Testing Manual for Epidemic Simulation

Test Strategy for Epidemic Simulation

Objectives

The main objectives of the testing strategy are to:

- Ensure the Simulation Application fulfills all specified functional requirements.
- Check the application's behavior under a range of scenarios and edge cases.
- Assess the user interface for both usability and accessibility.
- Verify the application's performance efficiency across varying load conditions.

Test Environment

- **Operating System**: Windows 10, macOS, or Linux
- **Java Version**: JDK 11 or later
- **JavaFX Version**: 11 or later

Setup Instructions

1. **Install Java and JavaFX**:
    - Ensure you have a Java Development Kit (JDK) installed. You can download it from here.
    - Install JavaFX SDK from here.
2. **Compile the Application**:
    - Make sure you are in the correct directory:

      Something like so:

      src/edu/rpi/cs/csci4963/u24/kims35/hw04/epidemicSimulation

```
javac *.java
```

Run the Application:

```
java --module-path /path/to/javafx-sdk/lib --add-modules
javafx.controls,javafx.fxml
edu.rpi.cs.csci4963.u24.wangn4.hw04.graph.SimulationApplication.java
```

Note: edit the `/path/to/javafx-sdk/lib` with your path to the JavaFx sdk

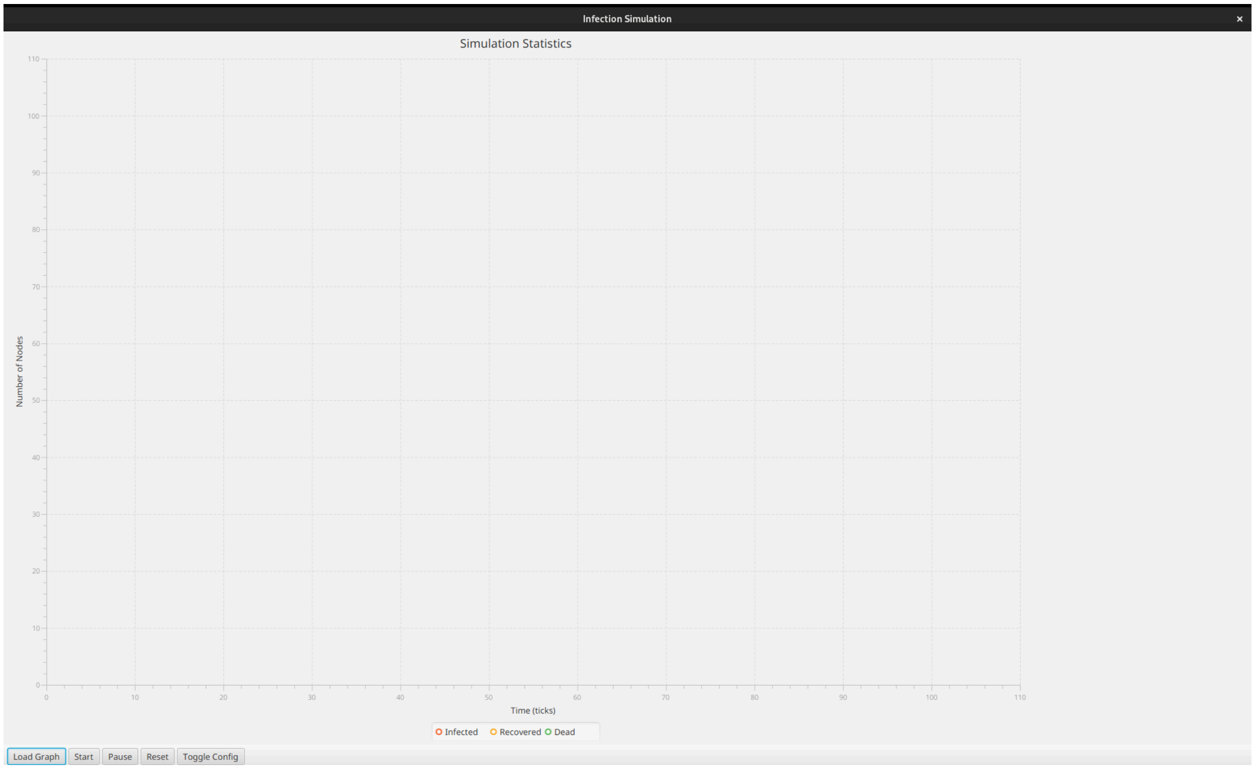
Test Cases:

Functional Tests:

Application Launch:

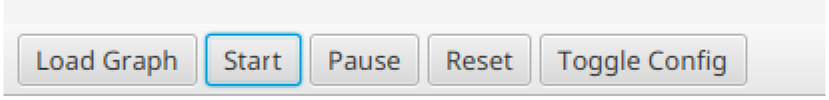
- **Test Case**: Launch the SimulationApplication.
- **Steps**:
  1. Open a terminal or command prompt.
  2. Navigate to the directory containing the compiled classes.
  3. Run the application using the command provided in the setup instructions.
- **Expected Result**: The application window should open without errors.

