

**INFO 6205**  
**Program Structures & Algorithms**  
**Fall 2020**  
**Final Project**  
**Team 6(Dayu Jia & Bowen Jiang)**

- **Introduction & Goal**

This is a project which simulates the spread of SARS-CoV-2.

There are two goals of the project:

At first, we want to compare some other diseases with COVID-19 to see how factors like  $k$  and  $R$  will affect the spread.

Secondly, this simulation can give us a visual way to understand what we can do to slow the spread and protect ourselves.

- **Preparation**

$R$  and  $k$  factors and the rate of spread.

$R_0$ , the basic reproductive number, is defined as the mean number of infections caused by an infected individual in a susceptible population. Most people will use  $R_0$  to evaluate the transmissibility of a disease. According to *Epidemiological and clinical features of the 2019 novel coronavirus outbreak in China* by Yang yang, we find that the  $R_0$  of COVID-19 is 3.77.

But after reading some articles like *Superspreading and the effect of individual variation on disease emergence* by J. O. Lloyd-Smith, we find that  $k$  factor should be considered too.  $K$  is the overdispersion parameter of a negative-binomial distribution.

This is just like the 80/20 rule. According to *Estimating the overdispersion in COVID-19 transmission using outbreak sizes outside China* by Akira Endo, we know that the  $k$  number of COVID-19 is .01, which means that 80% of secondary transmissions may have been caused by 10% of infectious individuals.

However, it's hard for us to implement  $R_0$  factors into the code.  $R_0$  is more like an observation than a factor in a function.

Therefore, we want to calculate the average rate of spread. Let's name it  $S$ .

In the first place, we need to find out how many people an individual will contact in a period of time. It's easy to understand that the average rate of spread equals to  $(R_0 * \text{initial number of individuals}) / \text{total number of contacted people}$ . But It's difficult for us to calculate the number. But after reading some data that our government published, we think 80 would be a proper number for us.

$$S = \frac{3.77}{80} = 4.71\%$$

And we, in the same way, can figure out that: S of SARS is 3.25%, S of measles is 18.75%.

Secondly, we should consider the k factor. As we said before, “80% of secondary transmissions may have been caused by 10% of infectious individuals”. Therefore:

$$S_1 = (S * 80\%) / 10\% = 37.68\%$$

$$S_2 = (S * 20\%) / 90\% = 1.05\%$$

We can get a chart:

	COVID-19	SARS	Measles
S of super	37.68%	16.25%	68.18%
S of normal	1.05%	0.77%	4.81%

## • Implementation

**How people get infected:** If an individual contact a patient in a dangerous distance, we will mark him/her as Contacted or Contacted by Super(which means super spreaders). Then we use random number to confirm whether this individual will be infected. If this individual is infected, then we use another random number to confirm whether this individual would be a super spreader.

```
double stdShadowTime = func.stdGaussian( sigma: 5, configuration.SHADOW_TIME);

if (MyPanel.worldTime - confirmedTime > stdShadowTime
    && (state == State.CONTACTED || state == State.CONTACTEDBYSUPER)){
    beInfected();
}

public void beInfected() {
    if (state == State.CONTACTED) {
        if (Math.random() <= (configuration.NORMAL_RATE * (1 - 0.15 * configuration.MASK_WEAR_RATE))) {
            infection();
        }
    }
    if (state == State.CONTACTEDBYSUPER) {
        if (Math.random() <= (configuration.SUPER_RATE * (1 - 0.15 * configuration.MASK_WEAR_RATE))) {
            infection();
        }
    }
}

public void infection() {
    if (Math.random() < configuration.K) {
        state = State.SUPERSPREADER;
    } else {
        state = State.CONFIRMED;
    }
    confirmedTime = MyPanel.worldTime;
}
```

Dayu Jia(NUID: 001569081) Bowen Jiang(NUID: :001582174)

**How people move:** In the beginning, all people are distributed evenly in the city. Then we make them move in a normal distribution way.

```
private PeoplePool(){
    City city = new City( centerX: 400, centerY: 400);

    int size = configuration.POPULATION_SIZE;

    for (int i = 0; i < size; i++) {

        int row = (int)Math.sqrt(size);
        int disX = configuration.CITY_WIDTH / (row + 1);
        int disY = configuration.CITY_HEIGHT / (row + 2);

        int x = disX * (i % row + 1);
        int y = disY * (i / row + 1);

        peopleList.add(new People(city, x, y));
    }
}
```

```
if (moveTarget == null || moveTarget.isArrived()){
    double targetX = func.stdGaussian(targetSig, targetXU);
    double targetY = func.stdGaussian(targetSig, targetYU);

    moveTarget = new MoveTarget((int) targetX, (int) targetY);
}

int dX = moveTarget.getX() - getX();
int dY = moveTarget.getY() - getY();

double length = Math.sqrt(Math.pow(dX, 2) + Math.pow(dY, 2));

if(length < 1){
    moveTarget.setArrived(true);
    return;
}

int udX = (int) (dX / length);
if (udX == 0 && dX != 0){
    if (dX > 0){
        udX = 1;
    }else
        udX = -1;
}

int udY = (int) (dY / length);
if (udY == 0 && dY != 0){
    if (dY > 0)
        udY = 1;
    else udY = -1;
}

if (moveTarget.getX() > configuration.CITY_WIDTH || moveTarget.getX() < 0){
    dX = dY = 0;

    if (udX > 0){
        udX = -udX;
    }
}

if (moveTarget.getY() > configuration.CITY_HEIGHT || moveTarget.getY() < 0){
    dX = dY = 0;

    if (udY > 0){
        udY = - udY;
    }
}

moveTo(dX, dY);
```

