Gebze Technical University

Computer Engineering

CSE443

Object Oriented Analysis and Design Fall 2020-2021

Final Project

161044044

Erkan YILMAZ

1)INTRODUCTION

1.1)Problem Definition

Our task is design and implement a visual simulation of an epidemic within a human society.

Context:

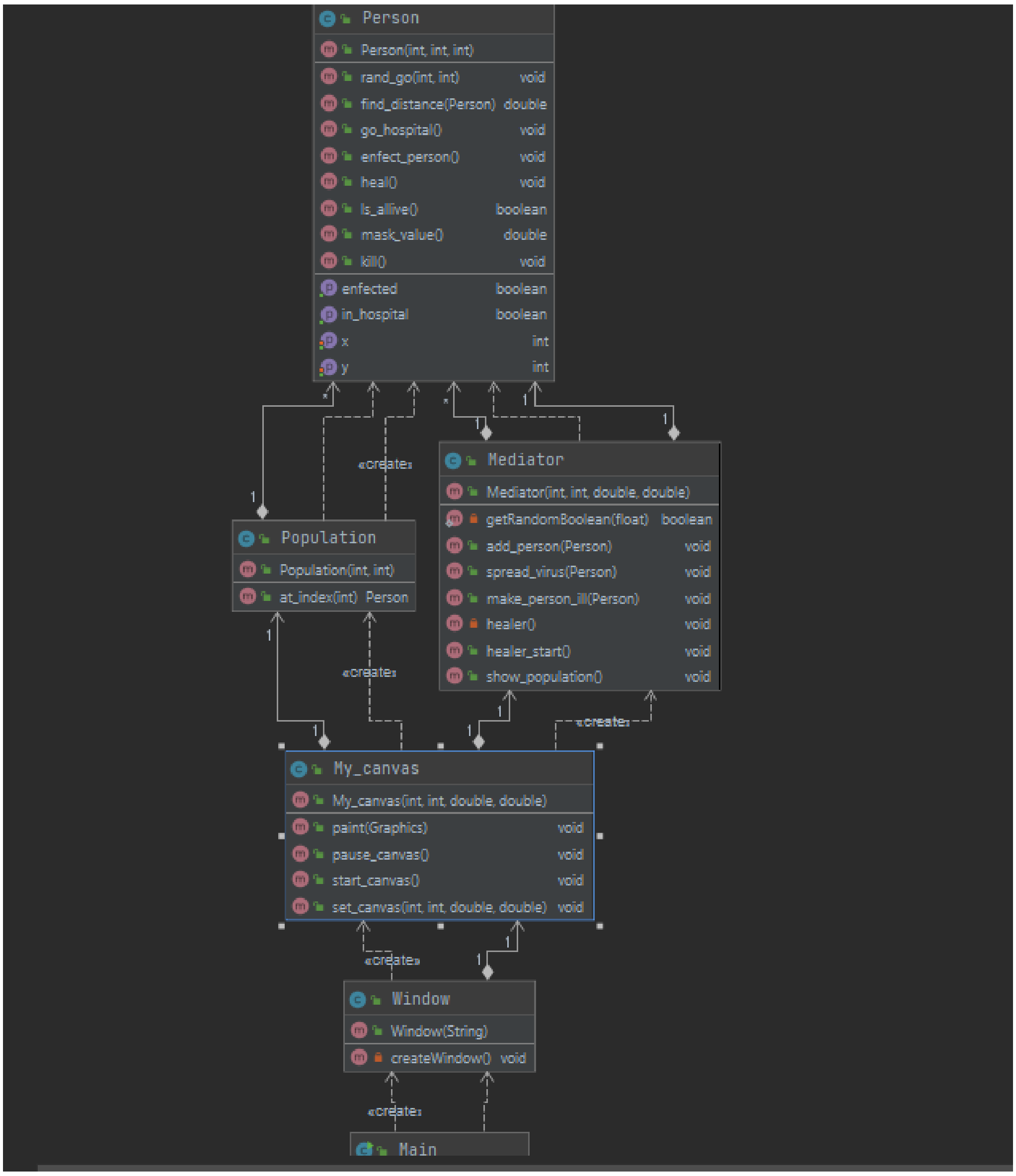
The society will be modeled as an empty 2D canvas of size 1000x600 pixels. Each individual in this society of population Po will be modeled as a square of size 5x5 pixels on this canvas, positioned randomly. Each individual will move on this 2D canvas along an initially random direction, at a constant speed of S pixels/second, and when s/he collides with another individual s/he will spend C seconds with her/him at the collision spot to simulate social interaction with a social distance D and then both will continue moving at once againt randomized directions. Nobody can escape the canvas, if an individual reaches the edge of the map, her/his movement direction should be once again randomized. At the beginning there will be one random infected individual in the population. Each individual will possess a numerical value indicating whether they wear a mask (M=0.2) or not (M=1.0), their speed S \in [1,500] of movement in pixels/second, the social distance D \in [0,9] (in pixels) that they practice when they collide with other individuals, and how social they are in terms of C seconds \in [1,5] they spend with every individual they collide with. The disease will have a constant spreading factor R \in [0.5,1.0] and a constant mortality rate Z \in [0.1, 0.9]. When two individuals with coefficients C_1 and C_2 collide they stay together (at collision position) for time C=max{C_1,C_2} to simulate interaction and then continue their randomized courses. If another individual is in collision course with either of them in the meantime, s/he cannot interact with them and ignores them as if they weren't there. Let two individuals I_1 and I_2 collide, with mask statuses M_1 and M_2, and social distances D_1 and D_2 respectively. Let I_1 be infected and I_2 be healthy. They stay together for a duration C=max{C_1,C_2} before parting, and the social distance between them is D=min{D_1,D_2}. The probability of I_1 infecting I_2 is P = min(R * (1+C/10) * M_1 * M_2 * (1-D/10),1) - An infected individual will die after 100 * (1-Z) seconds and disappear from the canvas. - Update the canvas every second or less; provide a timer, and show the total count of infected, healthy, hospitalized and dead. - Every infected individual, 25 seconds after her/his initial infection will be assumed to be at the hospital and will be removed temporarily from the canvas. The hospital however is assumed to have only B=Po/100 ventilators. After staying at the hospital for 10 seconds s/ he will return to the society at a random position as healthy. If the hospital ventilators are all full the individual will remain and continue moving/infecting in the society, until a ventilator becomes available or...s/he dies. The hospital is assumed to be able to cure all cases.

