INFO 6205

Program Structures & Algorithms

Final Project

Virus Transmission Simulation

Team 17:

Zechen Song, Xing Zhao

Introduction about the topic

For this project, the task is to simulate the spread of a virus transmission such as SARS-CoV-2, the pathogen behind COVID-19.

Factors need to be taken into account:

- The k and r factors of the disease
- The population density
- The usage and effectiveness of masks
- The prevalence of testing and contact tracing
- Any barriers to entry (including quarantining) into the subject area
- Any other factors that you deem appropriate

Aim of the project

The goal of this project is to learn from how to do the simulation and how to draw conclusion from the observation. In this project, we modeled and simulated the spread of virus, and took into account of how factors like the R value, population density, masks usage, mandatory testing policy, city numbers and quarantine policy impact the transmission of the virus. For this project, an interactive GUI is also provided to help the user to change different factors and simulate how the virus transmission will be when the factors has changed.

Complete project details

To simulate the relationship between increasing positive cases of covid19 over time, taking into account of masking and quarantine policy, we apply a random walk algorithm to this case. At beginning, to initialize, we create n*n two dimensions array as the city/state, and randomly add some people to the cells in this array, there is 5% of the population that has positive symptoms of covid19. For each day people randomly choose one direction to move (up, down, left, right), if two people (one is positive and one is negative) are in nearby cells, there is a possibility that negative person may be infected by covid19 positive person, we want to note that Infection rate is deponent on r value, and whether there is an mandatory mask policy. Also, people's moving speed is depend on whether we have a quarantine policy. By applying the algorithm and conditions above, after running our simulation, we got two data graphs. At the interactive GUI, the user can tweak the different factors for each simulation of whether mandatory mask policy, testing policy and quarantine policy is applied to each of the simulation. It is recommended to apply masking , testing and quarantine policy to simulation1 and none to the other, which a significant difference will be displayed. After the simulation, a graph of simulation will present increasing covid19 positive case numbers as time goes by, that actually fit pretty well with actual results.

Implementation - charts, algorithm

For this project, we implemented the Person class.

```java
public class Person {
    int x;
    int y;
    boolean isPositive = false;
    boolean isHealed = false;
    int positiveRemainDays = 0;

    public Person(int x, int y){
        this.x = x;
        this.y = y;
    }
```

The attributes x and y indicates the location of a Person object. There are also two boolean attributes indicating whether this person is infected by virus, and whether the person is healed from the disease. The positiveRemainDays attribute indicates the remaining days of recovering from the disease.

The Second class we implemented is the Node class.

```java
public class Node{
    boolean flag;
    Person person = null;
    int x, y;
    public Node(int x, int y, boolean flag){
        this.x = x;
        this.y = y;
        this.flag = flag;
    }
```

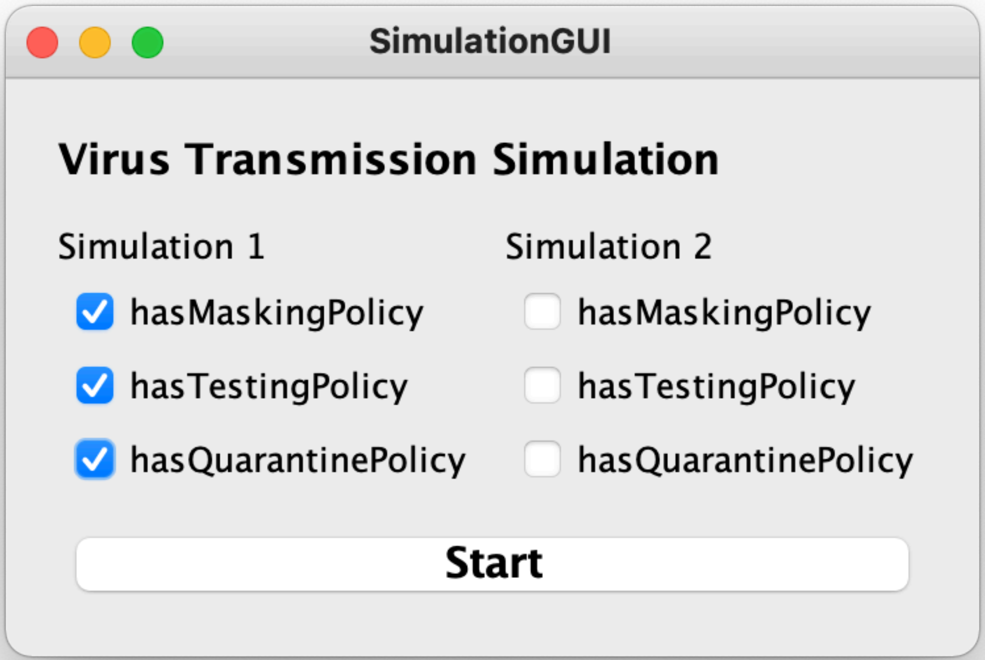
The x and y value indicate the location of a node object. And a person object is contained within the node object. If the person attribute is null of a node object, then it means this location do not have a person; While the person attribute is not null, then it means a person is contained in this location. The boolean attribute indicate whether the person is contained in this node.

