CSE443

Object Oriented Analysis and Design

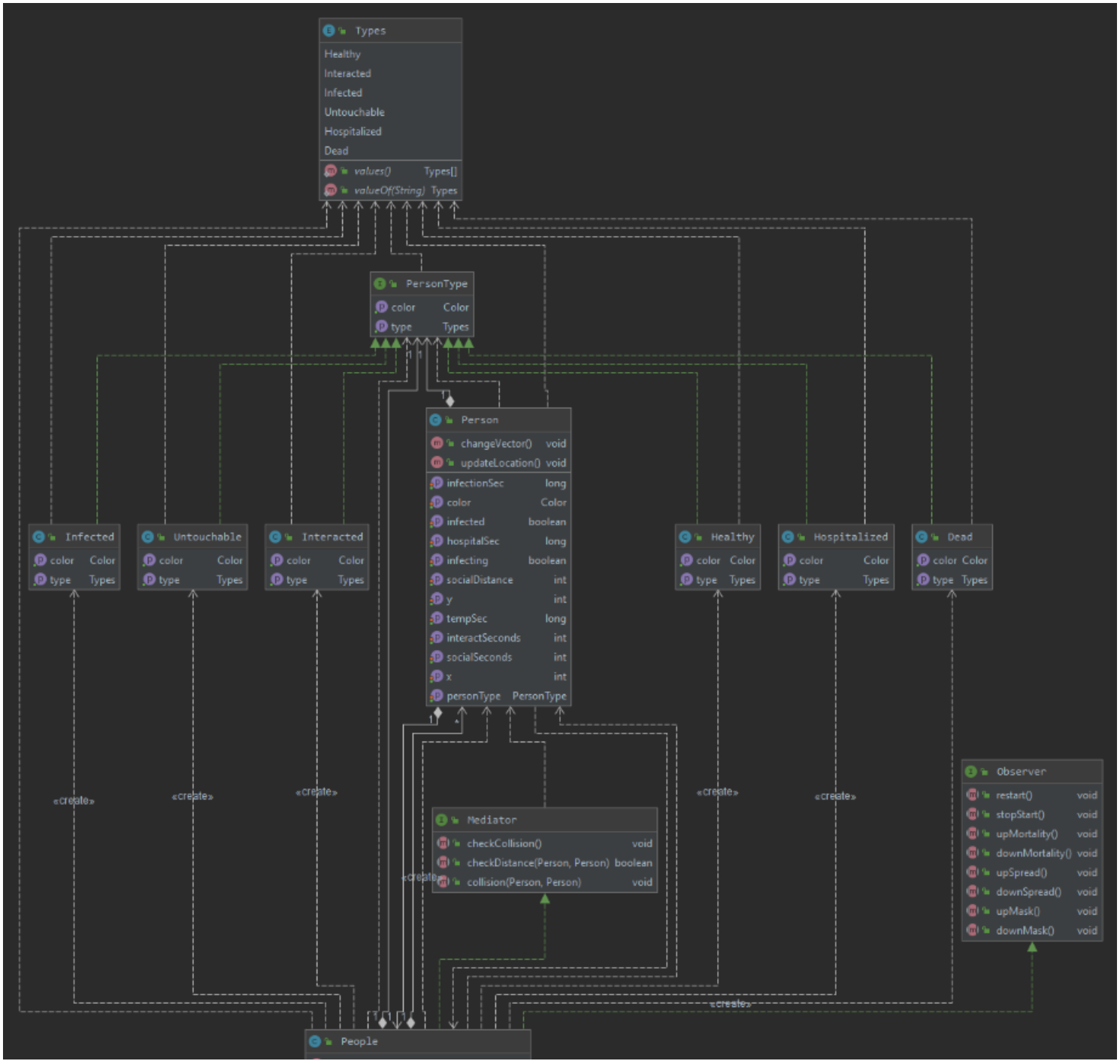
FINAL REPORT

Alperen Erdem 161044027
Jan 18

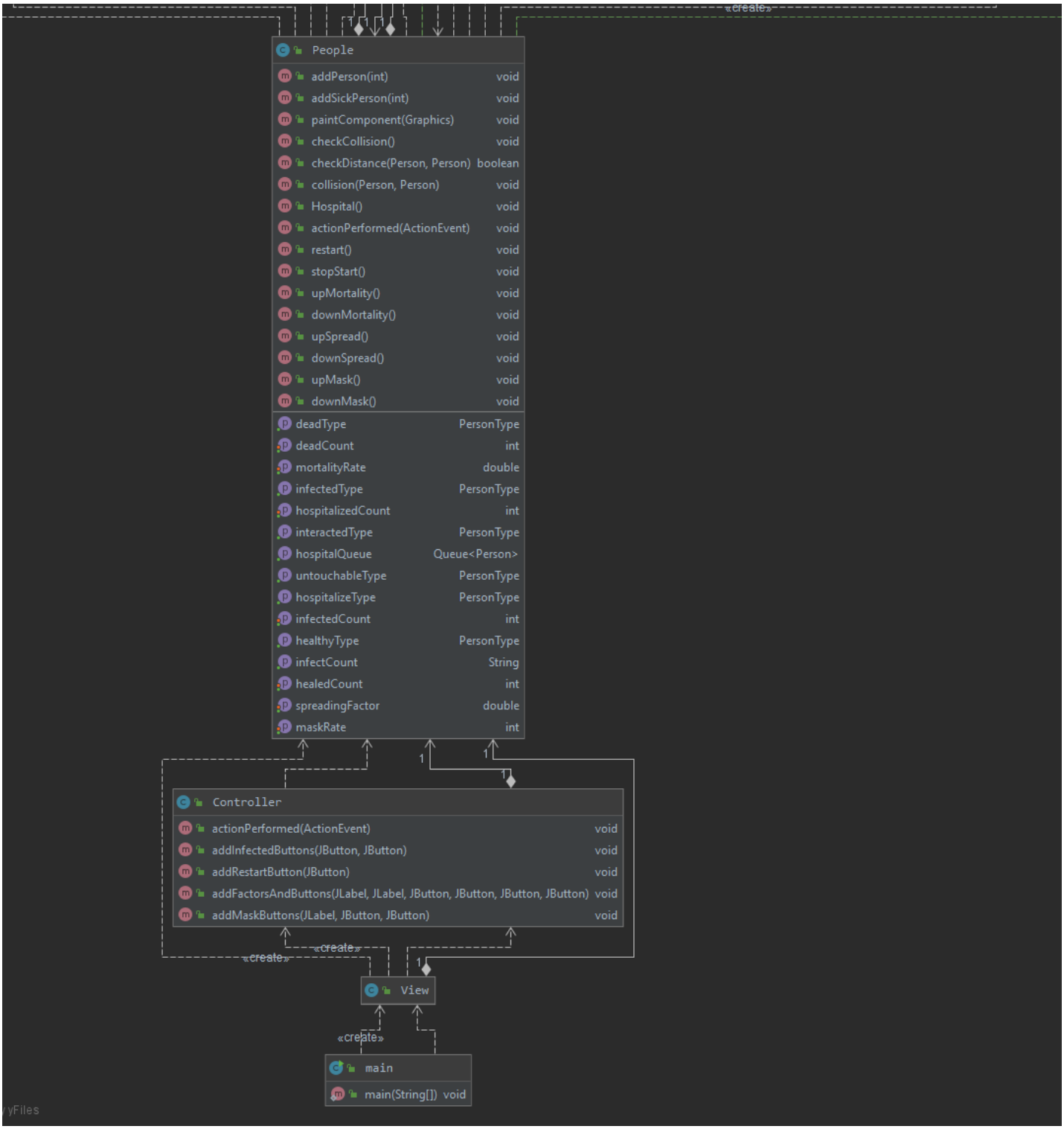
Contents

UML Diagrams

Flyweight,Mediator and Observer Side



MVC side


People
addPerson(int) void
addSickPerson(int) void
paintComponent(Graphics) void
checkCollision() void
checkDistance(Person, Person) boolean
collision(Person, Person) void
Hospital() void
actionPerformed(ActionEvent) void
restart() void
stopStart() void
upMortality() void
downMortality() void
upSpread() void
downSpread() void
upMask() void
downMask() void
deadType PersonType
deadCount int
mortalityRate double
infectedType PersonType
hospitalizedCount int
interactedType PersonType
hospitalQueue Queue<Person>
untouchableType PersonType
hospitalizeType PersonType
infectedCount int
healthyType PersonType
infectCount String
healedCount int
spreadingFactor double
maskRate int
Controller
actionPerformed(ActionEvent) void
addInfectedButtons(JButton, JButton) void
addRestartButton(JButton) void
addFactorsAndButtons(JLabel, JLabel, JButton, JButton, JButton, JButton) void
addMaskButtons(JLabel, JButton, JButton) void
View
main
main(String[]) void
«create»
1

