Project 8.4 Simulating an epidemic

In the SIR population model that we simulated in Chapter 4, we only modeled the number of individuals who were affected by the virus; we could not model the behavior of individuals interacting with each other. In contrast, in an ***agent-based simulation***, we model the behavior of individuals, called ***agents***, as they interact with each other over time.

In this project, you will create an agent-based simulation of the spread of a virus in a two-dimensional grid using the same techniques we used to simulate the Game of Life in Section 8.2.

In the simulation, each cell will be in one of three states:

- `SUSCEPTIBLE`: not infected but susceptible to becoming infected
- `INFECTED`: currently infected and capable of infecting others
- `RECOVERED`: was infected and done infecting neighbors; no longer susceptible

Just as in the Game of Life, in each generation (now one day) of the simulation, a cell will interact with its eight neighbors (and only its neighbors). In this case, an infected cell will randomly infect some its neighbors. The number it infects depends on the reproduction number R of the virus, defined to be the number of individuals that each infected individual infects on average.

The simulation

Before you continue, download the program that accompanies this project on the book website. The functions at the beginning of the program implement drawing the grid as the simulation progresses. The next two functions, `emptyGrid` and `initialize`, set up the grid at the beginning of the simulation. You do not need to edit any of these functions.

The `infectNeighborhood` function, similar to the `neighborhood` function in the Game of Life, looks at a cell's eight neighbors (and only its neighbors), and randomly infects some of them, depending on the reproduction number `R`. The function returns the number of newly infected individuals. When a cell is infected, it should turn black.

Question 8.4.1 *Complete the* `infectNeighborhood` *function by translating the commented pseudocode into Python.*

The `infection` and `main` functions are the main drivers of the simulation. The `infection` function is very similar to the `life` function in the Game of Life. In each day of the simulation, it iterates over every cell in the grid. For each cell, if it is infected, the `infectNeighborhood` function is called to infect the cell's neighbors. Once a cell infects its neighbors, it is marked as `RECOVERED` so that it no longer infects others in the future. It is also colored gray.

Once you complete the `infectNeighborhood` function, you should be able to run the program and see the simulation in action.

Question 8.4.2 *Run the simulation many times with different values of* `R` *to test it and see how the reproduction number affects the spread of the virus. Keep in mind that, since this is a random simulation, you will need to run it many times to get a sense of the average/typical behavior. What do you notice?*

Question 8.4.3 *Next, augment the* `infection` *function so that it creates and returns two lists. The first list will contain the number of new infections each day. The second list will contain the cumulative total number of infections up through each day. The* `main` *function is already set up to plot these two lists so that you can visualize the progress of the infection.*

Question 8.4.4 *Run the simulation again with different values of* `R`*. What do you learn from the plots?*

Finally, let's experiment with what happens when a fraction of the population quarantines (or "shelters in place") at home. We can simulate this by initializing some number of cells as `RECOVERED` at the beginning of the simulation. As a result, the virus will essentially "pass over" these cells, and they will not contribute to the infection's spread.

Question 8.4.5 *Simulate "sheltering in place" by adding another parameter named* `fractionSheltered` *to the* `infection` *function. In the function, randomly initialize this fraction of the cells to* `RECOVERED` *before the simulation begins. Color each of these cells blue to visualize their locations.*

Question 8.4.6 *Run the simulation again multiple times with increasing values of* `fractionSheltered` *and* `R = 2.5`*. What do you notice?*