

CMSC 335 Project 3

Overview

In this project you will construct a Java GUI that uses event handlers, listeners and incorporates Java's concurrency functionality and the use of threads. The following Oracle Tutorials may be useful to help you become comfortable with Thread processes.

- <https://docs.oracle.com/javase/tutorial/essential/concurrency/>

In addition, a zip file is included that includes several Oracle Java files that use different types of GUI components as well as threads. I recommend going through the reading above and GUI below and the examples to become familiar before attempting the final project.

Java FX:

- https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
- https://docs.oracle.com/javafx/2/ui_controls/jfxpub-ui_controls.htm
- <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>
- <https://docs.oracle.com/javafx/2/>

Java Swing (Legacy):

- <https://docs.oracle.com/javase/tutorial/uiswing/index.html>

Assignment Details

As a new engineer for a traffic congestion mitigation company, you have been tasked with developing a Java GUI that displays time, traffic signals and other information for traffic analysts. The final GUI design is up to you but should include viewing ports/panels to display the following components of the simulation:

1. Current time stamps in 1 second intervals
2. Real-time Traffic light display for three major intersections
3. X, Y positions and speed of up to 3 cars as they traverse each of the 3 intersections

Some of the details of the simulation are up to you but the following guidelines will set the guardrails:

1. The components listed above should run in separate threads.
2. Loop through the simulation with button(s) providing the ability to start, pause, stop and continue the simulation.

3. You will need to use basic distance formulas such as $\text{distance} = \text{Speed} * \text{time}$. Be sure to be consistent and define your units of measure (e.g. mile/hour, versus km/hour)
4. Assume a straight distance between each traffic light of 1000 meters.
5. Since you are traveling a straight line, you can assume $Y = 0$ for your X,Y positions.
6. Provide the ability to add more cars and intersections to the simulation through the GUI.
7. Don't worry about physics. Assume cars will stop on a dime for red lights, and continue through yellow lights and green lights.
8. Document all assumptions and limitations of your simulation.

Submission Requirements:

1. Submit all of your Java source files (each class should be in a separate .java file). These files should be zipped and submitted with the documentation.
2. UML class diagram showing the type of the class relationships.
3. Developer's guide describing how to compile and execute the program. The guide should include a comprehensive test plan that includes evidence of testing each component of the menu with screen captures and descriptions supporting each test. Documentation includes Lessons learned.
4. Test Plan in table format: #, Description, Screenshot, PASS/FAIL

Your compressed zip file should be submitted to the Project 3 folder in LEO no later than the due date listed in the classroom calendar.

Grading Rubric:

Attribute	Meets
Design	50 points Designs a Java Swing GUI that uses event handlers, listeners and incorporates Java's concurrency functionality and the use of threads.
Functionality	100 points Contains no coding errors. Contains no compile warnings. Constructs a Java Swing GUI that uses event handlers, listeners and incorporates Java's concurrency functionality and the use of threads Include viewing ports/panels to display the following components of the simulation: <ol style="list-style-type: none"> 1. Current time stamps in 1 second intervals 2. Real-time Traffic light display for three major intersections 3. X, Y positions and speed of up to 3 cars as they traverse each of the 3 intersections The components run in separate threads.

	<p>Loop through the simulation with button(s) providing the ability to start, pause, stop and continue the simulation.</p> <p>Provides the ability to add more cars and intersections to the simulation through the GUI.</p>
Test Data and Test Plan	<p>50 points</p> <p>Tests the application using multiple and varied test cases.</p> <p>Test plan in format described in requirements.</p>
Documentation and submission	<p>50 points</p> <p>Source code files include header comment block, including file name, date, author, purpose, appropriate comments within the code, appropriate variable and function names, correct indentation.</p> <p>Submission includes Java source code files, Data files used to test your program, Configuration files used.</p> <p>Documentation includes a UML class diagram showing the type of the class relationships.</p> <p>Documentation includes a user's Guide describing of how to set up and run your application.</p> <p>Documentation includes a test plan with sample input and expected results, test data and results and screen snapshots of some of your test cases.</p> <p>Documentation includes Lessons learned.</p> <p>Documents all assumptions and limitations of your simulation.</p> <p>Documentation is in an acceptable format.</p> <p>Document is well-organized.</p> <p>The font size should be 12 point.</p> <p>The page margins should be one inch.</p> <p>The paragraphs should be double spaced.</p> <p>All figures, tables, equations, and references should be properly labeled and formatted using APA style.</p> <p>The document should contain minimal spelling and grammatical errors.</p>

Any submissions that do not represent work originating from the student will be submitted to the Dean's office and evaluated for possible academic integrity violations and sanctions.