Design Patterns I Used

1-Flyweight Pattern

All my person objects sharing common features. Like all healthy ones act same and share same color. So instead of creating new variables for every object, I made a PersonType interface with color and type data. My People class acts like FlyweightFactory and creating all types.

Depend on interactions and time, type reference in person object is changing.

2-Mediator Pattern

Instead of making direct interactions between Person objects, my People object which implements mediator interface making these interactions for them. For example, checking their collision, distance, probability of getting sick…

3-MVC (Observer Pattern)

MVC is not design pattern but I used observer pattern while implementing MVC.

My people object also implements observer interface, Controller acts like subject. Depend on pressed buttons, controller notifying People, and People object is updating its values.

My people object also acts like part of view, it controls the area which person objects moves.

In my view object I'm creating all views and buttons, creating initial view of program, and sending their reference to mediator and controller object for future modifications.

Explain of Important Parts of Project

Hospital

I have hospital function in people class, with queue data structure of hospital and hospital queue. Queue data structure is perfect for hospital part of project.

All infected people are attending to hospital queue. After 25 seconds, they want to go hospital.

Hospital have capacity of total population/100. If there are empty places, infected people can come to hospital. If it's full, hospital will accept first person from hospital queue after one person is healed.

In the hospital, nobody can die, but in the hospital queue they can.

Collisions

When timer ticks, people object checking all person objects. If there are two object who are in collision distance (min{D_1, D_2}), that means they are in interacted status now. After collision for preventing loop collision, I made them untouchable (immune to collisions) for 0.5 seconds.

If there is one infected and one healthy, with help of probability function, healthy one can be infected.

Test Cases

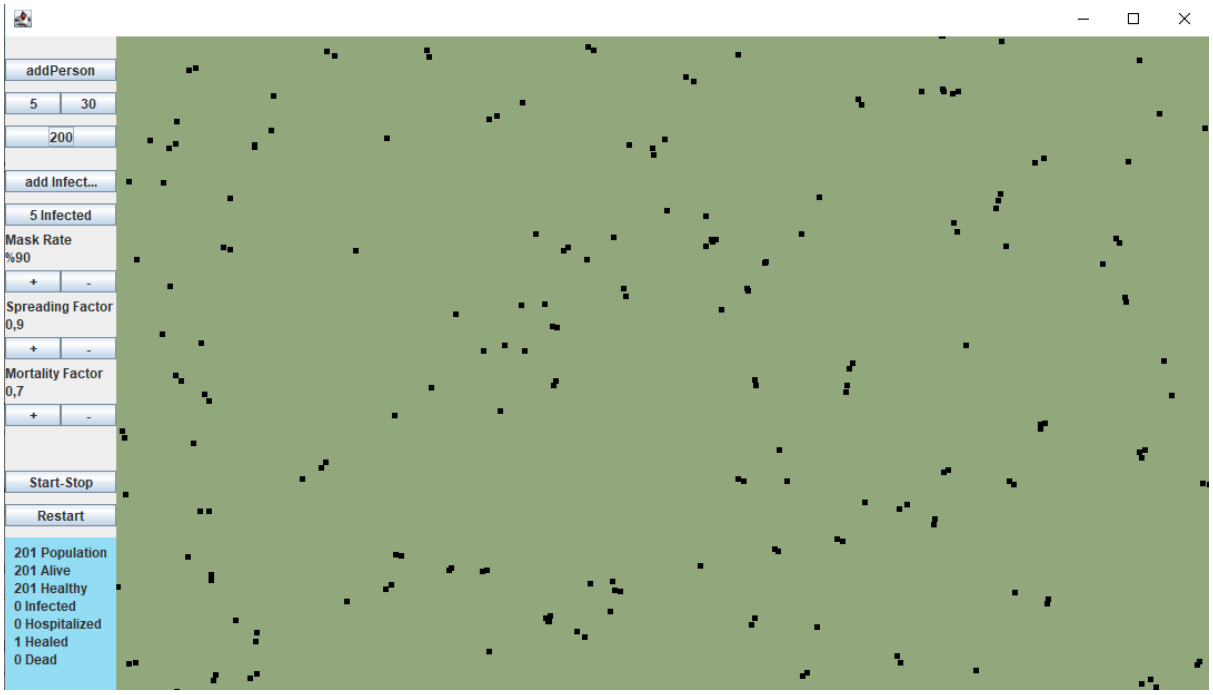
Before checking test cases I want to explain, I didn't change social distance from outside. Social distance is a random value which created in initialization like their sociality level.

