Java-SUMO Real-Time Traffic Simulation Project User Guide

---

1. Project Overview

1.1 Project Introduction

This project is a real-time traffic simulation system developed in Java with SUMO as the simulation core. It realizes bidirectional communication between Java and SUMO through the TraCI interface, supporting functions such as road network loading, vehicle injection, and traffic light control. It is suitable for scenarios like traffic flow analysis and signal control verification.

1.2 Core Features

- Visual road network display and loading
- Batch/custom vehicle injection
- Automatic/manual traffic light control
- Real-time simulation monitoring

1.3 Technical Architecture

Layer	Technology Selection	Core Functionality
Simulation Core	SUMO 1.18.0+	Provides underlying traffic simulation support
Control Interface	TraCI Java API	Enables real-time communication between Java and SUMO
Application Development	Java Swing (GUI)	Offers visual operation interface
Data File Layer	.net.xml/.rou.xml/.sumocfg	Stores basic simulation data

2. Environment Preparation

2.1 Install SUMO

- Download Link: SUMO Official Download Page
- Installation Steps:

  Windows: Run the installer, recommend default path *C:\Program Files\SUMO*, check *"Add SUMO to PATH"*

  Linux: Execute command *sudo apt-get install sumo sumo-tools sumo-doc*
- Verification: Enter *sumo --version* in the command line, successful if version information is displayed

2.2 Configure Java Environment

- Install *JDK 11+*
- Configure environment variable *JAVA_HOME*
- Verification: Enter *java -version* in the command line, successful if version information is displayed

2.3 Import Project Dependencies

- Open *IDEA* and import project source code
- Add *TraCI Java* dependency: *SUMO_HOME\tools\traci4j\lib\traci4j.jar*

3. Project Deployment and Startup

3.1 Core Project Files

map.net.xml: Road network file

map.rou.xml: Route file

sumo_config.sumocfg: SUMO simulation configuration file

3.2 Quick Configuration File Check

Open *sumo_config.sumocfg* to ensure correct paths

3.3 Three-Step Startup

- Open *IDEA*, run the "*main"* method of "*SumoGUI.java"* to launch the visual operation interface
- Click the *"Start SUMO Simulation"* button on the interface to automatically start *SUMO-GUI* and load the road network
- Once the *SUMO-GUI* interface pops up, start operations

4. Key Function Description

4.1 Vehicle Injection

Batch Injection: Click *"Inject Vehicles"*

Custom Adjustment: To modify vehicle quantity or departure frequency

4.2 Traffic Light Control

Automatic Mode: Default cycle switching

Manual Mode: *Toggle Traffic Light Phase* to manually cycle through states

4.3 Simulation Monitoring

- Real-Time View: *SUMO-GUI* interface intuitively displays vehicle position, speed, and traffic light status
- Speed Adjustment: Drag the *"Speed"* slider at the left side of *SUMO-GUI* to adjust simulation speed
- Pause/Resume: Click *"Pause"* to stop and *"Play"* to resume in *SUMO-GUI*

5. Common Issues and Solutions

5.1 SUMO Startup Failure

- Problem: No response when clicking *"Start SUMO Simulation"* or *"Failed to connect to TraCI"* error
- Solutions:

a. Check if file paths in *sumo_config.sumocfg* are correct

b. Verify SUMO environment variable configuration (re-run *sumo --version* for confirmation)

c. Close programs occupying port or restart the computer

5.2 Vehicles Not Displayed/Abnormal Disappearance

- Problem: Vehicles not shown in SUMO after injection or sudden disappearance while driving
- Solutions:

a. Ensure *time-to-teleport* in *sumo_config.sumocfg* is set to -1

b. Check if the road network file matches the route file

c. Reduce initial vehicle speed

5.3 Traffic Lights Not Working

- Problem: Traffic lights not displayed in SUMO or vehicles ignoring traffic lights
- Solutions:
  a. Confirm *"View→Show Traffic Lights"* is checked in SUMO-GUI
  b. Verify if junction ID matches traffic light configuration
  c. Restart the simulation and try again

5.4 Simulation Lag/Crash

- Problem: Simulation lags or crashes suddenly during operation
- Solutions:
  a. Close other memory-intensive programs
  b. Reduce the number of injected vehicles
  c. Lower simulation speed