1.2)System Requirements

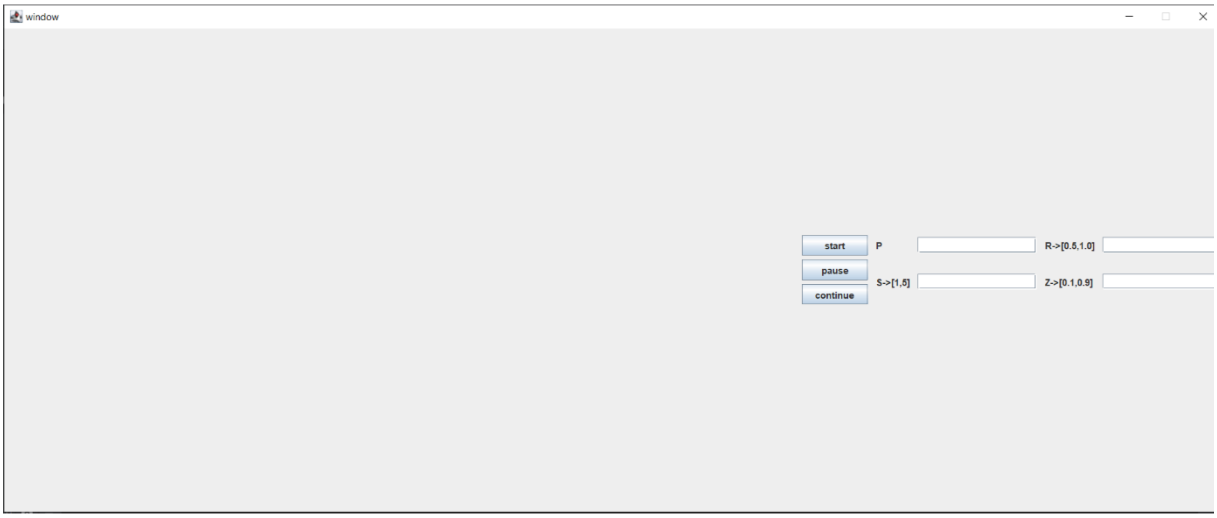
You must have java installed on your operating system to run programs. Running programs with the current version of Java will be useful for efficiency. Programs can work in both windows, linux and other operating systems installed java. The program was written using jdk-11.0.2 in IntelliJ IDEA. It requires 9,48 kb memory to keep source files. Project size is 1.43 mb with javadoc and input file. In order to run the program, additional input file is required. An input file named input.txt created in the specified format must be located in the project folder to perform the requested operation. For the programs to work properly, input file must be appropriate for the program.