Results can be vary, getting infected is probability.

Mask Wearing Rates

If %90 of population (even infected one) wear masks.

Spreading Factor= 0.9 Mortality Factor= 0.7



If %50 of population (even infected one) wear masks.

If %10 of population (even infected one) wear masks.

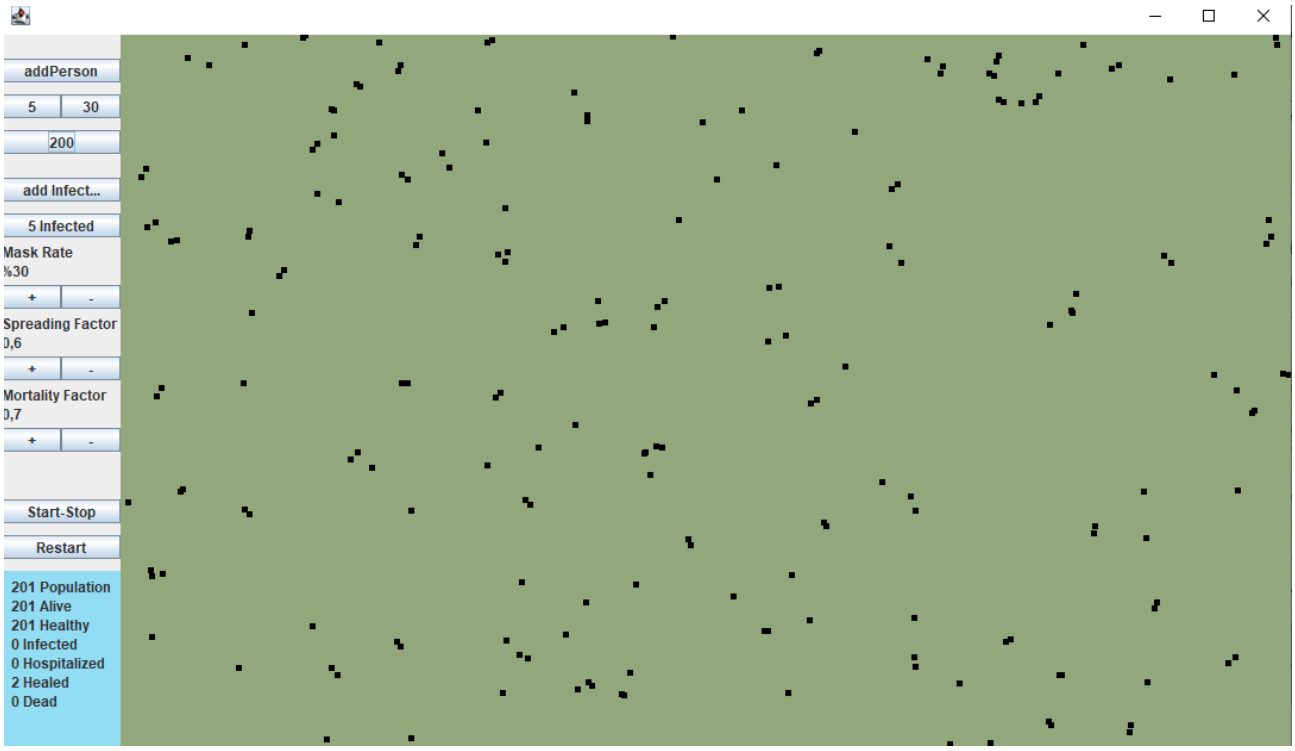
Spreading Factor

Mortality Factor = 0.7 Mask Rate = %30

When spreading factor is 1.0

When spreading factor is 0.8

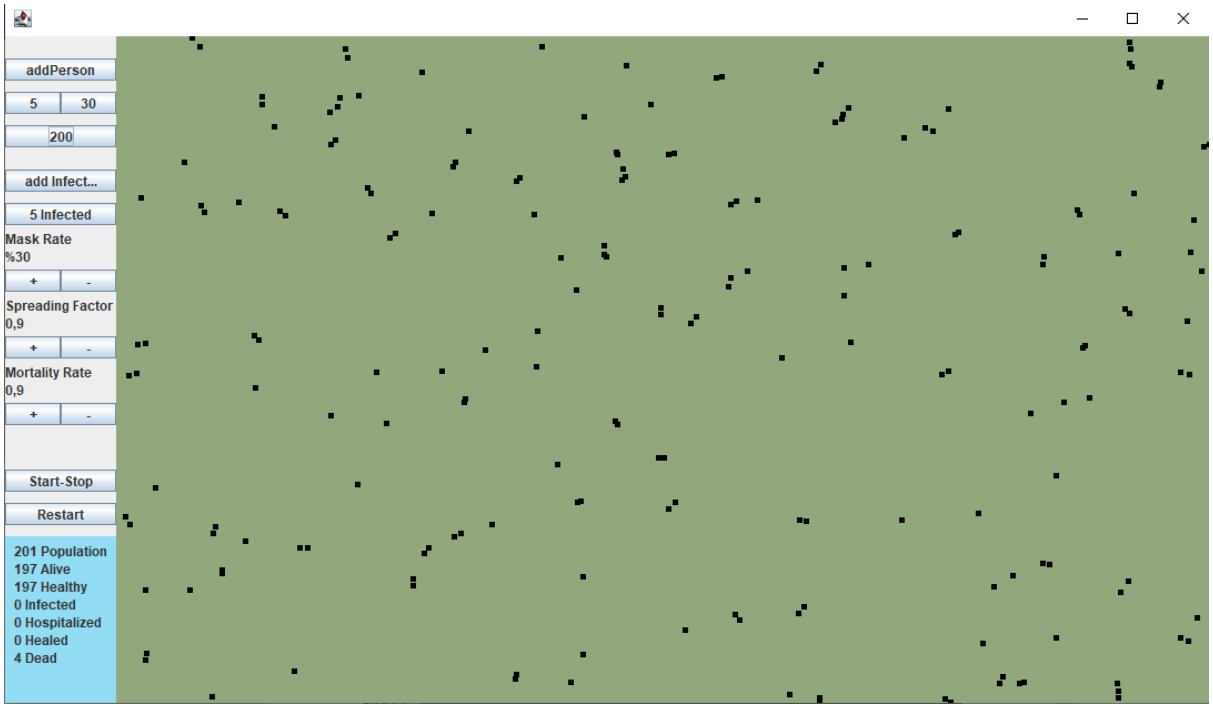
When spreading factor is 0.6



Mortality Factor

Spreading Factor= 0.9 Mask Rate= %30

When mortality factor is 0.9(that means no one can go hospital, they will die in 10 seconds)



When mortality factor is 0.7

When mortality factor is 0.4

When mortality factor is 0.2

Results

According to test cases, both spreading factor, mortality factor and mask rate directly affects results.

For example, if high percentage of population wear masks, Infected people will be cured or died before infecting anyone. So, lots of people won't face any disease. But lowering mask rate is increasing infected and dead count.

Lowering spreading factor is affecting as much as mask rate. Because if infected people can't spread disease, the disease will disappear.

Mortality rate is important too but in a different way. If infected people die too fast, they don't have time for spreading disease. Longer they alive, they have more time for spreading.

Social distance and sociality level of person is important too. Both are random personal variables in my design. So, I couldn't test them. Increasing social distance and sociality level will lower spreading.

And hospital capacity is another important thing. Because hospital is curing people without fail. With enough of hospital capacity (ventilator count), we can cure all infected people.

If our aim is keeping most people alive, wearing mask, increasing hospital capacity, increasing social distance, lowering sociality, and dying fast are most important ones.

If our aim is keeping most people healthy, wearing mask, lowering sociality and dying fast are most important ones.