

User Guide for simSpread.py

Overview:

To initiate a simulation that produces the effects of a disease being transmitted between individuals, simSpread.py was developed and edited.

Various techniques were established to process the simulation program in which few functions were formed to display the movement of individuals across the map that is output via a grid. In addition, to verify if people traveling/moving inside the map have been infected with the disease, further behavioral approaches were implemented to check if the person has been infected with a disease, or if they are immune to the disease; whether did they die because of that disease or did they recover from that illness?

To add, the model was also encoded to allow users to input parameters to automate the simulation outcome, and if the user does not input any parameters for the simulation to run, the simulation would be manually performed by using the variable values that are pre-defined within the code created.

Not only this, but the code generated could also move individuals across long distances using airports, from where they may have been infected with the disease.

Nonetheless, people were also granted the freedom to travel as they wanted around the map, in whatever direction they want, whether it being going towards north, south, east, west, north-east, north-west, south-east, or south-west.

How to process the simulation:

There are 2 ways to run the simulation:

(1) Directly without any parameter sweeps input – To run the simulation directly, the user would just call the python file i.e. **python3 simSpread.py**, they would then be notified that since no parameters have been input, pre-defined values from the code will be used to run the simulation. **HOWEVER**, the user would need to choose a neighborhood for proceeding further with the program. Lastly, after all timesteps are executed, the statistics of the simulation is shown at the end.

(2) Using parameter sweeps – To output results of the simulation using parameters, the user would call the file in the following way:

**Python3 simSpread.py <PopulationValue> <InfectedNumber>
<neighbourhoodName> <Airports>**

Example: Python3 simSpread.py 200 20 moore 10

Explanation: This would define the values to the arguments automatically, and then produce a simulation result based on what the user has input; The population in the above example given would be of 200 people, wherein 20 amongst those would be infected, and the individuals would move along the map in directions north, south, east, west, north-east, north-west, south-east, or south-west (**MOORE**), and finally, amongst 10 airports. Lastly, after all timesteps are executed, the statistics of the simulation is shown at the end.

Interpretation of the code:

The code runs as soon as the operator calls `simSpread.py`; it is the individual's choice if they want to use any parameters or continue with default values. The simulation is run 10 times, wherein each time, different results may be produced using the additional behaviors of `becameInfected`, `notInfected`, `Death`, and `Recovery`.

As mentioned earlier, using the parameter, the handler can set the total population and the no. of people that are infected in the simulation as per as they like, otherwise it is set to default values of 100 (population) and 5 (infected people).

The additional features that I had added into this simulation was **“Death”** and **“Recovery”**, wherein for the method of death, if the random number generated is less than the probability being calculated in the **infect** method, then we would conclude that the chances of survival for that person is minimal and count them as dead. However, for the recovery method, if the random number produced is less than the probability calculated, we would add that person into the list of those whom have **recovered** instead of the list of dead, and further deduct a number for the same person in the list of infected people.

Death Example:

- If `random.random < probability` , consider person as dead.

Recovered Example:

- If `random.random < probability`, the person is added into the recovered list, and removed from the infected list.