Cellular Automaton: Epidemic Simulation

In my epidemic simulation using cellular automatons I considered each cell an individual who has the potential of becoming infected. The initialization of the simulation has the following parameters: width, height, radius, death time, neighborhood model, probability of infection when within the radius of an infected cell, the decay rate of the probability of infection depending on how far away the cell is, and the decay rate of the probability for a cell to get removed depending on how long it has been infected.

There are also three possible methods to populate the simulation space with cellular automatons. All three of the methods have an initial infection population which is a percentage of the total population. This parameter may not be exactly the percentage of the population that is infected, as each cell has a chance to be infected equal to the initial infected population parameter. The other parameters of the three possible methods to populate the simulation space are different ways to model what the population's resistance. The first method just uses a constant resistance chance for every cell. The second method uses a Gaussian distribution of resistances skewed towards a target value with a specified standard deviation. This is meant to model the "healthy" population having a higher chance of removal while the unhealthy population, such as the young and old, will have less of a chance of surviving the infection. The third method gives the population random resistance values based on a uniform distribution.

For this simulation, there is no latency period. Once a cell is infected it will stay infected. This simulation does however model the cells gaining immunity to the infection by removing their chance of getting infected. If the cell has not gained immunity within the specified amount of time it will die. The infection has a chance to spread to any cells within the specified radius of affect. The neighborhood model is a parameter and can be either a Von Neumann neighborhood or Moore neighborhood with a specified radius. Since each infected cell within the neighborhood has the potential to infect the cell at the center of the neighborhood, it becomes more likely the cell contracts the infection as more neighbor cells become infected.

The simulation is terminated when the simulation space reaches a stable state. This is when the number of infections, deaths, and removed cells does not change after a certain number of time steps. The number of time steps to wait should be at least the number of time steps it can take for a cell to die after becoming infected. However, this value may need to be increased if probability of infection is extremely low.

My goal in developing this simulation was create a highly customizable simulation. I believe I accomplished this as there are many parameters with which one can initialize the simulation. Also, there are several options in which a user can populate the simulation space. Some of the more difficult issues I faced were creating a skewed Gaussian distribution and allowing for using either a Von Neumann or Moore neighborhood model. The implementation to create a skewed Gaussian distribution is still sensitive to its parameters. The comments in the code explain the nuances of the method.

Allowing for multiple neighborhood models as well as the size of the neighborhood required me to understand how each neighborhood grew as its radius increased. In summary a Von Neumann neighborhood population size can be calculated using the following summation:

$$\sum_{1}^{r} r * 4, \ \textit{where } r = \textit{the radius of the neighborhood}$$

Similarly a Moore neighborhood population can be calculated with this summation:

$$\sum_{1}^{r} r * 8, \ \textit{where } r = \textit{the radius of the neighborhood}$$

On top of understanding how each neighborhood's population grows, I needed to program two methods to traverse either neighborhood model's radii.

To visualize the simulation I used an output .csv file from the java program containing the infection, death, and removal counts of each day. A python script was then used to plot the data on a line graph. The parameters used for the graphed simulation were the following: 50x50 simulation space, 3 cell radius for a Moore neighborhood model, 5 days of infection before a cell dies, 5% chance of a cell to become infected if it is in contact with another infect cell, 50% infection chance decay rate, and a 95% removal chance decay rate. The simulation space was populated using a constant resistance chance of 25% for the entire population and an initial infected population of 1% the simulation space population.