**How to find close contacts:** We already know how to find and mark people contacted by a patient, we only need to make sure that these suspected people is clean if they don't become patients in enough time.

```
for (People person : people) {
    if (person.getState() < State.CONFIRMED) {
        continue;
    }
    if (distance(person) < SAFE_DIST){
        if (person.state == State.CONFIRMED) {
            state = State.CONTACTED;
            confirmedTime = MyPanel.worldTime;
        }
        if (person.state == State.SUPERSPREADER) {
            state = State.CONTACTEDBYSUPER;
            confirmedTime = MyPanel.worldTime;

            break;
        }
    }
}

if (MyPanel.worldTime - confirmedTime > configuration.ISO_TIME
    && (state == State.CONTACTED || state == State.CONTACTEDBYSUPER)) {
    state = State.NORMAL;
}
```

**How people get isolated:** At first, we have a class named Isolation which gives us a space to isolate people(In reality, people may be isolated in hospitals or their own home, but it doesn't affect our simulation). Then we will put people in this space.

```
if (state >= State.CONTACTED && MyPanel.worldTime - confirmedTime >= configuration.ROOM_RECEIVE_TIME) {  
    if (Math.random() < configuration.ISO_RATE) {  
        Room room = Isolation.getInstance().pickRoom();  
  
        useRoom = room;  
        state = State.FREEZE;  
        setX(room.getX());  
        setY(room.getY());  
        room.setEmpty(false);  
  
        return;  
    }  
}  
  
if (state >= State.CONFIRMED && MyPanel.worldTime - confirmedTime >= 10) {  
    Room room = Isolation.getInstance().pickRoom();  
  
    useRoom = room;  
    state = State.FREEZE;  
    setX(room.getX());  
    setY(room.getY());  
    room.setEmpty(false);  
}
```

**Other factors, like wearing masks, safe distance and people's mobility:**

According to Dr Ghebreyesus from WHO, we find that wearing mask can reduce the chance of being infected by 15%.

```
public void beInfected() {  
    if (state == State.CONTACTED) {  
        if (Math.random() <= (configuration.NORMAL_RATE * (1 - 0.15 * configuration.MASK_WEAR_RATE))) {  
            infection();  
        }  
    }  
    if (state == State.CONTACTEDBYSUPER) {  
        if (Math.random() <= (configuration.SUPER_RATE * (1 - 0.15 * configuration.MASK_WEAR_RATE))) {  
            infection();  
        }  
    }  
}  
  
public void infection() {  
    if (Math.random() < configuration.K) {  
        state = State.SUPERSPREADER;  
    } else {  
        state = State.CONFIRMED;  
    }  
    confirmedTime = MyPanel.worldTime;  
}
```

According to Lijuan Li, an academican of the Chinese Academy of Engineering, the safe distance between people would be like 2 meters. So we set the safe distance in the configuration and use it in our code about contact.



```
for (People person : people) {
    if (person.getState() < State.CONFIRMED){
        continue;
    }
    if (distance(person, < SAFE_DIST){
        if (person.state == State.CONFIRMED) {
            state = State.CONTACTED;
            confirmedTime = MyPanel.worldTime;
        }
        if (person.state == State.SUPERSPREADER) {
            state = State.CONTACTEDBYSUPER;
            confirmedTime = MyPanel.worldTime;
        }

        break;
    }
}
```

As for the people's mobility, we use people's willing to move to confirm this. If a healthy individual doesn't want to move, then he would do nothing when action() happens.

```
public boolean wantMove(){

    return func.stdGaussian(sig, configuration.WANTMOVE) > 0;
}
```

- **Unit tests**

We have ten java files. Configuration and MyPanel cannot be tested. And Point, Isolation, Room, MoveTarget, and City all almost have only constructive functions and generators. Func file only has some random functions.

