**Final project: Plague spreading on isolated island**

Name: Jingying Hu

Course No.: CBE 5790

Date: 2018.11.29

**Introduction**

I have used deterministic model and stochastic model to simulate the population change during flue spreading in a small and isolated island. Inspired by my past work and a video game called “Plague Inc”, I try to animate a similar scenario where people get infected with flue by contacting the sick. And topics such as random walk, object-oriented will be included in the program.

Plague Inc. is a game for player to pick a kind of flue and virus to spread across the world. Players are able to facilitate the spreading of sickness by inducing mutation of the virus or assign symptoms like cough. At the same time, the vaccine is also under research. The player has to make more and more people die of the sickness before they get vaccinated. The more people died, higher grade the player will get. Although the games seem to be wicked, but it’s very interesting to figure out how contagious sickness spreads between people by assigning different powerful features to the sickness as well as utilizing the environmental condition in the game.

My model is different from the simulation in the game, because the observational units in my model is residents on the island while those in the game are locations around the world. Therefore, my units are keeping moving unless they are dead. But it will also tell how deadly the contagious plague is. More importantly, we can more directly observe how free riders benefit from others being vaccinated through my animation.

**Algorithm**

1. Distribute healthy residents and sick residents in a fixed grid. Healthy residents are divided into people who are willing to vaccinated and free riders. The number of each kind of people is designable and their initial locations are random. In case residents may get out of the grid, the boundary condition of the grid is periodic.
2. People get contacted when they reach certain distance between each other. But the probability that they get infected after this contact is designable. People who are newly infected is then deleted from the healthy people list during the cycle. And at the same time, newly infected residents will be assigned to the sick people list.
3. People may die of the plague or recovered from the plague. If a sick resident is dead, then the resident will be removed from the sick people list. If a sick resident is recovered, then the resident will be added to the immune people list.
4. Doctors will also be distributed to inject vaccine for people who are willing to be injected. Doctors’ initial positions are randomly assigned. The number of doctors and when will doctors arrive are designable. Healthy residents who get vaccinated will be added to the immune people list.
5. The plague will stop either when all the sick people is dead or all the healthy people is dead.
6. Different group of residents will be assigned with different colors during animation: The healthy is green; The sick is red; The immune is blue; The doctor is yellow.

**Results & discussion**

First of all, I am interested in finding out several features a deadly plague should have from my program. To do the test, I set the scenario as 3200 residents living in a 1000\*1000 grid.

The simulation result clearly shows that if the plague doesn’t have a higher probability in death than in recovery, the plague is likely to be conquered by vaccine as shown in the Fig. 1. Also, sick residents who are dead can not work as the contagious media to spread the plague. Therefore, as expected, when sick residents have a higher probability to die of the plague, the plague can only spread in a small scale. As shown in Fig. 2, many residents who are willing to be vaccinated have been successfully vaccinated while only a small number of residents are in the sick status.

Then I decreased all the probabilities of death, recovery and vaccination to 0.001. The plague has a large scale of spreading as shown in the top left of Fig.3. Therefore, the probabilities of death, recovery and vaccination are negatively related to the spreading of plague. The higher these probabilities, the smaller scale where the plague is.

At last, in order to see whether free riders can well benefit from surrounding residents who get vaccinated. I increased the probability of being vaccinated. As the plague spread from top to bottom in the Fig. 4, I observed that immune residents are clustered. But the plague keeps spreading. I think free riders do have a small probability of being infected, but other people gets vaccinated has nothing to do with the existence of the sick residents. As long as the sick residents keep moving, free riders still get infected.

During this project, I found my biggest challenge is that how to find an interesting problem worth being animated. Since all my python experience comes from my coursework, to originally create a problem and animate is a big difficulty to me. Problems too easy are not worthy of being animated. While for problems too complex, I am not sure whether I can finish it in time.

