Testing Manual

EpidemicSimulation.java

Chev Kodama

Testing Manual

Table of Contents

Page #

1.0 Testing Overview

This section outlines the testing plan for EpidemicSimulation.java.

1.1 Testing Plan

The testing plan for EpidemicSimulation.java consists of two parts:

1. Unit testing for the Node, Graph, and EpidemicGraph classes.
2. Manual testing for the user interface (src/main/java/hw4/EpidemicSimulation.java).

2.0 Unit Testing

This section describes the unit tests implemented and how to run them.

2.1 All JUnit Test Outputs

```
+-- JUnit Jupiter [OK]
| +-- NodeTest [OK]
| | +-- testAddNeighborInvalid() [OK]
| | +-- testRemoveNeighbor() [OK]
| | +-- testConstructorValidLabel() [OK]
| | +-- testGetLabel() [OK]
| | +-- testGetState() [OK]
| | +-- testAddNeighborDuplicate() [OK]
| | +-- testNumNeighbors() [OK]
| | +-- testConstructorInvalidLabel() [OK]
| | +-- testAddNeighborValid() [OK]
| | +-- testGetNeighbors() [OK]
| | '-- testNextState() [OK]
| +-- GraphTest [OK]
| | +-- testAddEdgeInvalid() [OK]
| | +-- testGetNodes() [OK]
| | +-- testGetState() [OK]
| | +-- testInfectNode() [OK]
| | +-- testGetNeighborsNonExistentNode() [OK]
| | +-- testNumEdges() [OK]
| | +-- testNumNodes() [OK]
| | +-- testNumNeighborsNonExistentNode() [OK]
| | +-- testRecoverNodeInvalid() [OK]
| | +-- testAddNodeAfterInitState() [OK]
| | +-- testRecoverNode() [OK]
| | +-- testNodeExists() [OK]
| | +-- testRemoveNode() [OK]
| | +-- testGetStateNonExistentNode() [OK]
| | +-- testInfectNodeInvalid() [OK]
| | +-- testNumNeighbors() [OK]
| | +-- testAddEdgeValid() [OK]
| | +-- testAddNodeValid() [OK]
| | +-- testGetRandom() [OK]
| | +-- testGetNeighbors() [OK]
| | +-- testConstructor() [OK]
| | '-- testAddNodeInvalid() [OK]
| '-- EpidemicGraphTest [OK]
|   +-- testInvalidConstructorParameters() [OK]
|   +-- testInitializeGraphSuccess() [OK]
|   +-- testInfectDegree() [OK]
|   +-- testInfectRandom() [OK]
|   +-- testInitializeGraphFileNotFound() [OK]
|   +-- testInfectBFS() [OK]
|   +-- testInitializeGraphInvalidNode() [OK]
|   '-- testNextTick() [OK]
+-- JUnit Vintage [OK]
'-- JUnit Platform Suite [OK]

Test run finished after 184 ms
[         6 containers found      ]
[         0 containers skipped    ]
[         6 containers started    ]
[         0 containers aborted     ]
[         6 containers successful  ]
[         0 containers failed      ]
[        41 tests found            ]
[         0 tests skipped          ]
[        41 tests started          ]
[         0 tests aborted          ]
[        41 tests successful       ]
[         0 tests failed           ]
```

2.2 How to Run Unit Tests

Step 0) Ensure the Java Development Kit and the JUnit platform standalone console version 1.10.2 is installed on your computer.

Step 1) Open a terminal in the EpidemicSimulation directory.

Step 2) Run the following command:

Windows:

```
scripts\Run_JUnit.bat
<PATH_TO_JUNIT_PLATFORM_CONSOLE_STANDALONE_1.10.2
.JAR>
```

macOS/Linux:

```
scripts/Run_JUnit.sh
<PATH_TO_JUNIT_PLATFORM_CONSOLE_STANDALONE_1.10.2
.JAR>
```

3.0 Manual Testing

This section lists the manual tests performed on EpidemicSimulation.java.

3.1 List of Manual Tests

The EpidemicSimulation.java has passed all manual tests.

The files used for manual testing are called "invalid_node_graph" and "graph.txt". **They are stored in "EpidemicSimulation/src/test/resources".**

Description	Configurations	Input file valid?	Procedure	Expected Result
Upload valid file	**Default**	**Yes**	**Run EpidemicSimulation.java** **Click "Set Up"** **Click "Upload Graph File"** **Select "graph.txt"**	**The graph is initialized and the "Total Number of Nodes: 0" changes to "Total Number of Nodes: 1000"**
Upload invalid file (only need one invalid file test because all of the file errors are tested in the unit tests for EpidemicGraph; this test is testing that EpidemicSimulation properly relays those errors)	**Default**	**No**	**Run EpidemicSimulation.java** **Click "Set Up"** **Click "Upload Graph File"** **Select "invalid_node_graph.txt"**	**Alert error popup: "ERROR: File contains an invalid node label."**
Invalid Death chance	**Death: -0.3 The rest default**	**N/A**	**Run EpidemicSimulation.java**	**Alert error popup:**

			Click “Set Up” **Click “Edit Configurations”** **Change Death chance to -0.3** **Click the “Save” button**	**“Death chance must be between 0 and 1, inclusive.”**
Invalid Infection duration	**Infection duration: 0** **The rest default**	**N/A**	**Run EpidemicSimulation.java** **Click “Set Up”** **Click “Edit Configurations”** **Change Infection duration to 0** **Click the “Save” button**	**Alert error popup: “Infection duration must be an integer greater than 0.”**
Invalid Lambda	**Lambda: 1.1** **The rest default**	**N/A**	**Run EpidemicSimulation.java** **Click “Set Up”** **Click “Edit Configurations”** **Change Lambda to 1.1** **Click the “Save” button**	**Alert error popup: “Lambda must be between 0 and 1, inclusive.”**
Invalid Number of threads	**Num threads: 0** **The rest default**	**N/A**	**Run EpidemicSimulation.java** **Click “Set Up”**	**Alert error popup: “Number of threads**

			Click “Edit Configurations” Change Number of threads to 0 Click the “Save” button	must be an integer greater than 0.”
Infect Randomly (500)	Default	Yes	Complete all steps in the “Upload valid file” test. Click “Infect Randomly” Enter “500” Click “OK”	500 nodes are infected and the simulation begins running.
Infect by Degree (5)	Default	Yes	Complete all steps in the “Upload valid file” test. Click “Infect by Degree” Enter “5” Click “OK”	595 nodes are infected and the simulation begins running.
Infect using BFS but input a value greater than total number of nodes (5000)	Default	Yes	Complete all steps in the “Upload valid file” test. Click “Infect using BFS” Enter “5000” Click “OK”	998 nodes are infected and the simulation begins running.
Pause/Play	Default	Yes	Complete all steps in the “Infect Randomly” test. Click the “Pause” button.	The simulation stops when the pause

			Click the “Play” button.	button is clicked. The simulation begins again when the play button is clicked.
Speed up/Slow down	Default	Yes	Complete all steps in the “Infect Randomly” test. Click the “Speed Up” button 10 times. Click the “Slow Down” button 10 times.	The simulation speeds up when the speed up button is clicked. The simulation slows down when the slow down button is clicked.
Reset	Default	Yes	Complete all steps in the “Infect Randomly” test. Click the “Reset” button.	The graph is returned to the set up screen and the simulation is reset.
Reset while paused	Default	Yes	Complete all steps in the “Infect Randomly” test. Click the “Pause” button.	The graph is returned to the set up screen

			Click the “Reset” button.	**and the simulation is reset.**