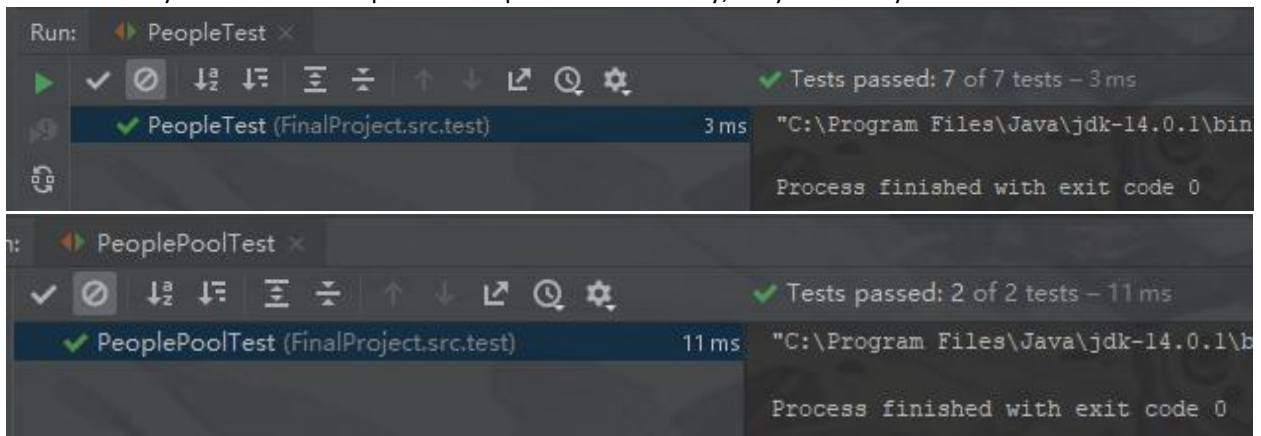
```
public class func {
    private static final Random r = new Random();

    /*
     * stdX = (X - U) / sigma
     * X = sigma * stdX + u;
     */
    public static double stdGaussian(double sigma, double u){
        double X = r.nextGaussian();

        return sigma * X + u;
    }
}
```



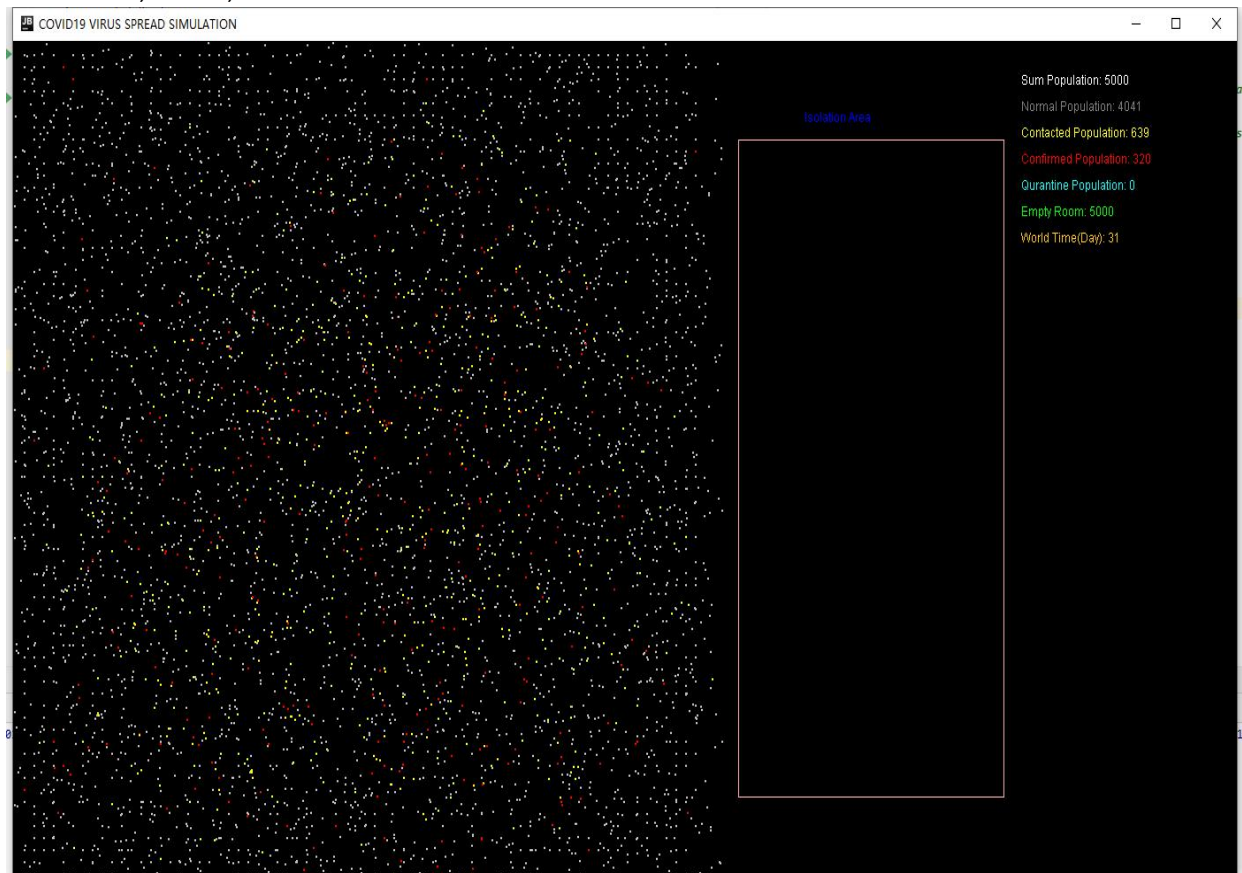
So we can only unit test our People and PeoplePool file. Luckily, they work very well.