  Main window should look like so:

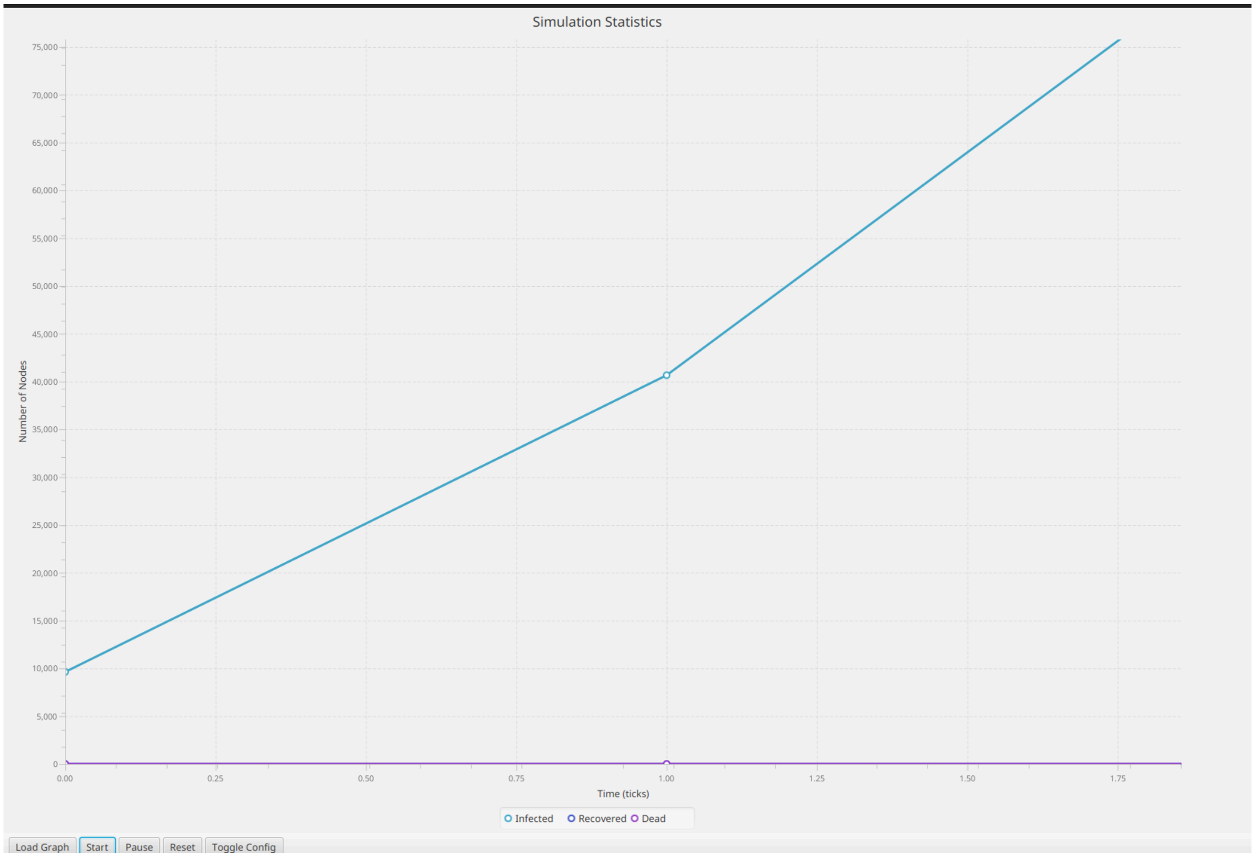


Starting the Simulation:

- Test Case: Start the simulation by clicking the Start button on the toolbar:
- Steps: After loading the graph and setting config (or just using default config values) the user can start the simulation.

