

# Fourth Quarter Final Research Paper: JLSim - Visual Microscopic Traffic Simulation Application with Extensive User Interface

Jinyu Liu

June 9, 2008

## Abstract

At the beginning, deciding upon a project idea was excruciatingly hard. I eventually decided I wanted to make some sort of simulation because that would be probably the most fascinating type of project to code and to test. I eventually decided upon creating a traffic simulation even though traffic simulations have been done before after suffering through all kinds of bad traffic, sometimes caused by pure congestion. I decided then that I wanted to make a somewhat simple but still effective simulation of traffic flow based on some basic user-defined variables.

After seeing many traffic simulation web applets that work but have limited interaction on the part of the user, I wanted to create a traffic simulation that had an extensive user interface through which the user had high freedom and customizability options. A project in this field is called VISSIM, created by Tom Fotherby. This is an extensive project written in 11,000 lines of Java code to produce traffic simulation using a microscopic approach. The finished application should be able to at least somewhat realistically simulate real world behavior of traffic. Therefore, the primary goal of this project is to provide an accurate simulation of real world traffic behavior for multiple settings based on user-defined data. VISSIM also included a pretty powerful design mode, where the users could place roads onto a grid and essentially create scenarios by themselves. Currently, a design mode is not in the plans as it would take quite a bit of time but I am planning to implement loading and saving scenario files based on the current situation at the point of saving or loading.

**Keywords:** traffic modelling algorithms, microscopic traffic simulation approach, scenarios/design mode.

## 1 Introduction

### 1.1 Scope of Study

This project will be more advanced than standard web applets, but it will not be a substitute for professional traffic simulations. The idea for the project is for it to be platform-independent, as long as the computer running the application has the JRE (Java Runtime Environment). This application will provide an extensive user interface that allows the user to change program variables such as number of cars and more. Therefore, this program will be useful for getting a general sense of what would happen in a hypothetical situation by setting the variables to reflect those in the real world.

Iterative development plan includes first developing a simple prototype or layout for the program. This involves simply creating the 800x600 application window, creating a JMenu bar at the top of the screen, and dividing the lower half into a JPanel and GUI functionality. The implementation will be done in steps to allow for testing simultaneous to development.

### 1.2 Expected results

By the end of the year, this application is expected to realistically simulate real-world traffic behavior for

road networks defined by the user. The program should provide a default road network displayed at the left of the screen, but with GUI objects such as textboxes and checkboxes for which the user can modify the default road network to be similar to something found in the real world. While I do not expect this project to be as extensive as a professionally developed traffic simulation, it should be advanced enough to be of general use for modelling basic traffic situations in basic road networks.

I am desiring to learn further about the general subject of microscopic traffic simulation, using traffic algorithms that can emulate real world behavior. In addition, it will increase my proficiency in Java and using the swing API to generate effective GUI's. This project should let us see what would happen in basic road network situations including some more complicated situations such as merging. This project will also be able to examine factors such as how car speed and velocity affects traffic.

Hopefully, my project at the end of the year will allow users to see how traffic functions under multiple scenarios that they themselves can create and then save to a text file. Obviously, while these scenarios will be simple and will not include a lot of the randomness present in real world traffic situations, I am aiming for the project to accurately simulate simple traffic simulations assuming simple conditions.

In addition, the data output from this project, such as the amount of time a car spends on the screen before it goes off the screen and "dies", can be statistically analyzed by the user with simple linear regression in relation to such user-defined variables such as car volume, car velocity, car acceleration, light lengths, number of roads, braking distance or even multiple regression on any number of these user-defined variables. The data results should be consistent with what is to be expected, the more traffic there is, generally the longer it takes for a particular car to move from point A to point B. The user can reasonably expect that higher car speeds will allow cars to reach a point B, while more roads which diffuse the total car volume will also allow cars to reach the other side of the road.

### 1.3 Type of research

Use-inspired basic research would best describe my project. My project seeks to research the effectiveness of microscopic traffic simulation through use and testing in basic road networks. I pursue fundamental understanding of microscopic traffic simulation and the accompanying traffic algorithms, but the main motivation is to see the application of the concepts in an actual simulation program, which is the aim of my program.

The research is done along with the implementation of the project because I am studying how traffic simulations work out under various conditions. In addition, the process of implementing the program will show me how effective microscopic traffic simulation compared to other types of traffic simulation and how it operates on basic road networks created using user-defined or user-loaded data.

## 2 Review of current literature and research

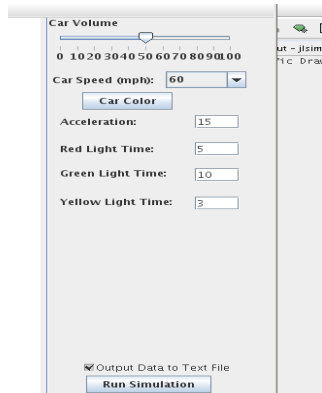
- VISSIM: Visual Traffic Simulation VISSIM is a project developed by Tom Fotherby that visually simulates traffic using the microscopic traffic simulation approach. His project is very extensive, covering 11,000 lines of Java code and even including a "design" mode for the user to define their own road networks. While the project is now discontinued, the project is open-source and downloadable via the web. This project effectively emulates real-world situations on simple road networks without accounting for person by person characteristics such as slow and fast drivers, drivers who keep a lot of distance or drivers who tailgate.