Then, the model and algorithm for the simulation is implemented in the State class.

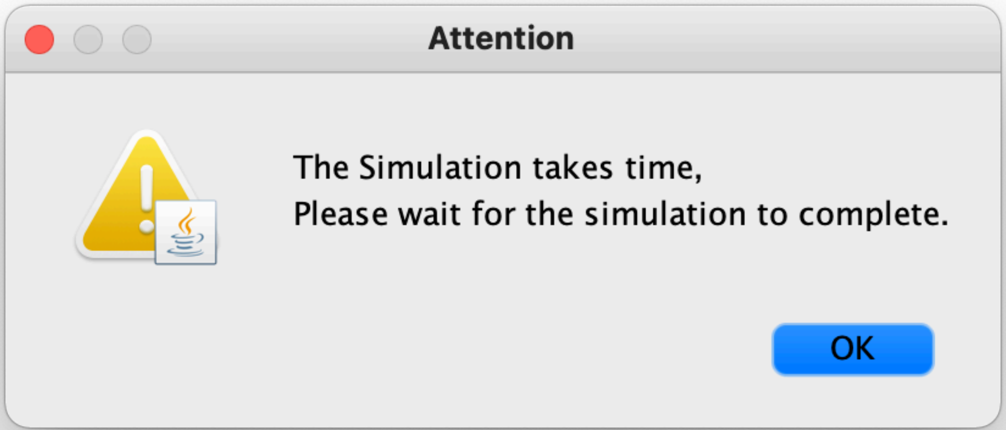
```java
public class State {
    private int len = 1000;

    private Node[][] nodeMap;

    private double rvalue = 1.2;
    private double decreasingIndex = 0.998;

    private int[] dx = {0, 0, 1, -1};
    private int[] dy = {1, -1, 0, 0};

    private List<Integer> xList = new ArrayList<>();
    private List<Integer> yList = new ArrayList<>();

    public double peopleDensity;
    public boolean hasMaskingPolicy;
    public boolean hasTestingPolicy;
    public boolean hasQuarantinePolicy;
    public int cityNum;

    public State(double peopleDensity, boolean hasMaskingPolicy, boolean hasTestingPolicy,
                 boolean hasQuarantinePolicy, int cityNum, Node[][] nodeMap) {
        this.peopleDensity = peopleDensity;
        this.hasMaskingPolicy = hasMaskingPolicy;
        this.hasTestingPolicy = hasTestingPolicy;
        this.hasQuarantinePolicy = hasQuarantinePolicy;
        this.cityNum = cityNum;
        this.nodeMap = nodeMap;

    }
```

The virus transmission simulation model is mainly implemented by the random walk. We created a 2 dimensional array as the city map for this simulation. The Node[][] nodeMap is the attribute that implementing the city map, and the attribute len = 1000 is the default length of the city map. Then the people are added to to the city map randomly based on the people density. And we choose 5% of the people added in the map to be positive at the initialization of the simulation. When the simulation begins, each person randomly choose to move to one of the four directions. The attributes dx and dy are to implementing the random walk algorithm. When 2 persons meet (2 persons in the same location), there is some possibility that the healthy one infected by the virus positive one. Whether the person is infected by another one, is depend on the r value, and whether there is mandatory mask policy and mandatory testing policy. And the speed of the moving of each person in the city map is determined by whether there is a quarantine policy. And the default simulation time is 500 days. The xList for a simulation is to store the days, while the yList stores the positive number of the population in the specific day. For this project, 2 simulations are performed each time. Then the xList1 and yList1 for simulation1, and xList2 and yList for simulation2 are added to the XYSeriesCollection. And the XYSeriesCollection is used to illustrate the simulation of virus transmissions, which outputs an image for the simulation.