## • Outputs & Analysis

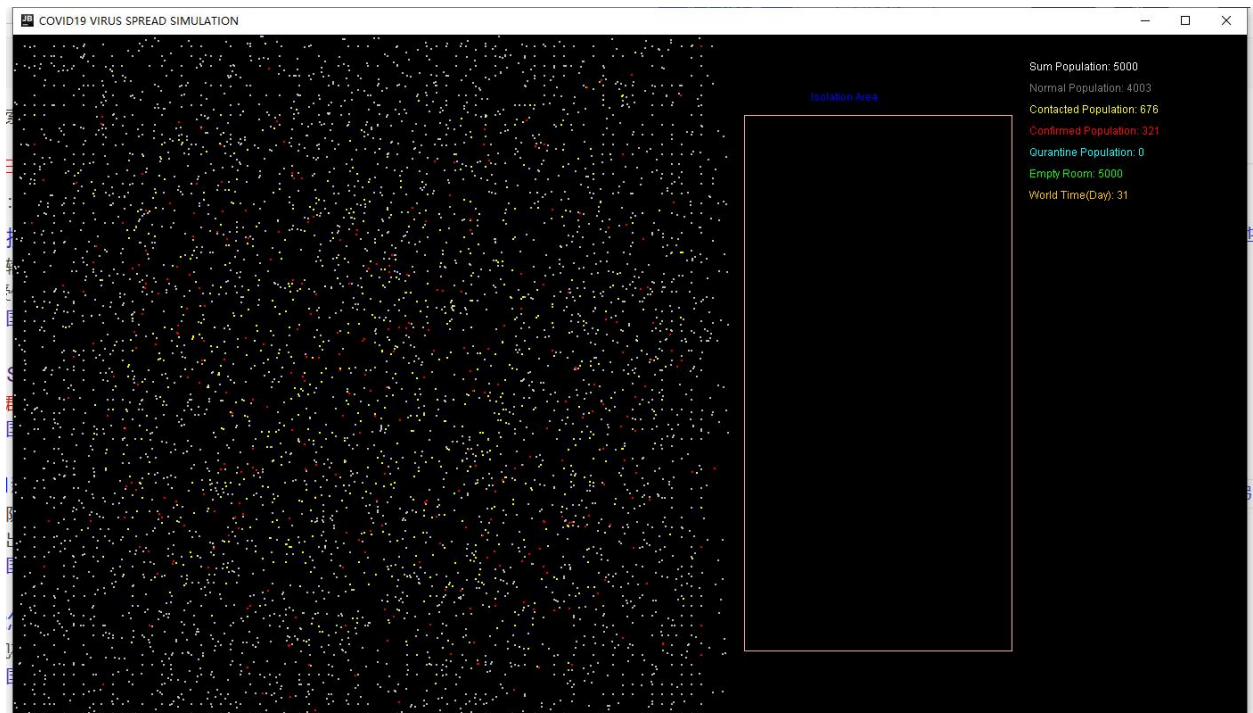
**Comparison of the three different kinds of viruses** without wearing masks and isolation.