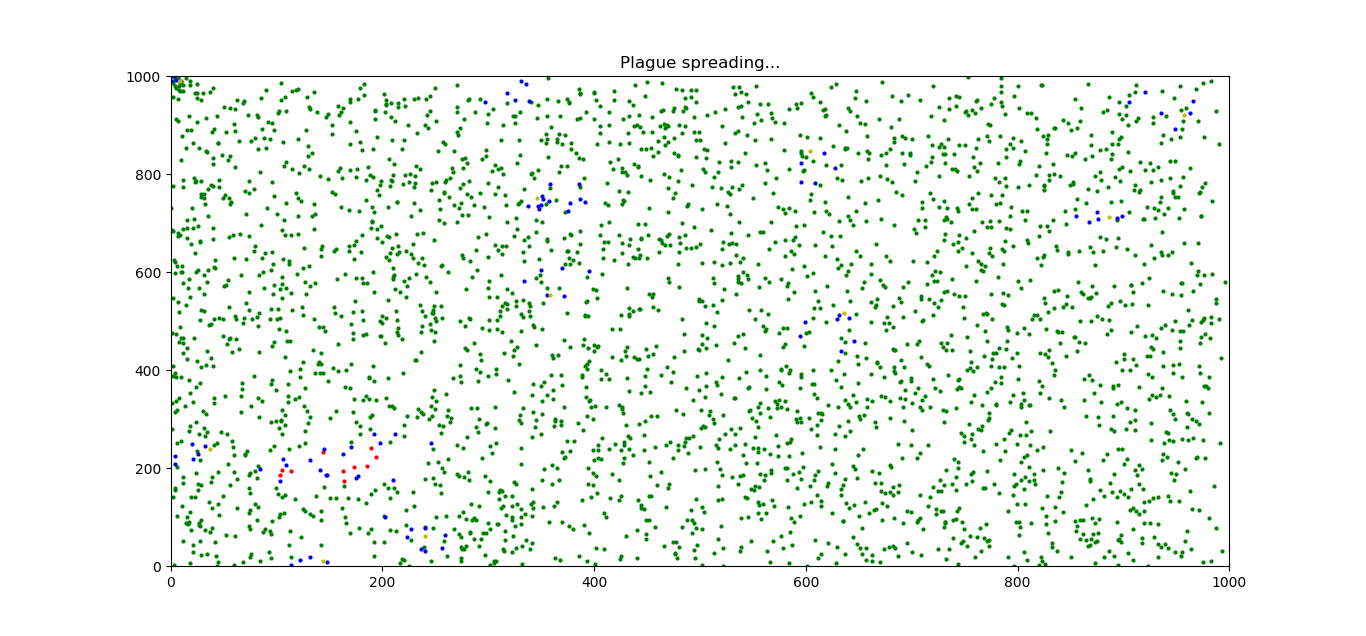
****

Fig. 1 simulation result after calling the function as myfinal (n = 3200, N = 1000, s = 1, d = 10, doctor = 10, f = 50, tDoc = 10, ngen = 100000, Pinfect = 0.1, Precover = 0.01, Pdead = 0.01, Pimmune = 0.01)

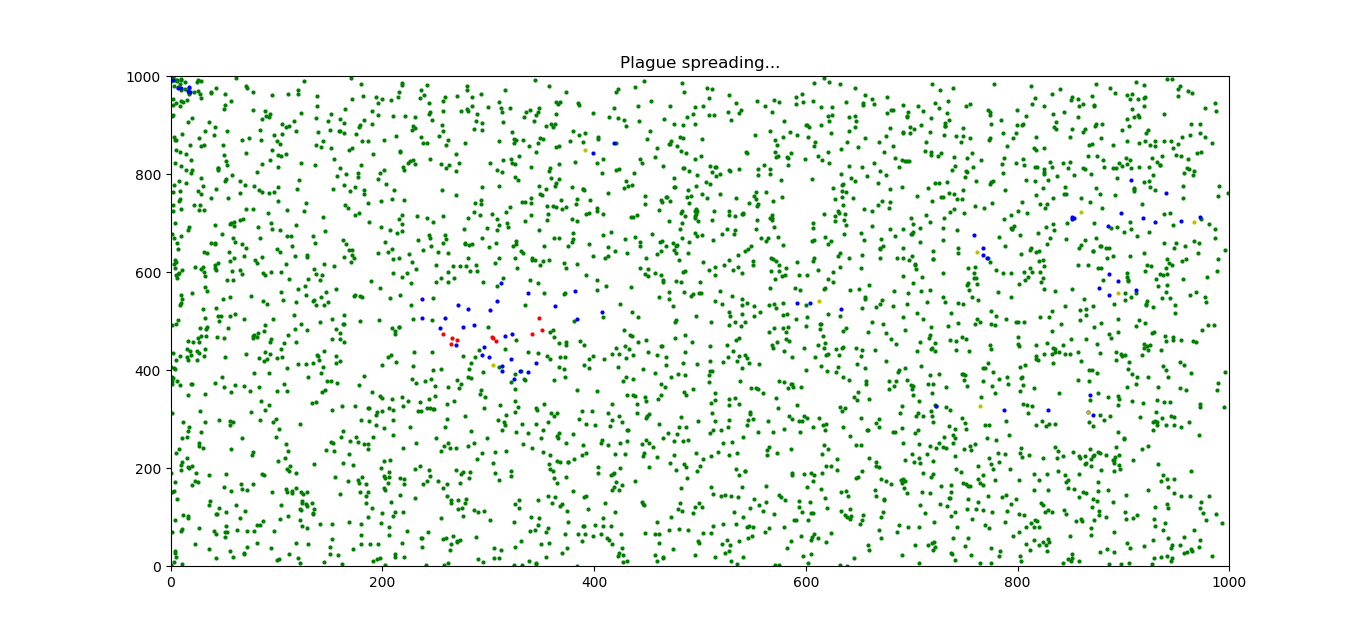
****

Fig. 2 simulation result after calling the function as myfinal (n = 3200, N = 1000, s = 1, d = 10, doctor = 10, f = 50, tDoc = 10, ngen = 100000, Pinfect = 0.1, Precover = 0.01, Pdead = 0.1, Pimmune = 0.01)