## 3 Procedures and Methodology

Resources required: Netbeans Java IDE, JGrasp, Java SE, Java JRE. Programming language: Java The main procedure will include developing from a simple prototype, a 800x600 application window with

the basic JMenu, JPanel and GUI options at the right. Functionality will be implemented step-by-step for the application. This allows for easy simultaneous testing of the program.

The GUI functions are first placed using essentially a "drag-and-drop" GUI building interface using Netbeans Form Designer, but then I will have to code the names, labels, and the event responses and event handlers to specific GUI widgets.



Screenshot of main right side GUI options.

The main program will be divided into three classes: TrafficDraw, Car, and TrafficLight. The TrafficDraw class is the main class that handles both the visual and underlying implementation of the program. Here, the program updates objects in the simulation during each frame and repaints. In addition, TrafficDraw handles all the user-defined variables and any output requested by the user. The Car class de-

fines the Car object and allows users to define specific variables for each Car object such as starting position, color, size, specific pre-set car, velocity, acceleration, and orientation (whether a car moves horizontally or vertically).

## 4 Background on microscopic traffic simulation

In contrast to macroscopic traffic simulation models, microscopic traffic simulation models simulate single vehicle-driver units, thus the dynamic variables of the models represent microscopic properties like the position and velocity of a single vehicle. In addition, the "cars" are generally represented by simple geometric shapes rather than actual 3D car objects or 2D car sprites. Allows for a large number of car objects to be on-screen at the same time and to be dynamically updated.

In addition, there is a third type of simulation model according to the level of detail on which the traffic system is based. This third type is called mesoscopic modelling and has a mixed fidelity as compared to high fidelity for microscopic modelling and low fidelity for macroscopic modelling. Microscopic represents the smallest amount of detail per individual car, but it is able to simulate much more cars and more efficiently than the other two types of simulation models.

## 5 Preliminary Testing

Input data will be mainly user-defined data instead of the default data, for example 20 cars, cars moving at 20mph and so on. In addition, the program will output data such as a measure of how "bad" the traffic is, depending on the total situation. For crash analysis, the program should eventually output dynamically individual and independent probabilities for each car currently being simulated within the traffic simulation.

Error analyses are both visually examining the program and comparing the data with those of real-world data or other established traffic simulation programs

such as VISSIM. In addition, error analyses can be checked by checking the data output of my program. The data output will include collision coordinates + probability analyses.

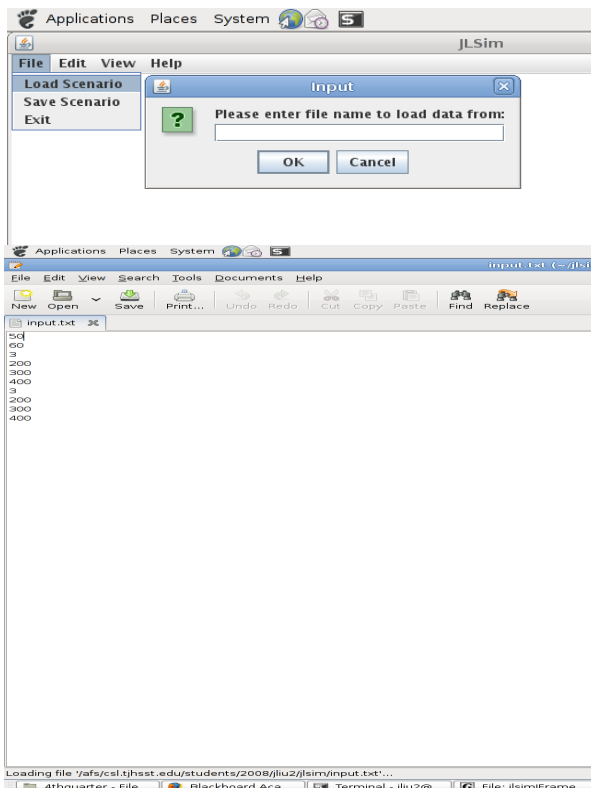
To verify the program, I will use specific structural and functional testing to verify each part of the application is working properly, from the JPanel that handles the visual traffic simulation to the user interface on the right. Eventually, dynamic testing may be used to test random inputs, but for now path and branch testing will be very effective in determining whether the application can successfully emulate real world behavior in a variety of situations.

Users may also use the load/save scenario function to save particular traffic configurations they are working with and to rerun them later. This ease of retesting a particular configuration allows for more rapid and efficient user testing.

Loading data option screenshot on top, with a sample saved text file that represents specific program variables at designated line locations.

