- Stop
- Stress Test
- Signal
- Switch

3. Basic operation

- Start:

To start the SUMO programme, press the start button.

- Step:

Advances the simulation by one step.

- Vehicle Type:

A dropdown menu to select the vehicle type to add. Options are: Standard, Ferrari, Formula 1, Bugatti and Red Car.

- Add Car:

Inserts a new vehicle of the selected type (default: standard) in a random position on the map

- Stop:

Stops the SUMO program.

- Stress Test:

Stress tests the program with the maximum number of cars on the map with automatic steps.

- Signal:

A dropdown menu to select intercetion for the traffic light change.

- Switch:

Allows you to manually switch the state of the traffic light.

Progress Summary

Project Organisation and Challenges:

As we realised that we did not keep each other up-to-date enough we had more short meetings for the second milestone while finding it still hard to coordinate five group members with different time- and work-schedules especially since the time for the second milestone is right during the time of the exam retakes.

We tried to distribute the tasks as evenly as possible, so that everybody had the opportunity to get involved.
Salaheddine did a lot of the coding framework, gave a lot of technical input and kept track of our shared git repository. Yiyuan was responsible for the code documentation and the traffic light control feature.
Mauricios task was it to help code the stress test and the GUI, the user guide draft was Elias duty and I wrote this progress summary, tested the code and helped with the team coordination as well as the traffic light feature and the implementation of our GUI.
As mentioned, it was a challenge to coordinate our whole team. To tackle this problem we started communicating more on different platforms (WhatsApp and Discord) as well as scheduled more meetings even when not everybody could join. We keep everybody in the loop with documenting what we decided during our meetings so that everybody could read what we settled on regarding task distribution and necessary next steps.

Next steps:

For our next steps we started having a weekly meeting on Mondays, so that is going to happen before our presentation in the exercise on Tuesday.
Regarding the third and final Milestone our next steps consist of further streamlining our ways of communicating and keeping up our meeting cycle during the days before the Christmas break, where we decided to have week of before starting to work on the final steps again.
Before the break we want to revisit the tasks everybody has done for the project so far and decide on the next steps for each of us.

Declaration of Authorship

I hereby declare that the submitted project is my own unaided work or the unaided work of our team. All direct or indirect sources used are acknowledged as references. I am aware that the project in digital form can be examined for the use of unauthorized aid and in order to determine whether the project as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future projects submitted. Further rights of reproduction and usage, however, are not granted here. This work was not previously presented to another examination board and has not been published.

Mauricio de Souza Hilpert

Elias Heß

Laura Gradwohl

Yiyuan Ma

Salaheddine Mabchour

Frankfurt am Main, 14.12.2025