SARS-CoV-2,  $k = 0.1$ ,  $R = 3.77$

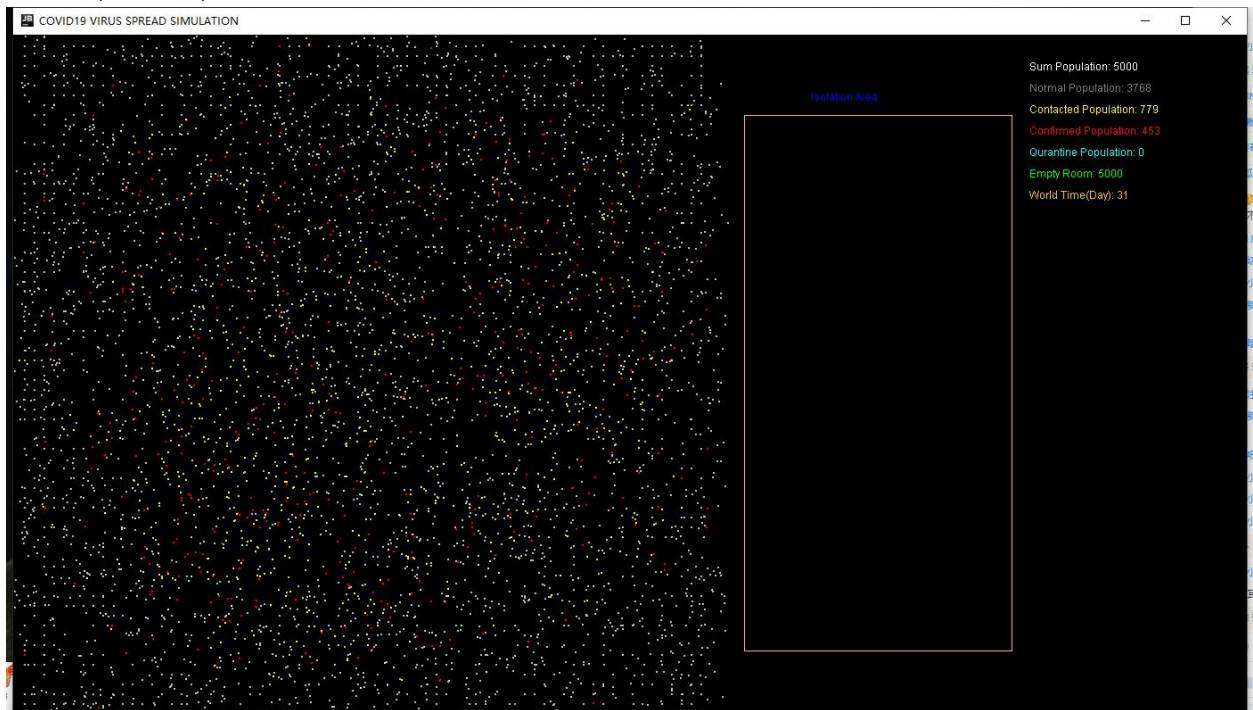


SARS 2003,  $k = 0.16$ ,  $R = 2.6$

Dayu Jia(NUID: 001569081) Bowen Jiang(NUID: :001582174)



Measles,  $k = 0.22$ ,  $R = 15$

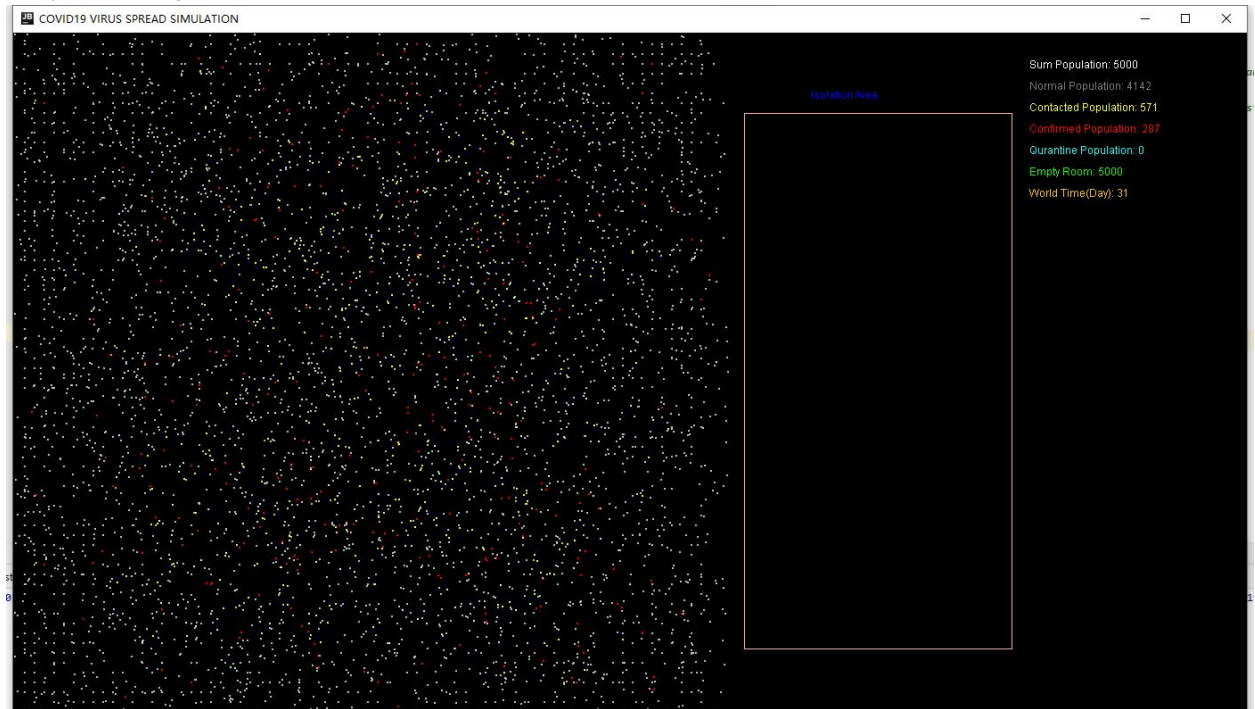


Under the condition without wearing mask and having any isolation. After 1 month distribution, considering  $k$  values of three different viruses, the distribution rate of measles is lower than SARS and COVID-19. The  $k$  values of COVID-19 and SARS are similar, so there is no significant difference in the distribution performance.