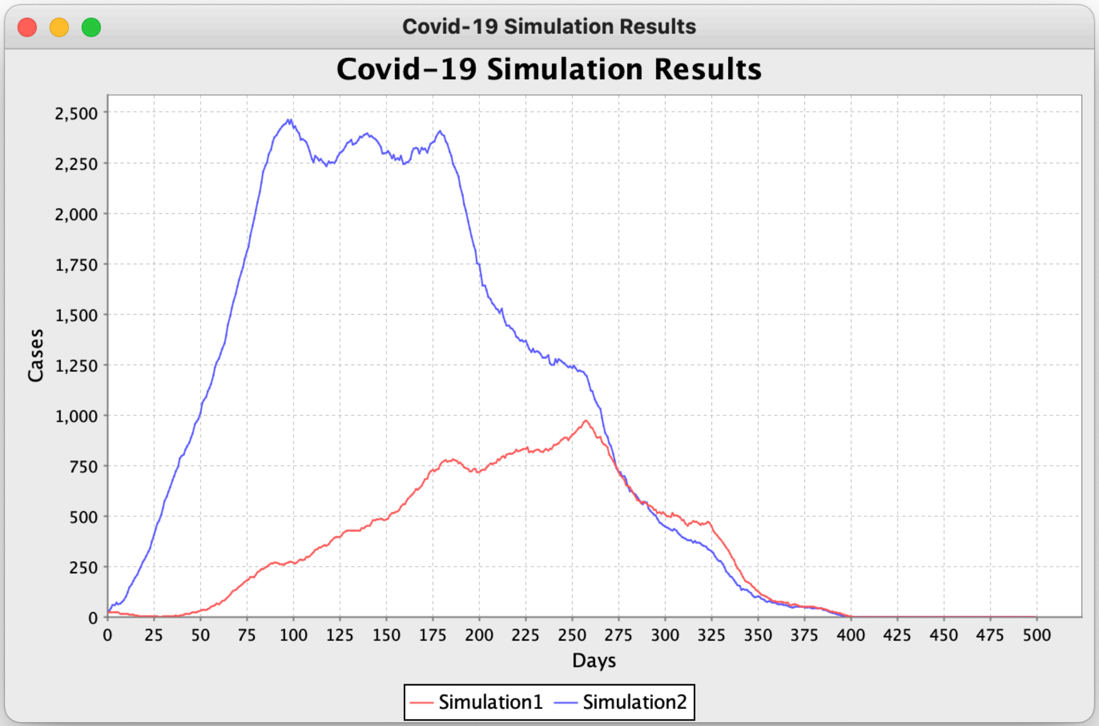
An iterative GUI was also implemented in this simulation.



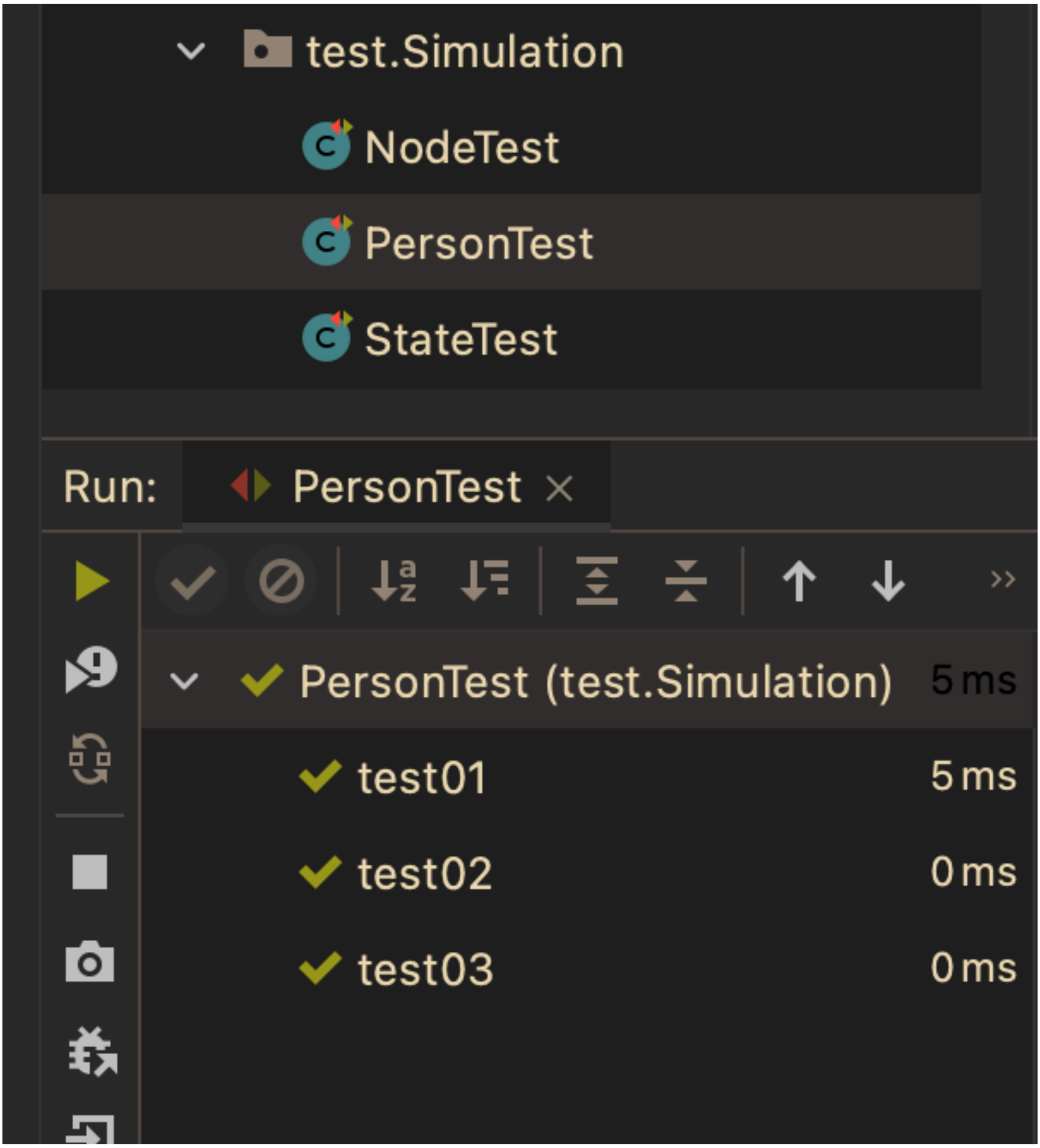
In the GUI of the simulation, the client can tweak the different factors of each simulation. We use Java Swing to implement the graphic user interface, and the main method is implemented in the SimulationGUI class. When user clicks the Start button, the main method of the program runs, and the simulation begins. and the simulation takes about half a minute, the program will ask you to wait for the simulation to complete.