2)METHOD

2.1)Class Diagrams



2.2) Use Case Diagrams



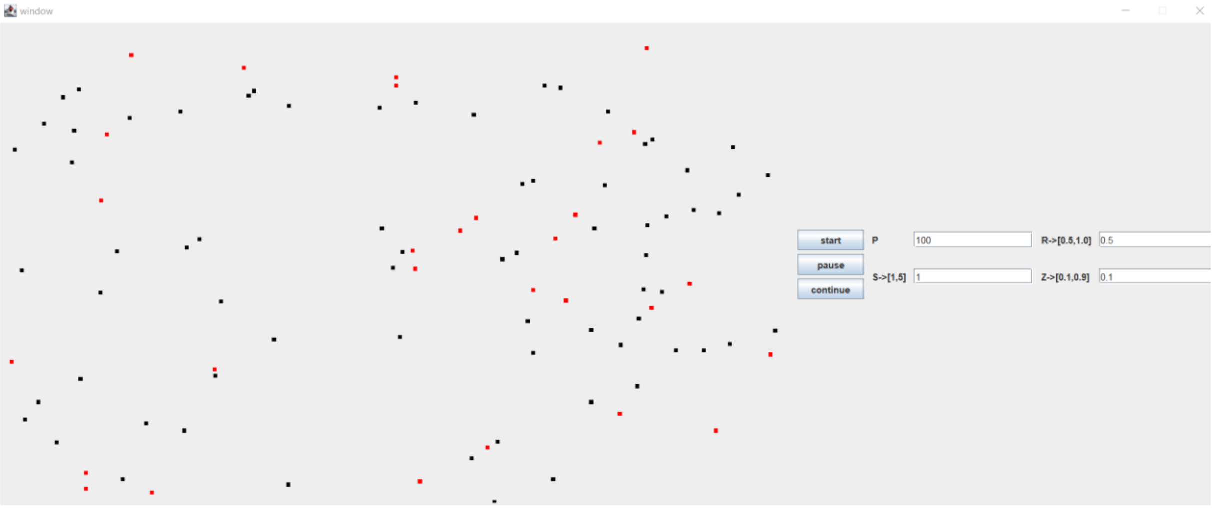
P

R->[0.5,1.0]

S->[1,5]

Z->[0.1,0.9]

First, we have a start screen and empty canvas. We have entered p value for the size of population and press enter. After that, enter an R value constant spreading factor and press enter. After that, enter a Z value for constant mortality rate and press enter. After that, enter an S value for speed.
If you enter the wrong value program exit with -1
After this step press start button.



P 100

R->[0.5,1.0] 0.5

S->[1,5] 1

Z->[0.1,0.9] 0.1

We create a population that a population with one in three people sick and one in five people with an M=1.0.This canvas repaint for every 50 milliseconds.