Dayu Jia(NUID: 001569081) Bowen Jiang(NUID: :001582174)

Because of R factor, it's obvious that measles have a lot more patients than other virus, and COVID-19 and SARS are similar.

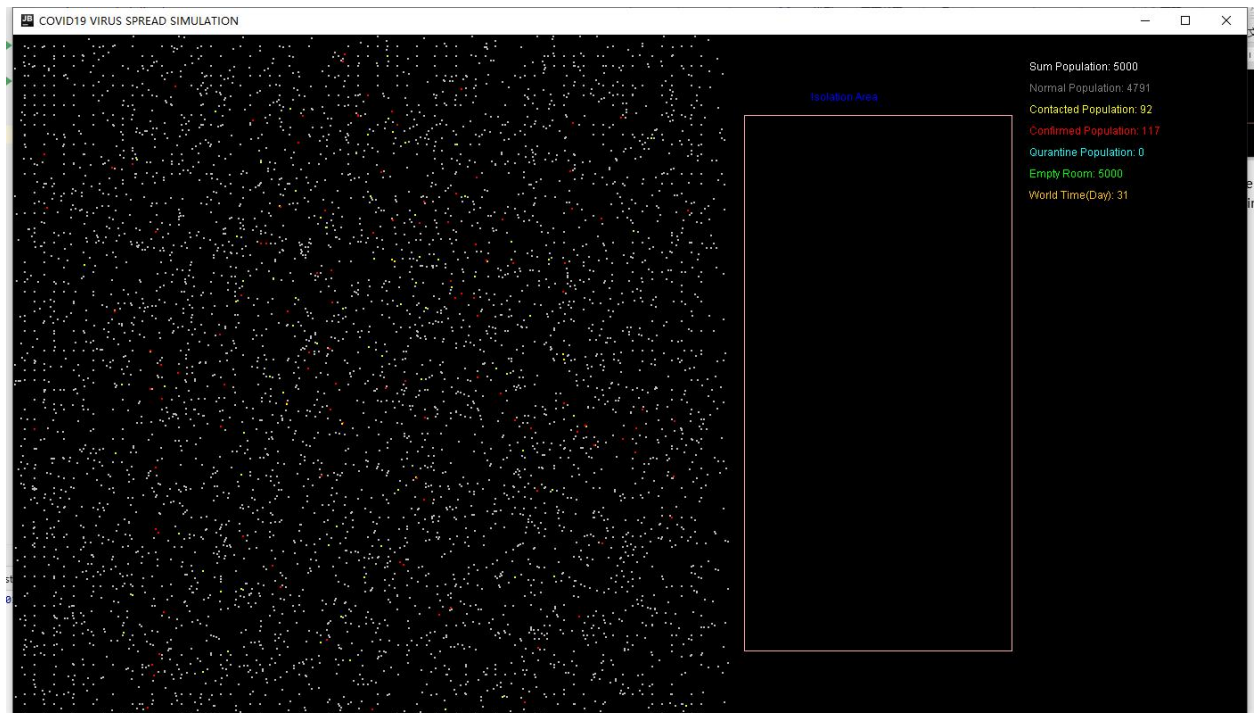
### People wearing masks



Wearing mask without setting any isolation area, it can be found that masks can reduce the probability of virus infection to a certain extent, however, wearing masks is of limited help to the overall infection.