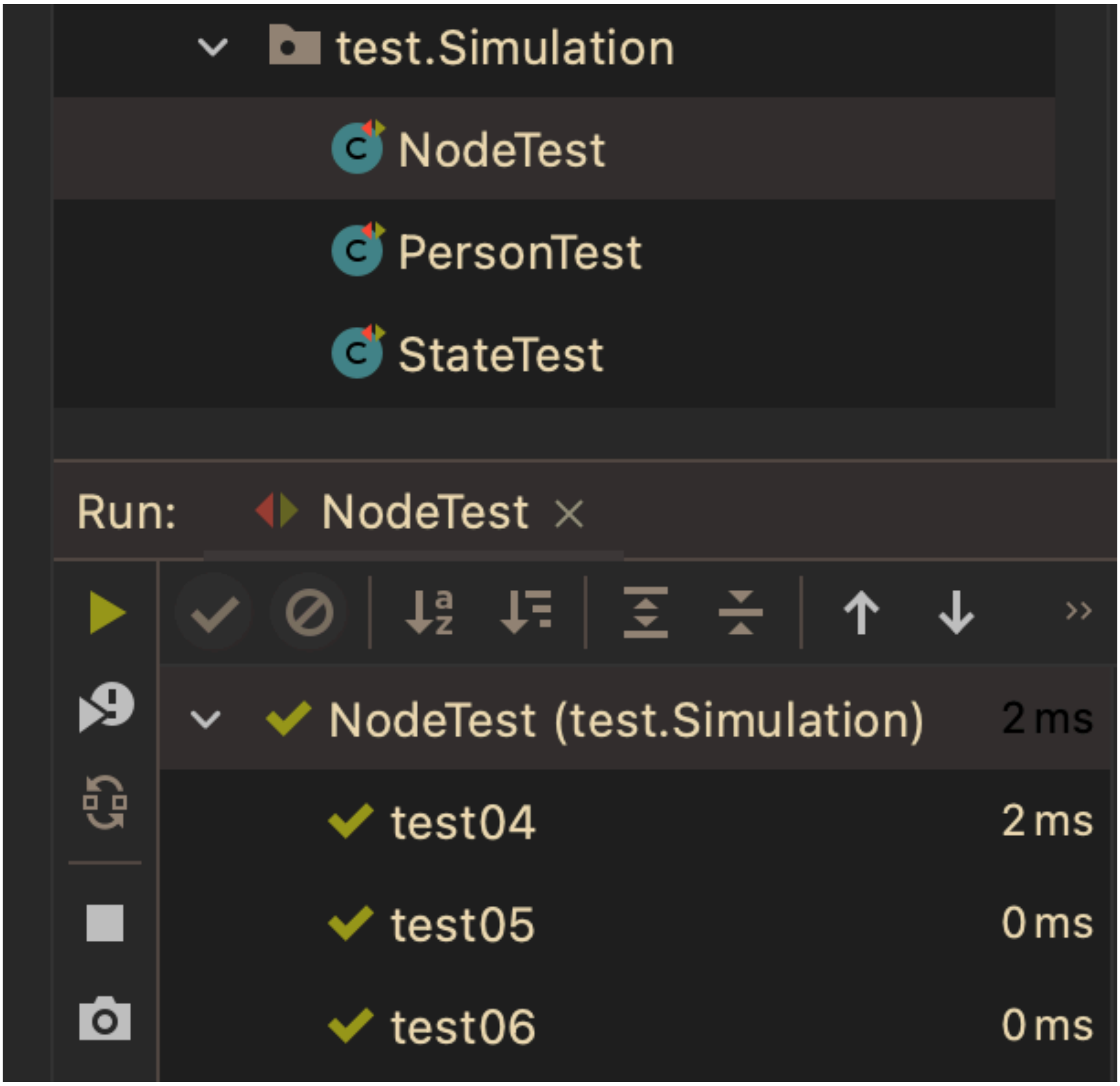


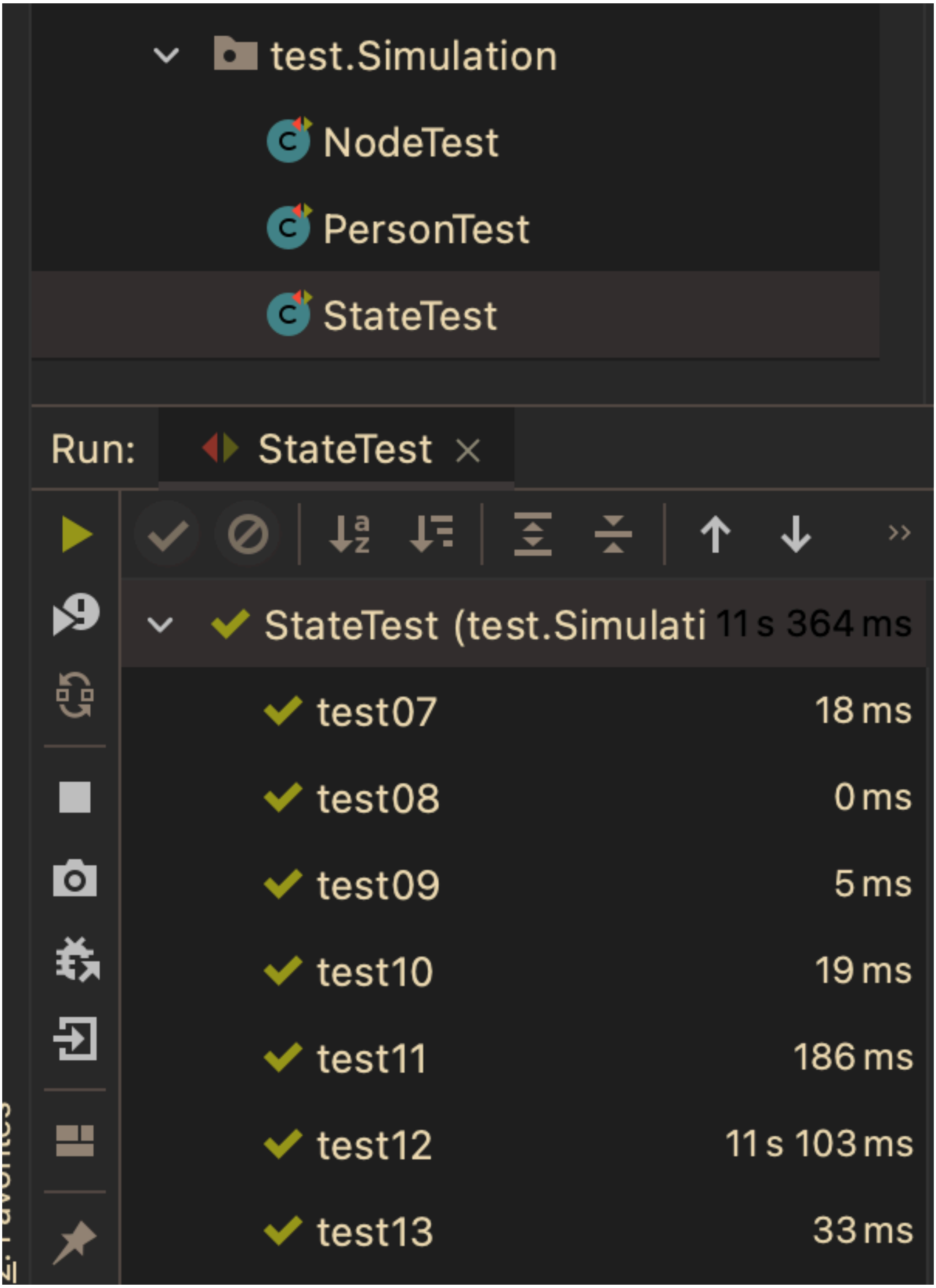
Once the simulation complete, a graph of the simulation will be displayed.