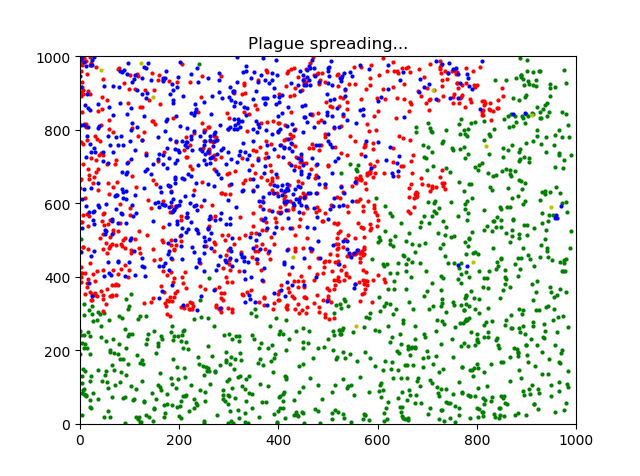
****

Fig. 3 simulation result after calling the function as myfinal (n = 3200, N = 1000, s = 1, d = 20, doctor = 10, f = 50, tDoc = 10, ngen = 100000, Pinfect = 0.1, Precover = 0.001, Pdead = 00.1, Pimmune = 0.001)

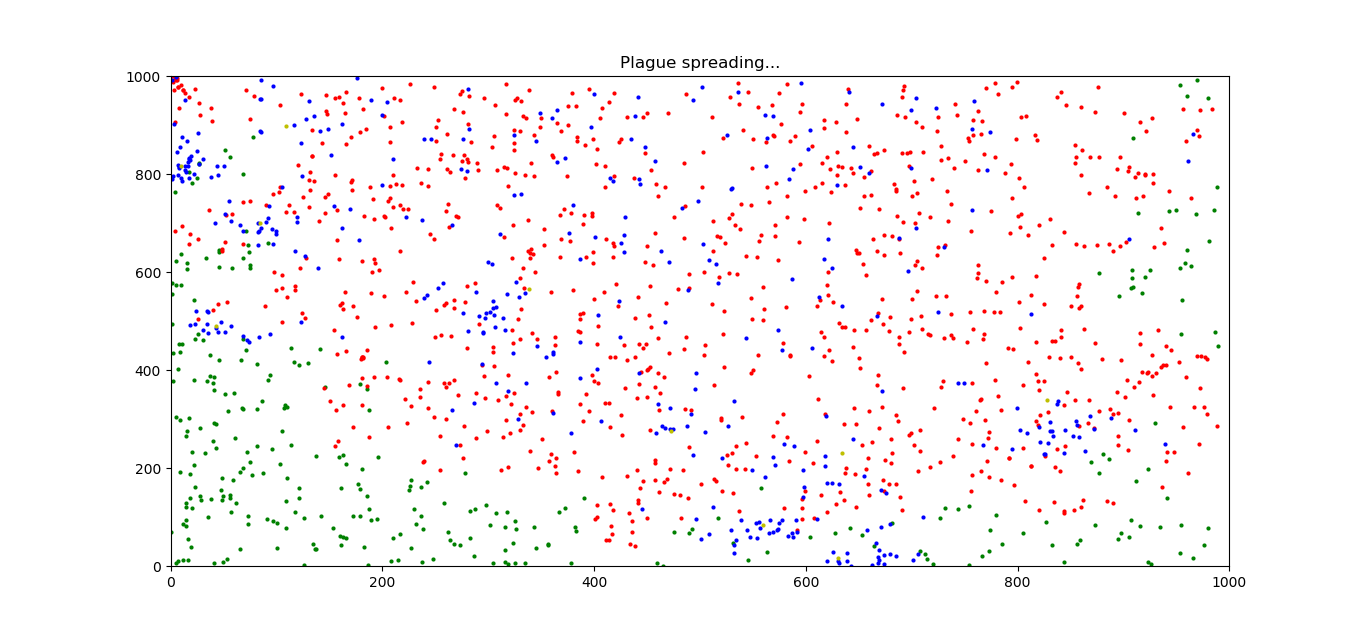
****

Fig. 4 simulation result after calling the function as myfinal (n = 3200, N = 1000, s = 1, d = 20, doctor = 10, f = 50, tDoc = 10, ngen = 100000, Pinfect = 0.1, Precover = 0.0001, Pdead = 00.1, Pimmune = 0.1)

**References:**

**People who helped me:**

Bryan Hobocienski: Thanks to Bryan who helped me figure out the idea and possibility of the project and he suggested me to use “sklearn” package to find out residents who are going to be infected next. He did great contributions to my project.

James F. Rathman: Thanks Dr. Rathman for teaching 5790 modeling and simulation. Referring to the “drunkard\_class” example he published on carmen, I was able to do an object-oriented program with thousands of moving objects.

**Packages called:**

numpy

matplotlib.pyplot

sklearn.neighbors

**Videos viewed:**

A Random Walk & Monte Carlo Simulation || Python Tutorial || Learn Python Programming