-Basic requirements: 800x600, platform-independent, executable jar file -Advanced requirements: Accurate simulation of real-world traffic behavior, runs at a smooth frame rate (not more than 100 and not less than 30), able to run user-defined data on road networks, provides a usable crash analysis probability mechanic.

Traffic simulation algorithms with the microscopic traffic simulation approach: including algorithms to handle 1) Car speed, velocity, acceleration 2) Car breaking distance 3) Traffic light length, looping 4) Turning, switching lanes 5) Car reaction time



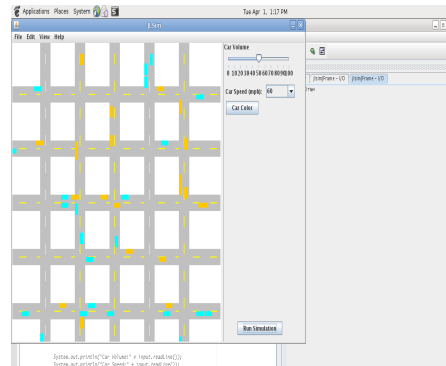
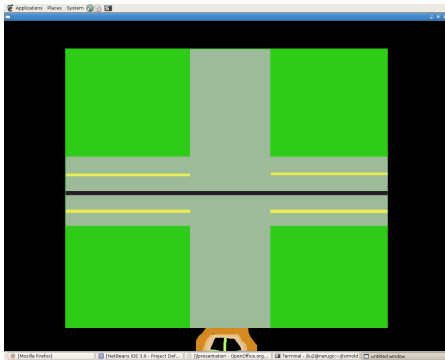
## 6 Analysis

By the end of the year, this application is expected to realistically simulate real-world traffic behavior for road networks defined by the user. The program should provide a default road network displayed at the left of the screen, but with GUI objects such as textboxes and checkboxes for which the user can modify the default road network to be similar to something found in the real world. While I do not expect this project to be as extensive as a professionally developed traffic simulation, it should be advanced enough to be of general use for modelling basic traffic situations in basic road networks.

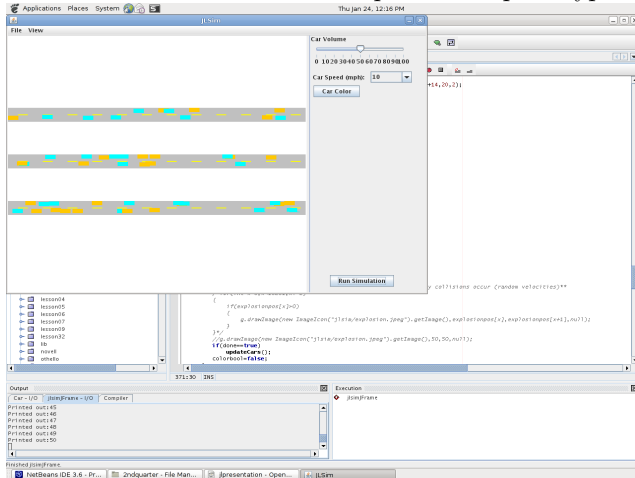
Using the microscopic traffic simulation approach, my application should allow for upwards of 50 cars on screen at the same time, all being dynamically updated at the same time. The properties and variables defined for each individual car will also update dynamically with user input change.

## 7 Quarter-by-quarter Proto- type Development

the start of 2nd quarter and completely switched over to Java/Netbeans IDE because javax.swing allows for faster and easier GUI development. In addition, jar files are much easier to distribute than relying on gcc compilations of c source files.



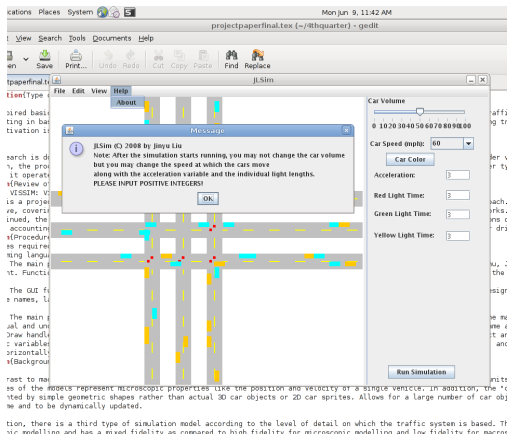
Screenshot of 1st quarter prototype



Screenshot of 2nd quarter prototype

The discrepancy between the 1st quarter prototype image and the 2nd quarter prototype image is immediately apparent. This is because I originally attempted to code my traffic simulation application using C/Glut, but decided against it at

Screenshot of 3rd quarter prototype



### Screenshot of 4th quarter prototype

As you can see, the 4th quarter prototype features traffic lights, intersections, and more extensive GUI options.

## References

- [1] T. Fotherby, "VISSIM: Visual Traffic Simulation", <http://www.tomfotherby.com/Contents/Education/Project/index.html>, 2002
- [2] E. Lieberman, A. K. Rathi, "Traffic Simulation", <http://www.tfhrc.gov/its/tft/chap10.pdf>