**UML Class Diagram**

**A screenshot of a computer

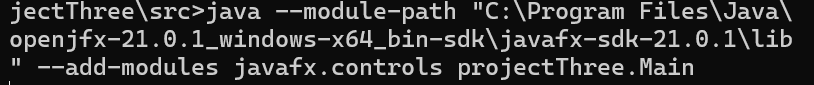
AI-generated content may be incorrect.**

**User Guide**

**How to set up and run application**

**Step 1:** After compiling the code ensure that the directory structure is similar to: \projectThree\src\projectThree for .java files and

**Step 2:** Go back to Command Prompt, type java --module-path [path to javafx-sdk] --add-modules javafx.controls projectThree.Main

****

***Figure 1.*** *Command Prompt – Running Java Program*

**Step 3:** Once a window opens, click the **Start** button.

Note: The **Start** and **Stop** button are the same button. If the simulation has not been started, then the **Start** button will be showing. If the simulation is running and/or paused, then the **Stop** button will be showing.

**A screenshot of a computer

AI-generated content may be incorrect.**

***Figure 2.*** *GUI – Main Screen*

Note: The program can only has a limit of up to five cars and stoplights due to fixed viewport size.

**Step 4.** A small window opens, click the choice box for **Number of cars** and click a number. See figure 3 below.

**Step 5.** Click the choice box for **Number of stoplights** and click a number. See figure 3 below.

**Step 6.** Click the **Ok** button. See figure 3 below.

**A screenshot of a computer

AI-generated content may be incorrect.**

***Figure 3.*** *Pop Up Window – Prompt for Input for steps 4-6*

**Step 7.** The simulation will now begin. The car is displayed via a box with a number. The stoplight is displayed via a circle and a line underneath.

Note: The cars in the simulation do not stop for yellow lights. If a car is slightly passed the line of a stoplight, it is allowed to drive on through. Cars overlap each other. This can be seen as cars being side by side. The direction of travel is left to right. Therefore, every car approaches every stoplight from the left and can only stop for a red light on the left. Lastly, from the left edge of the viewport to the right edge of the viewport is the boundary for the total distance the cars will be travelling.

To learn how to *stop* the simulation go to step 10.

**Step 8.** To pause the simulation, click the **Pause** button to the right of the **Stop** button.

Note: The **Pause** and **Continue** button are the same button. If the simulation is paused, then the **Continue** button will be showing. If the simulation is running, then the **Pause** button will be showing.

A screenshot of a computer

AI-generated content may be incorrect.

***Figure 4.*** *Running Simulation – Pause Button*

**Step 9.** To continue the simulation, click the **Continue** button to the right of the **Stop** button.

A screenshot of a computer

AI-generated content may be incorrect.

***Figure 5.*** *Paused Simulation – Continue Button*

**Step 10.** To stop/end the simulation, click the **Stop** button to the left of the **Pause**/**Continue** button.

To continue, repeat **Steps 3-7**.

