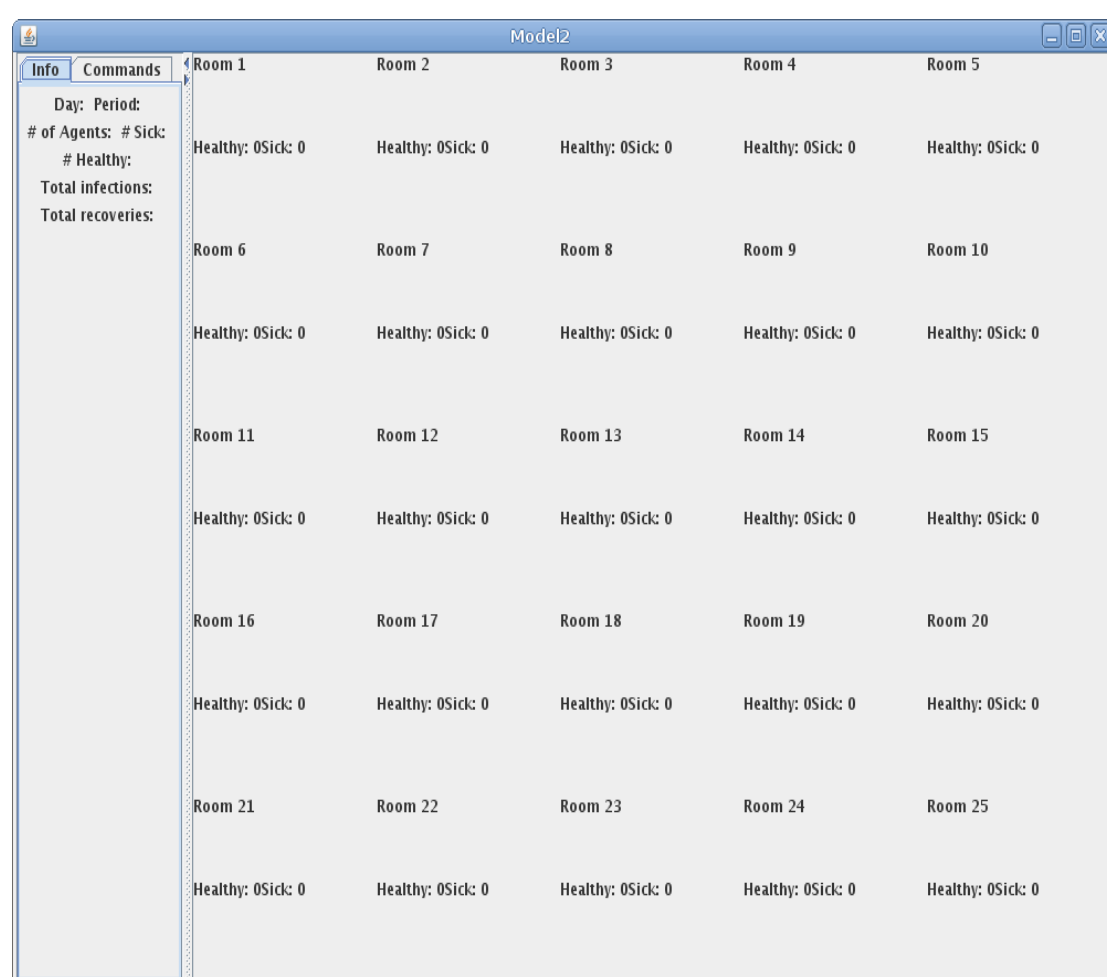


Simulation of the Spread of a Virus Using Agent Based Modeling

Matt Wade

Abstract

My goal is to make an agent based modeling simulation that shows the spread of a cold through a school. It will start with an amount of infected students and healthy students received as inputs and will show how much the virus spreads or possibly recedes over time. The program will answer the question as to how quickly and fully different types of sicknesses will be able to spread through the population of a school once introduced. This will show how likely it is for a disease to be spread by a set amount of sick people coming to school with the sickness. This will show if the danger of infecting others is actually a valid excuse not to come to school or if you should come to school unless you actually don't feel like you will be able to do work.



Info	Commands	Room 1	Room 2	Room 3	Room 4	Room 5
Day: Period		Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0
# of Agents: # Sick:						
# Healthy:						
Total infections:						
Total recoveries:						
		Room 6	Room 7	Room 8	Room 9	Room 10
		Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0
		Room 11	Room 12	Room 13	Room 14	Room 15
		Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0
		Room 16	Room 17	Room 18	Room 19	Room 20
		Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0
		Room 21	Room 22	Room 23	Room 24	Room 25
		Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0	Healthy: 0Sick: 0

Procedures

My program has three main classes. An Agent class which defines what values an Agent will store and how to construct it. A class that creates the GUI. And the most complicated of the three, the Model class which is where all of the calculations occur. In the Model class there is an arraylist containing all of the Agents in the simulation. With these Agents the class has to define a step() function which moves the simulation forward. This function has to update the locations of all of the agents, check to see if any of them get infected, and check to see if any recover from being sick. First it goes through the list of Agents and moves them all to the next location in their schedule. In order to check for any new infections it goes through the entire list of Agents finding each sick Agent. Whenever it finds a sick Agent it finds any healthy Agents in the same location and checks a randomly generated number against the sick Agent's infectiousness value. If the random number is lower then the healthy Agent is switched to sick and the method continues on through the rest of the list. To check if any Agents recover from sickness it goes through the list checking the recoverytime value and if it equals zero the Agent is switched to healthy. The GUI class contains another important function, the ability to graph data gathered through the simulation. In the Model class I added new arraylists to retain information over time for all of the main variables (number of sick agents, number of healthy agents, infections per step, etc.). Whenever the graphing method is called the Model class gets information from two drop down menus as to which arraylists are going to be the x and y axis variables and sends them to the graph class. The graph class then takes these variables and goes through the arraylist graphing a scatterplot of the data.

Expected Results

My program will answer the question as to how quickly and fully different types of sicknesses will be able to spread through the population of a school once introduced.

This will show how likely it is for a disease to be spread by a set amount of sick people coming to school with the sickness. This will show if the danger of infecting others is actually a valid excuse not to come to school or if you should come to school unless you actually don't feel like you will be able to do work.

At the moment my program is pretty much in its final state in terms of the actual simulation. All of the methods relating to the simulation, such as step(), checkinfection(), or checkrecovery(), are all completed and working as they should. This means that the results I am getting right now are pretty much the same results I will have when the whole project is completed except for any analysis tools that I plan to add to the program, such as a picture showing the locations of all of the Agents, and a graph that shows the number of sick and healthy Agents over time. As of now though, it will give you data on the number of sick and healthy agents, the total number of infections and recoveries, the number of steps taken, and the locations of all the agents.