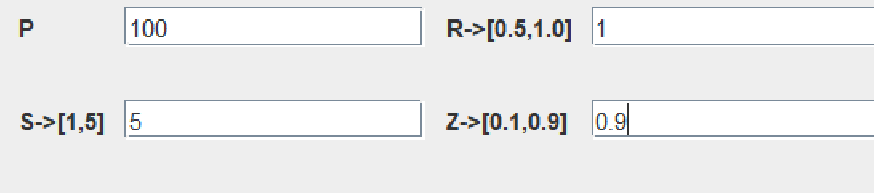
2.3) Problem Solution Approach

In this Project we have population and there is some interaction between elements of this population. I use Mediator design pattern for this interaction.
I have Mediator.java class for implementing interactions between People.java classes.
Another problem that we have to solve is creating a hospital system. And do its job
fully multythreaded. I use producer, consumer approach fort his problem. And synchronize
Them with using BlockingQueue.

The screen refreshes itself every 50 milliseconds, and people move S amounts of time each screen refresh.

So S can be 1 ,2,3,4 or 5.

3)TEST CASES

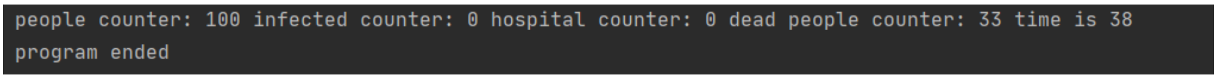


P 100

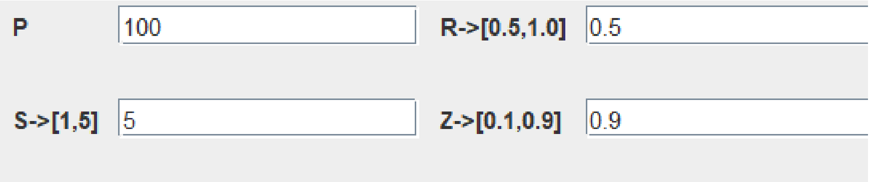
R->[0.5,1.0] 1

S->[1,5] 5

Z->[0.1,0.9] 0.9



```
people counter: 100 infected counter: 0 hospital counter: 0 dead people counter: 33 time is 38
program ended
```

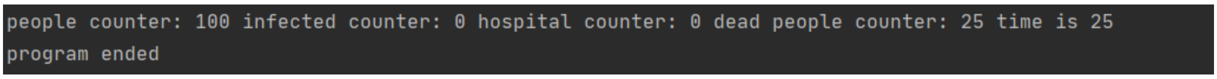


P 100

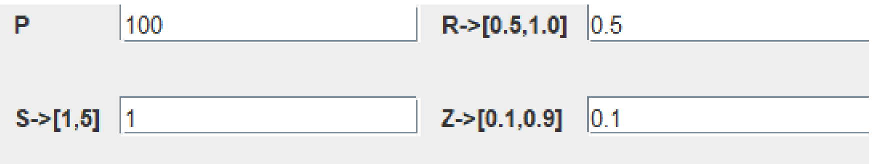
R->[0.5,1.0] 0.5

S->[1,5] 5

Z->[0.1,0.9] 0.9



```
people counter: 100 infected counter: 0 hospital counter: 0 dead people counter: 25 time is 25
program ended
```

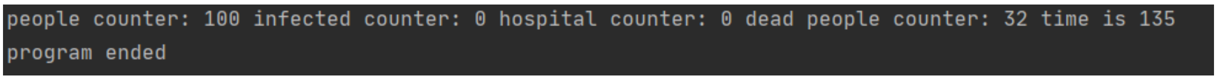


P 100

R->[0.5,1.0] 0.5

S->[1,5] 1

Z->[0.1,0.9] 0.1



```
people counter: 100 infected counter: 0 hospital counter: 0 dead people counter: 32 time is 135
program ended
```