User Guide Draft

Real-Time Traffic Simulation with Java

1. Introduction:

This is a prototype of a real-time traffic simulation for our final simulation.
It is developed in the programming language Java.
Our prototype connects to SUMO via the TraaS API and makes it possible for us to manage and interact with our traffic simulation.

The embedded system provides a GUI to control vehicles, traffic lights and to have an overview of the real-time traffic simulation.

Therefore our current system is just a small simulation with the needed requirements for Milestone 2.

2. System Requirements:

To run our simulation, the following applications are needed:

- JDK 17 or higher
- SUMO
- Traas, TraCI and LibTraCI
- A working .sumocfg file
- A developing environment (we are using Eclipse IDE)

3. How to start the simulation:

1. Open the current project in the development environment
2. Make sure to add all necessary external libraries
3. SUMO needs to be installed on the system
4. In the Main.java file edit the “sumoGuiExe” and “sumoCfg” Strings file path as needed and saved in your file manager
5. In GUI.java change the netPath String file path as needed and saved in your file manager
6. Start the simulation by running the main class of the project#

Through those steps the system connects to SUMO and begins the simulation in real time.

```java
public static void main(String[] args) throws Exception {

    //File path to the Sumo-Gui executable and Sumo configuration
    String sumoGuiExe = "\"C:\\Program Files (x86)\\Eclipse\\Sumo\\bin\\sumo-gui.exe\"";
    String sumoCfg = "\"C:\\Users\\yelme\\eclipse-workspace\\Milestone2\\sumo\\Demo2.sumocfg\"";
```

```java
private void initLayout() {
    // ---- Map laden (Lanes) ----
    // TODO: später konfigurierbar machen (FileChooser / aus sumocfg net-file lesen)
    String netPath = "C:\\Users\\yelme\\eclipse-workspace\\Milestone2\\sumo\\Demo2.net.xml";
    List<LaneShape> lanes = MapDataLoader.loadLanes(netPath);
    mapPanel = new MapPanel(lanes);
```

4. User Interface Overview:

The GUI contains the following components:

4.1 Map:

The map displays:

- The street network of the simulation
- Colored vehicles with unique IDs
- Traffic lights with current state

4.2 Control Panel:

The control panel makes it possible to interact with the simulation.
Typical controls include:

- Integration buttons for vehicles into the simulation
- To control traffic lights
- To start/stop the simulation

5. Core Functions:

5.1 Vehicle Implementation:

The User can implement vehicles into the simulation through the Gui:

Steps:

1. Select the start point and road
2. Configure the speed and color of the vehicle
3. Apply the settings and get started

5.2 Traffic Light Control:

Traffic lights are controlled via the GUI

Possible actions:

- State of traffic lights
- Switching the states of traffic lights
- Seeing the network changes on road via traffic lights

5.3 Live Simulation:

After configuring and starting the simulation, the simulation runs by itself while changing traffic lights states and vehicles positioning.

6. Final Word:

Our User Guide Draft shows the user the necessary steps to set up and start the project. Through the steps the user gets to know how to naviagte the fundamentals of the simulation like traffic light controlling, vehicle implementation and what applications are needed to make the simulation work.