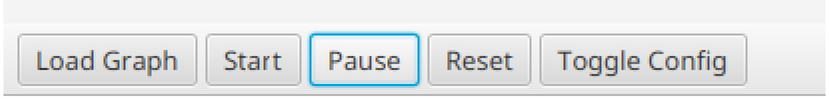


- **Expected Result:** Upon starting the graph should update (the lines start moving)

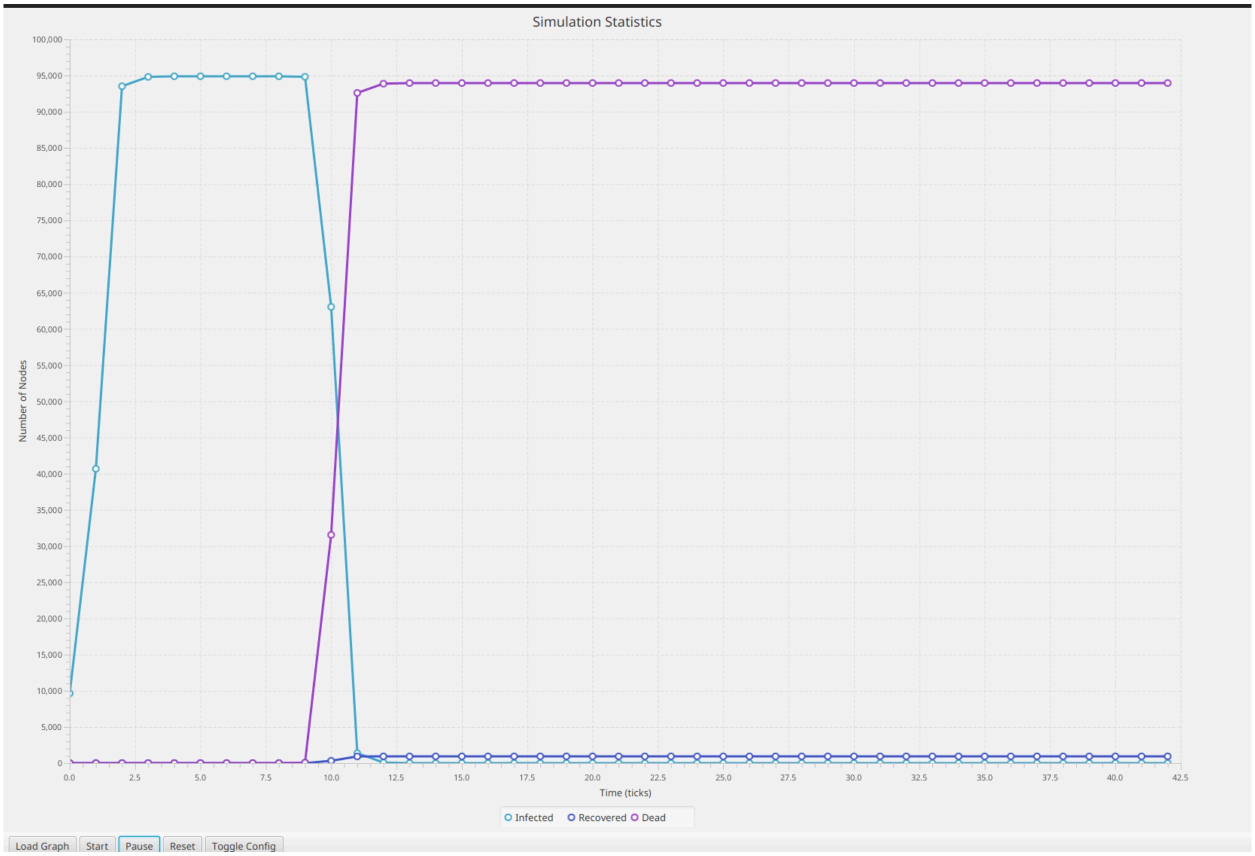


Pausing the Simulation:

- **Test Case/ Steps:** Pause the simulation by clicking the Pause button on the toolbar:



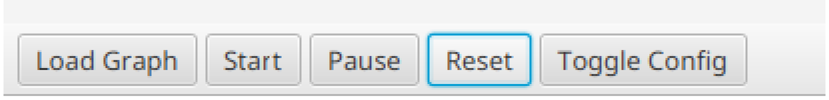
- **Expected Result:** The graph should stop updating. (lines don't move no more :0)