### Reducing population mobility



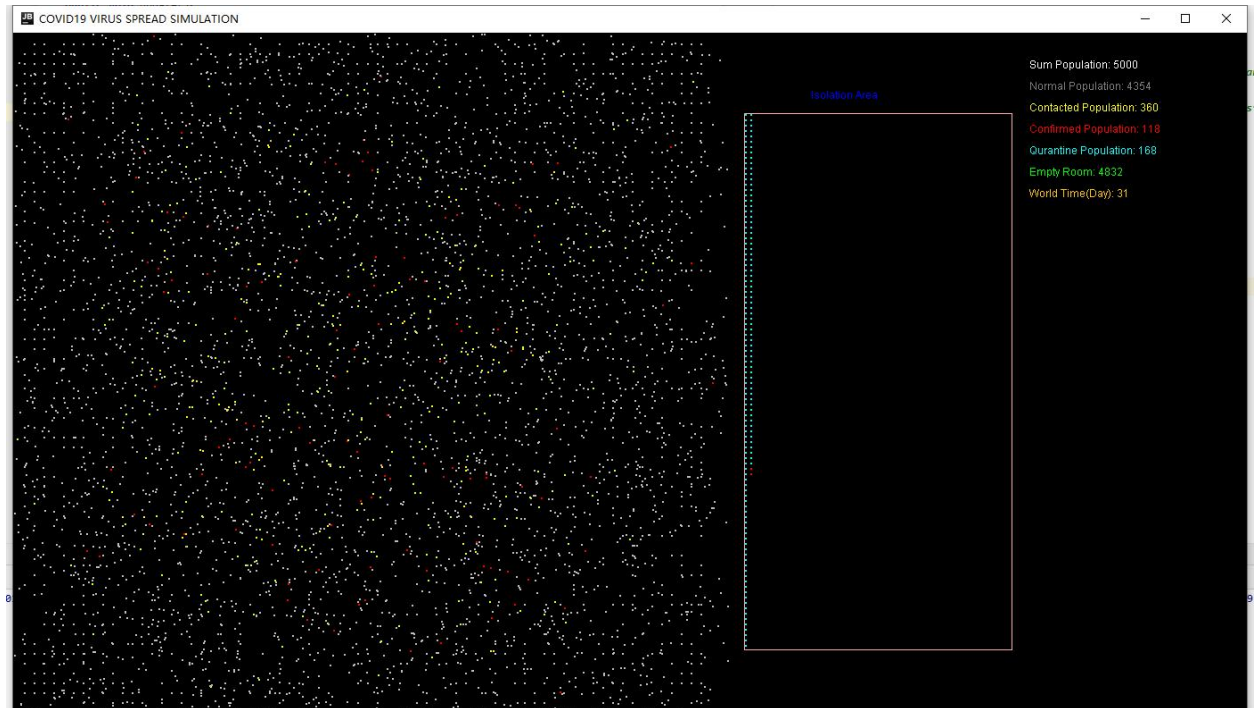


Reducing population mobility can effectively reduce the transmission of the virus, like 2/3.

### Set Isolation area

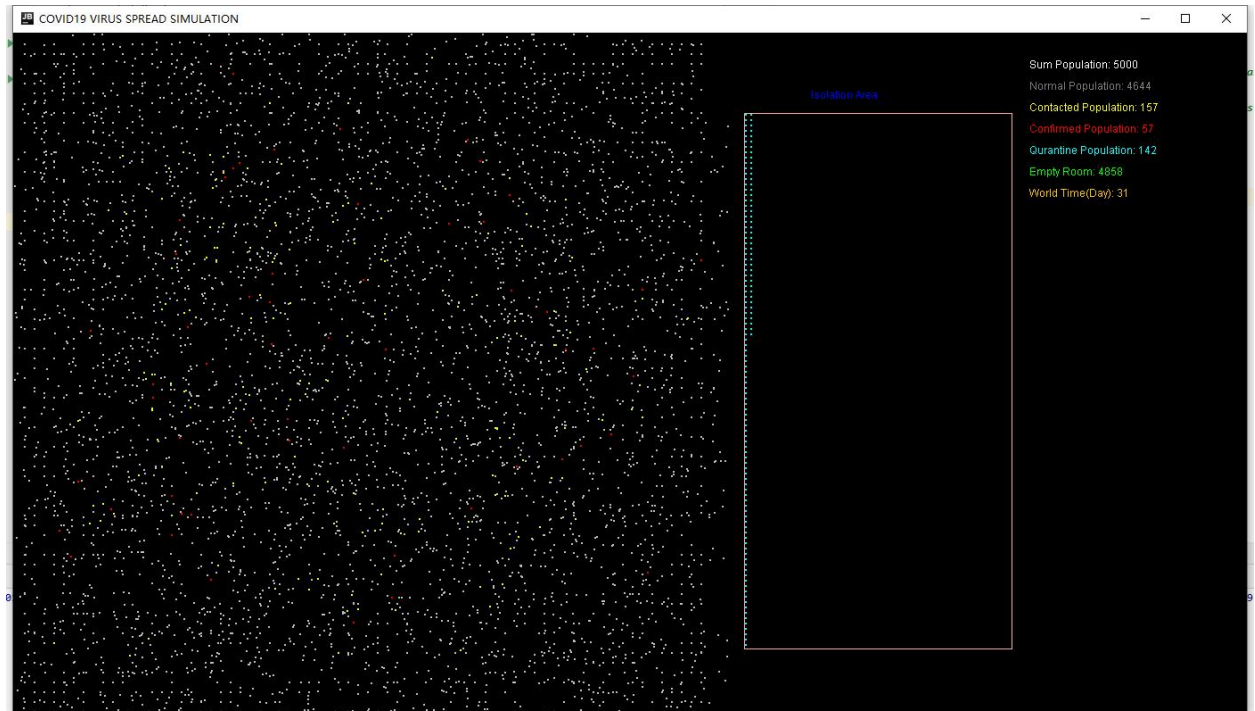
We need to set a isolation rate, we cannot isolate every contact considering the real situation.

