**Project 2**

Xavier Garrido

University Of Maryland Global Campus

August 31, 2021

Test Cases:

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Pass/Fail |
| 1,1. Pause, Stop, Restart, Resume.  Add car, Pause, Resume,  Add light, Pause Resume.  Add Car, Pause Resume.  Add Light, Pause, Resume | Stop button wont mess up resume function when clicked. So It should stop, then resume, then (add Car, pause, resume add Light, pause, resume)x3. | Stop button wont mess up resume function when clicked. So It should stop, then resume, then (add Car, pause, resume add Light, pause, resume)x3. | Pass |
| 1,2 exit  1,3 exit  2,1 exit  2,2 exit  2,3 exit  3,1 exit  3,2 exit  3,3 exit | The appropriate number of cars and lights should show up for each user input. | The appropriate number of cars and lights should show up for each user input | Pass |
| 3,3 Pause at each traffic light | The distance should be 1000m between each traffic light and each car should be stopping at each light on red. | The distance should be 1000m between each traffic light and each car should be stopping at each light on red. | Pass |

Test Row 2:

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generatedTest Row 2:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Test Row 3:

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

UML Class Diagram

<https://lucid.app/lucidchart/acf46f09-e506-4f7c-ada9-03f3bb82bd9c/edit?viewport_loc=-946%2C-470%2C3328%2C1662%2C0_0&invitationId=inv_1c2f635d-8043-44cc-a492-390bc5e14400>

User’s Guide:

1. Put all java files into a package called “project3”.
2. Then compile and run the program in a JVM.
3. A GUI will pop up, use the dropdown to choose the desired number of cars and lights.
4. When you are ready to start the simulation, press “Start”.
5. When the simulation starts, car 1 is always going to be the top car, car 2 will always be the middle car, and car 3 will always be the bottom car.

Limitations:

* Max Cars 3
* Max Lights 3
* Cannot remove lights
* Cannot choose car speeds
* Cannot choose light interval times

Assumptions:

* All cars only travel on X axis, so the Y axis will not be displayed and is assumed to be 0.
* All cars stop on red light immediately and do not need to accelerate or slow down.
* The user only wants to add cars and lights, not remove them.
* The user only wants up to 3 cars and/or 3 lights.
* The user knows how to read “seconds”
* The user understands the metric system.

Lessons Learned:

I need to use arrays more for holding objects. Creating methods that create threads and hold them in arrays are actually really useful. I haven’t had to work with many projects where I needed to store objects in arrays yet, but I tried it in this project and cleaned up the code quite a bit. Instead of setting up an object and thread for each light and car, I was able to create methods that did all of that for me. I am also becoming more comfortable with using the paintComponent() method. I don’t know how often I will actually have to use it in the real world, but I finally feel like I have a strong understanding of how and when it can be useful (such as real time simulations like this project).