**Step 11.** Click the **X** button at the top right of the window to close/end the program.

**A screenshot of a computer

AI-generated content may be incorrect.**

***Figure 6.*** *GUI – Closing the Program*

**Test Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Description | Input | Expected Output | Actual  Output | Pass/Fail |
| 1 | Checking stoplight color transition | Number of cars: 1  Number of stoplights: 1  Click Ok button | The simulation begins and a car and stoplight are visible in the viewport. The stoplight transitions from its initial color(red/green) to its next color. Green -> Yellow -> Red -> Green | The viewport contains the car object and stoplight object. The stoplight transitions to the next color after the timer reaches zero. | Pass |
| 2 | Checking stoplight timer and color transition during pausing | Inputs from test case 1 | When pausing the stoplight timer stops. When resuming the stoplight picks off from where it stops and continues.  When pausing before and after color change the stoplight behaves normally. | The stoplight’s timer stops when the simulation is paused. The stoplight’s timer resumes when the simulation continues. The stoplight color transitions normally during pausing and resuming | Pass |
| 3 | Checking Stop button while simulation is paused | Inputs from test case 1  Click Pause button  Click Stop button | Viewport clears its actors. | The window is brought back to its original state | Pass |
| 4 | Checking cars stop at red stoplights and go for green and yellow | Number of cars: 3  Number of stoplights: 3  Click Ok button | When a car is to the left of a red light it will stop. | Car 2 stops at the stoplight to the left. Car 3 drives through the green light to the right.  New Simulation: Car 2 drives through the yellow light to the left. | Pass |
| 5 | Checking if multiple cars overlap when at a red light and if the car stops. | Inputs from test case 4 | When a stoplight is red each car that arrives at the stoplight will stop. | Car 1 and 3 stop at the stoplight in the middle and have a speed of 0 km/h | Pass |
| 6 | Checking Stop button while simulation is running | Inputs from test case 4  Click Stop button | Viewport clears its actors. | The window is brought back to its original state | Pass |
| 7 | Checking Pause button for long periods | Number of cars: 5  Number of stoplights: 5  Click Pause button wait for 1 minute then click Continue button  Click Pause button wait for 5 minutes then click Continue button | The timer for the stoplights will pause when the simulation is paused and will resume when the simulation is running. The stopwatch at the top of the viewport will do the same as well. | Each timer picks off from where they stopped, so does the stopwatch. | Pass |
| 8 | Checking car distance | Inputs from test case 7 | When a car is at the end of the right side of the viewport, the distance will be close to the total distance(as seen from the top left hand side). When a car loops back to the left, the distance will be around 0. When a car is at the middle of the viewport, the distance will be half the total distance. | New Simulation: Car 4’s distance is around 5790 km on the right.  Car 3’s distance is around 336 km on the left.  Car 1’s distance is around 3081 km at the center of the screen | Pass |
| 9 | Checking top right X button during simulation | Inputs from test case 7  Click the X button | The window will close. | The window closed. | Pass |
| 10 | Checking top right X button when no simulation is running | Click the X button | The window will close. | The window closed. | Pass |

**Screenshots of Test Cases**

**Test Case 1:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 2:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 3:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 4:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 5:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 6:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 7:**

**A screenshot of a computer

AI-generated content may be incorrect.**

1-Minute Pause in between

**A screenshot of a computer

AI-generated content may be incorrect.**

5-minute Pause in between

A screenshot of a computer

AI-generated content may be incorrect.

**Test Case 8:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 9:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Case 10:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Program Assumptions and Limitations**

* The cars in the simulation do not stop for yellow lights.
* If a car is slightly passed the line of a stoplight, it is allowed to drive on through.
* Cars overlap each other. This can be seen as cars being side by side.
* The direction of travel is left to right. Therefore, every car approaches every stoplight from the left and can only stop for a red light on the left.
* The left edge of the viewport to the right edge of the viewport is the boundary for the total distance the cars will be travelling.

**Lessons Learned**

Like always with GUI programs, this project was quite challenging. The project allowed me to exercise my JavaFX knowledge and learn a couple of workarounds for certain areas of the GUI. Adding multithreading to the project was very challenging. There were many head scratching moments and bugs in the program. I did not have an easy time with the project, but the advice of starting early helped me create the final product in time. Working with System time also helped me get a better understanding of how to create timers and stopwatches with the System.getCurrentMillis() function.

The project also gave me the opportunity to practice concurrency. Concurrency and multithreading are a whole different beast when programming. There were days when the project did not progress at all, possibly even got worse, but through and through I was able to develop a better understanding of concurrency and the randomness of threads. Understanding when to use wait() and notify() was a big help. I would love to work more with multiple threads in the future.

The documentation phase was straightforward. I applied the same process as the last project onto this project. This helped me speed up the documentation process. The part that took the longest was creating the test plan and getting the screenshots of each case. This phase gave me the opportunity to better my way of documenting code.