Set isolation rate = 0.1



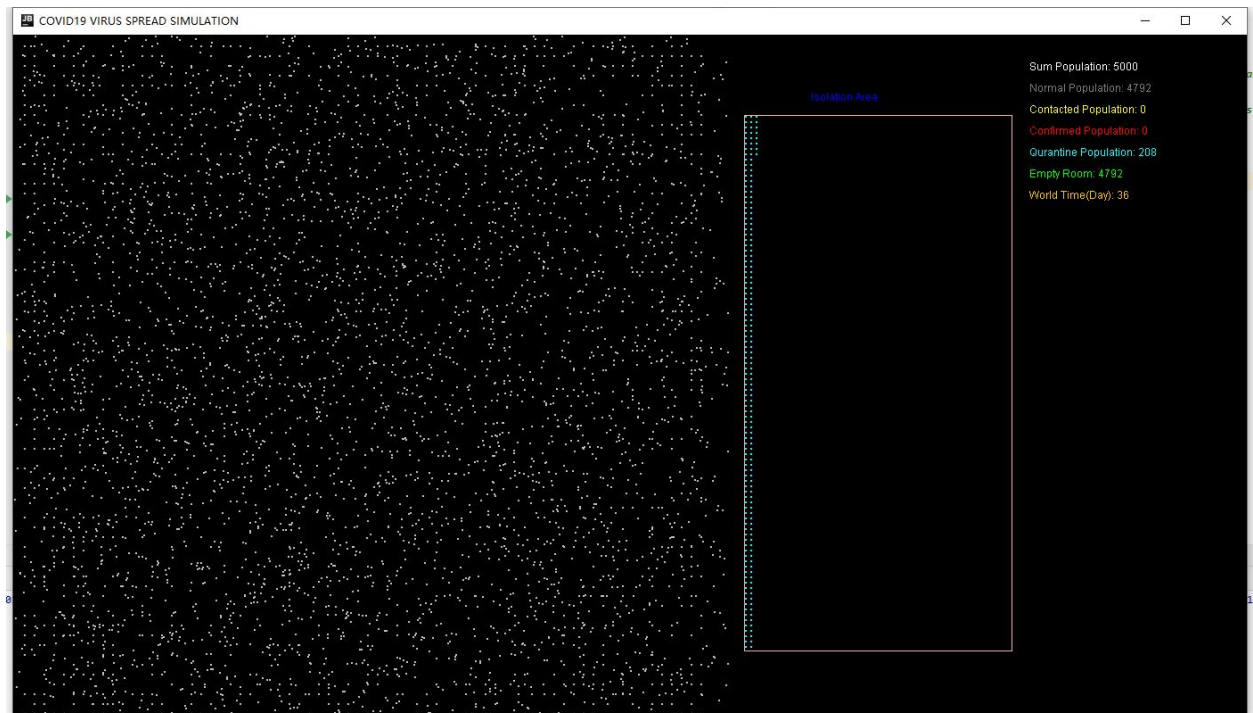
Even if we just set the rate as 0.1, which means only 10% contacts will be isolated, it still has a good result. 10% isolation can reduce the chance of infection just like everyone wearing a mask. What if we set the rate higher?

Set isolation rate = 0.5



It works better than we expected: there are only 7 new patients, although there are still some suspected individuals, in a month. So we decide to set the rate up to 0.9 to see what would happen.

Set isolation rate = 0.9



If we isolate patients and suspected individuals with a high rate, which is feasible for a government, close contacts can be quickly controlled without expanding their ability to transmit. Neither super-spreaders nor ordinary patients can spread quickly. We can almost end the spread in a month!

At first we don't think this is a correct result because it's so incredible. But then we think of some cases in our country, China. China had several patients discovered in the past few months, but none of these cases caused bad results. The reason may be just the isolation because the government always found the contacts quickly and then tested and isolated them until they are safe and healthy. Therefore, setting isolation zones is a very effective way to combat the epidemic.

## • Conclusion

1. As we can see, when we do nothing to the simulation, it will spread quickly from many groups in the city. And there are some groups bigger than others because of super spreaders, or the  $k$  factor. The simulation of SARS looks similar to COVID-19. As for measles, it spread a lot more quickly. But its bigger groups, comparing to COVID-19, are not that obvious.
2. Wearing masks is a good way to reduce the chance of getting infected.
3. Reducing mobility is an effective way to slow the spread. So let's try our best to stay home.
4. Setting isolation areas is the best way to stop, not just slow, the spread. For a citizen like us, this is more or less like the former way: staying home. If we can stay home as best as we can, and isolate ourselves in time when situation is bad. I'm sure the virus will pass more quickly than we think.