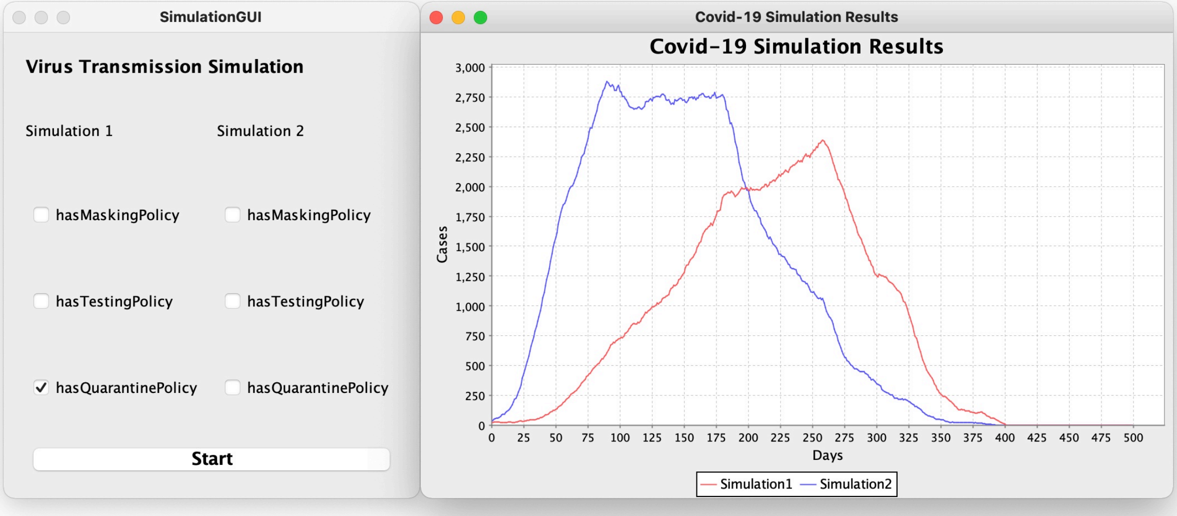
The JUnit tests for each class are implemented in the project.


test.Simulation
NodeTest
PersonTest
StateTest
Run:
PersonTest
PersonTest (test.Simulation) 5 ms
test01 5 ms
test02 0 ms
test03 0 ms

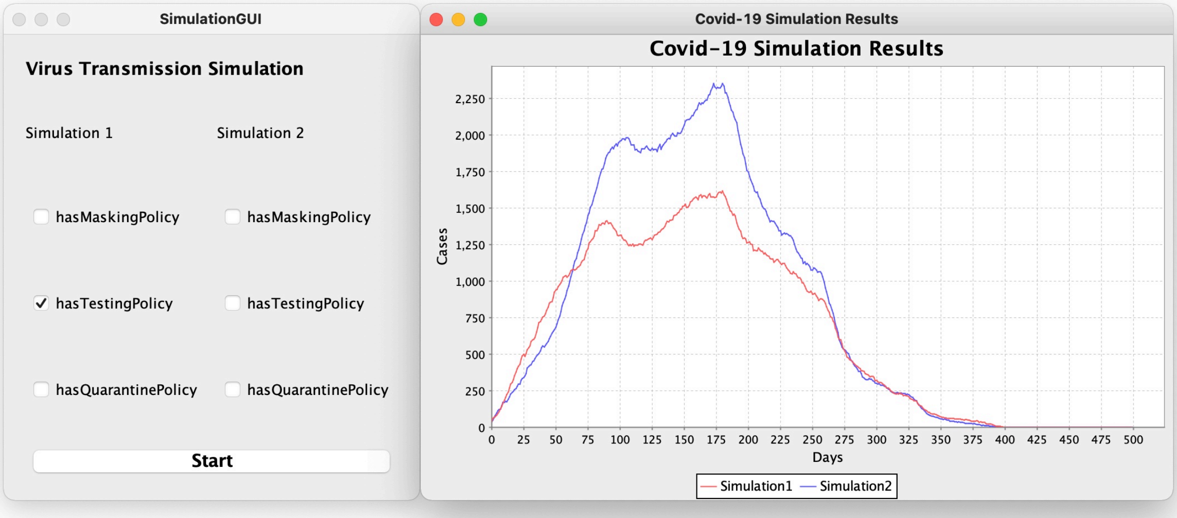

test.Simulation
NodeTest
PersonTest
StateTest
Run:
NodeTest
NodeTest (test.Simulation)
2 ms
test04
2 ms
test05
0 ms
test06
0 ms