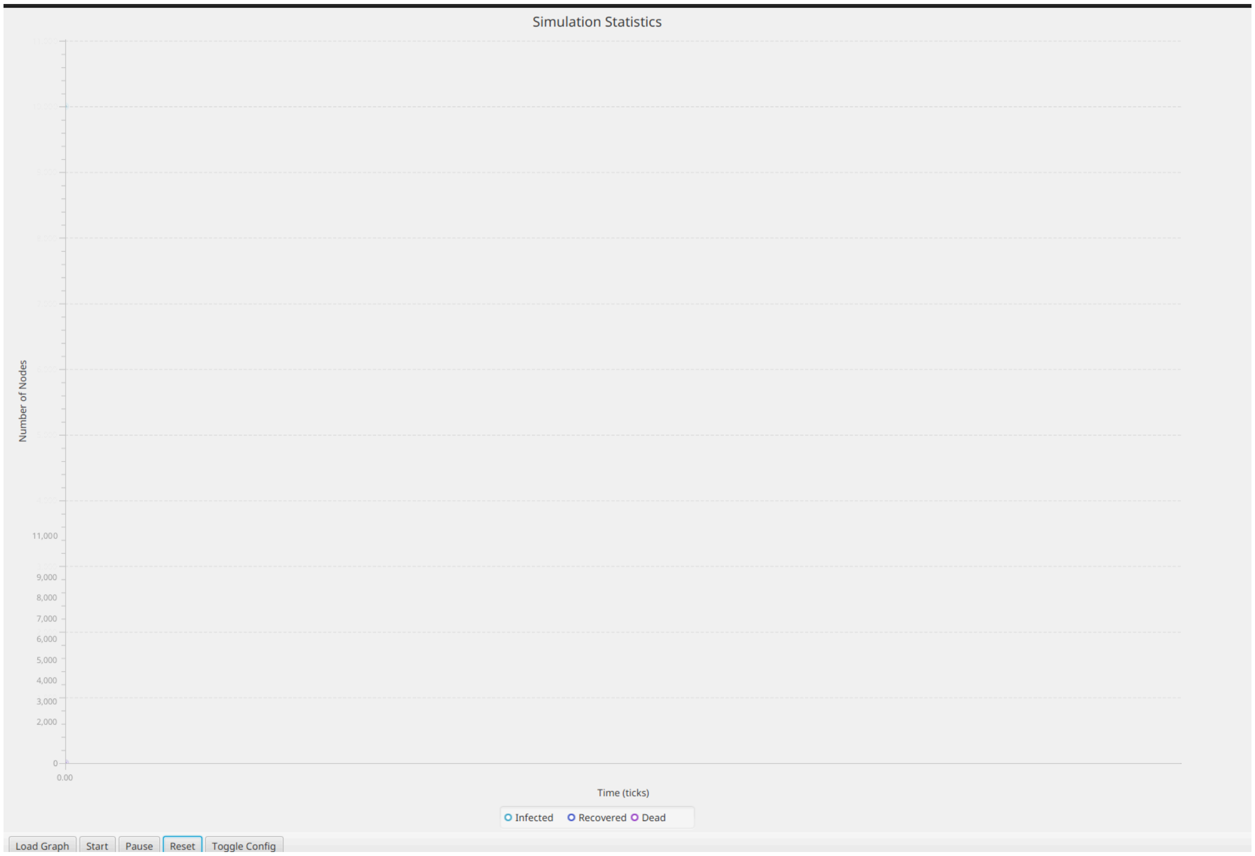
-

Resetting the Simulation:

- **Test Case/ Steps:** Reset the simulation by clicking on the Reset button on the toolbar:



- **Expected Result:** The graph should be cleared and paused.



Testing Suite Description:

To ensure the functionality and robustness of the Epidemic Simulation application, I conducted a series of essential tests as part of our comprehensive testing suite.

First, I performed boundary value tests on the graph structure and node states. These tests evaluated how the simulation handles extreme cases, such as graphs with nodes having maximum or minimum degrees, and attempts to infect nodes in sparsely or highly connected graphs. This helped confirm that the simulation processes errors correctly and provides appropriate user feedback.

Next, I implemented concurrency tests to simulate the interactions of multiple threads running the simulation simultaneously. These tests focused on the multithreading features, such as updating node states and propagating infections, aiming to uncover synchronization issues or race conditions that could occur during the simulation.

I also crafted usability tests to assess the user interface, ensuring it was straightforward and accessible. These tests verified that interactive elements, such as buttons and menus, functioned correctly in response to user actions and that the application offered suitable visual and auditory cues for various simulation events like infection spread, recovery, and death scenarios.

Furthermore, I conducted performance tests to evaluate the simulation’s responsiveness and stability under different load conditions. These included scenarios with large graphs, high numbers of initially infected nodes, and prolonged simulation sessions, ensuring the application remained smooth and crash-free.

Finally, I included tests to validate the accuracy of the simulation’s model and mechanics. These tests checked that nodes could only be infected, recovered, or marked as dead according to the defined rules, that the infection propagation rate matched the specified parameters, and that the simulation statistics were updated correctly. I simulated entire epidemic scenarios to verify that the simulation logic worked flawlessly from beginning to end.

By integrating these diverse tests, I built a thorough testing suite for the Epidemic Simulation, ensuring any potential issues were identified and resolved, ultimately providing a robust and insightful experience for users.