test.Simulation
NodeTest
PersonTest
StateTest
Run:
StateTest
StateTest (test.Simulati 11 s 364 ms
test07 18 ms
test08 0 ms
test09 5 ms
test10 19 ms
test11 186 ms
test12 11 s 103 ms
test13 33 ms

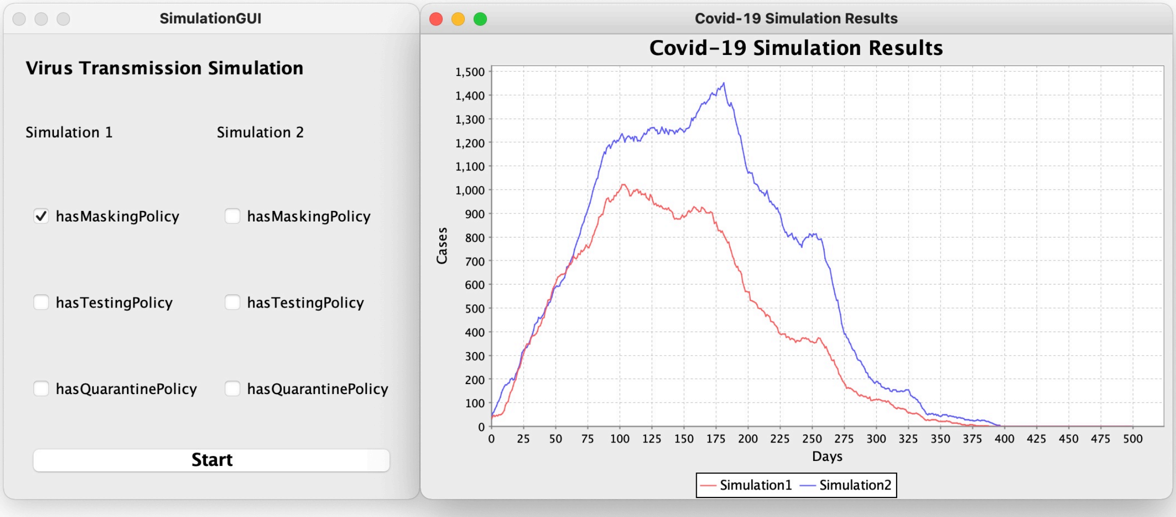
Output

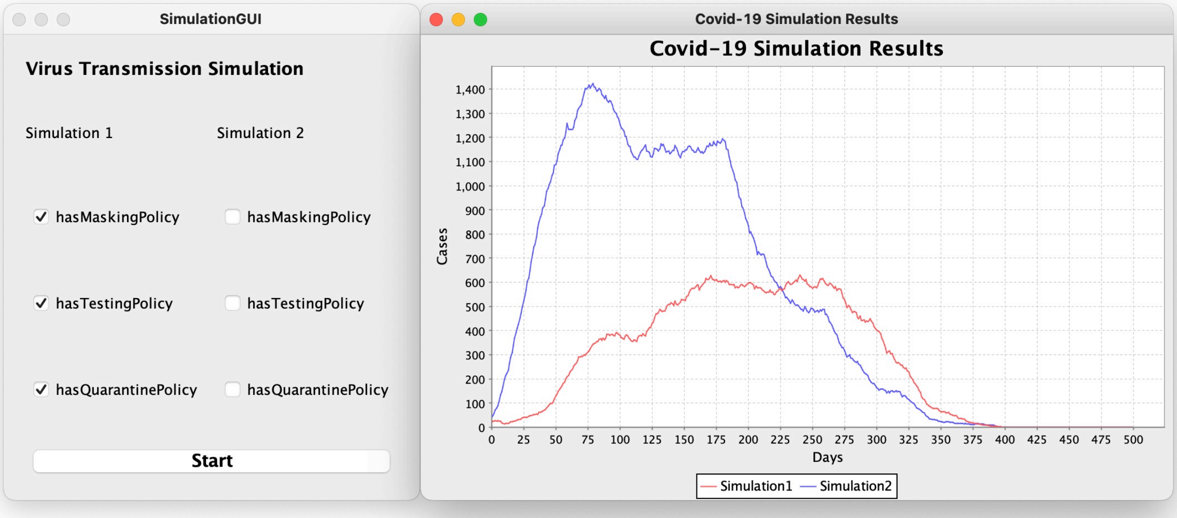


SimulationGUI
Virus Transmission Simulation
Simulation 1
Simulation 2
hasMaskingPolicy
hasMaskingPolicy
hasTestingPolicy
hasTestingPolicy
hasQuarantinePolicy
hasQuarantinePolicy
Start
Covid-19 Simulation Results
Covid-19 Simulation Results
Cases
Days
0 25 50 75 100 125 150 175 200 225 250 275 300 325 350 375 400 425 450 475 500
0 250 500 750 1,000 1,250 1,500 1,750 2,000 2,250 2,500 2,750 3,000
Simulation1
Simulation2

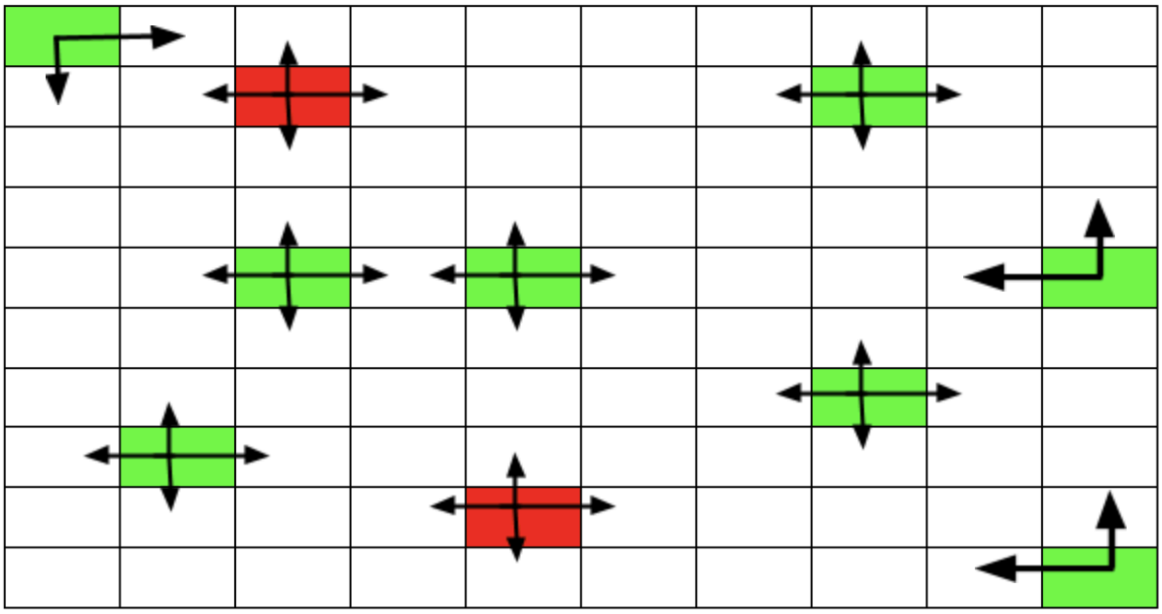


SimulationGUI
Virus Transmission Simulation
Simulation 1
Simulation 2
hasMaskingPolicy
hasMaskingPolicy
hasTestingPolicy
hasTestingPolicy
hasQuarantinePolicy
hasQuarantinePolicy
Start
Covid-19 Simulation Results
Covid-19 Simulation Results
Cases
Days
0 25 50 75 100 125 150 175 200 225 250 275 300 325 350 375 400 425 450 475 500
0 250 500 750 1,000 1,250 1,500 1,750 2,000 2,250
Simulation1
Simulation2





Mathematical analysis/evidence



The Random walk model simulation the people in the real world. Each person travels to a towards a random direction at each day. When a tested positive person meet a healthy person at some location, whether the virus is transmitted to the healthy one is decided by r and different policies, which simulation the virus transmissions in the reality properly.

Conclusion

After implementing the virus transmission simulation project, we tested different settings to initialize the simulations. With an amount of the experiments taken, the outputs of the simulation shows clearly the how these factors influenced the simulation. By our observation, each of the masking policy, testing policy and quarantine policy will help decrease the growing speed of the virus transmission, and reduce the total infected number of the virus. To conclude, strictly enforcing the mandatory mask policy, testing policy and quarantine policy will slow down and reduce the severity of the virus transmission.