



Creating “Contagion”

A virus is infecting your world. As people move randomly and come face to face with a sick person, they too become sick. Learn more about how illnesses can spread through this simulation.

Created by: Susan Miller, University of Colorado, School of Education

This curricula has been designed as part of the Scalable Simulations Design project.
It was created using ideas from and portions of prior work completed by
Fred Gluck.

This material is based upon work supported by the National Science Foundation under Grant No. DRL-1312129 and CNS-1138526. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Vocabulary/Definitions

- Agent Attribute**.....an assigned feature of an agent (such as scent)
- Algorithm**a set of instructions designed to perform a specific task.
- Brackets**method of setting information apart using “[“ and “]”
- Broadcast**A way for agents to communicate with other agents that are not adjacent to them - agents broadcast (or send out) a signal
- Collision**.....an event wherein two agents interact when next to each other.
- Decrement**to decrease by one
- Diffusion**the process in which an attribute moves from areas of strong concentration to weak concentration
- Increment**to increase by one
- Hill Climbing**a specific form of searching/seeking technique, or algorithm, by which the seeking/searching agent uses information (agent attribute) embedded in the floor.
- Method**.....a set of rules to follow in a specific situation
- Parentheses**method of setting information apart using “(“ and “)”
- Polling**.....the process of contacting and communicating with each agent
- Propagated**the spreading of an agent attribute
- Randomly**to occur in non-systematic ways
- Rule Order**the order in which rules are placed for each agent
- Simulation Property**.....an attribute (value) accessible by all agents.

Student Handout 1A:

Part 1 – Basic Contagion

You will be modeling or simulating an epidemic. We will begin with a very simple model and adjust it to make it more accurate.

Tasks:

1. Create a new project called Contagion
2. Create agents for the simulation. You will need the following agents:
 - Ground
 - Person (two depictions, healthy and sick)
3. Create a basic worksheet that is covered in ground, with many healthy people and one sick person.
4. Program the agents using a method to make decisions:
 - Every 0.1 seconds DECIDE what happens next
 - If you are healthy and next to a sick person, determine if you get ill using the method TEST_GET_SICK.
 - Create a rule so that when a healthy person is next to a sick person, he SOMETIMES gets sick. Consider what percent of the time that should be.
 - If you are healthy and not next to a sick person, you should just move randomly on the ground.
 - If you are sick, you should move randomly on the ground.
5. Once your simulation is programmed, run it to see what happens. Work with a partner to answer these questions:
 - What happens when the percent is changed to zero? What would that mean in real life?
 - What happens when the percent is changed to 100. What would that mean in real life.
 - What percent is realistic for this simulation if it were the common cold? What percent is realistic for this simulation if it were Ebola?
 - What causes the simulation to stop? Why?

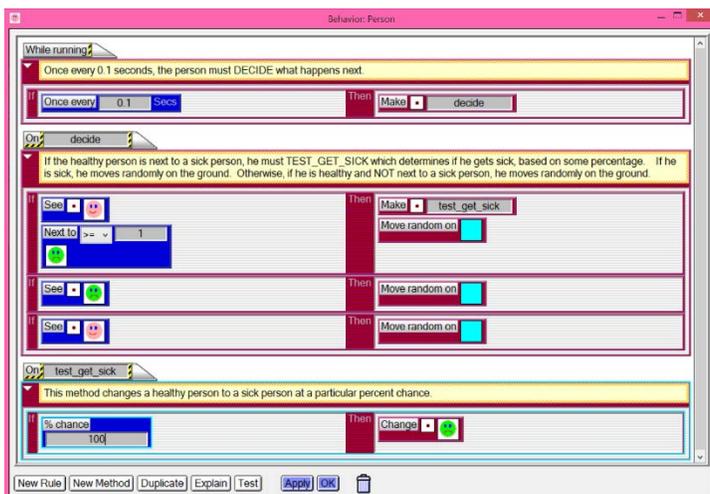
Student Handout 1B:

Part 1 – Basic Contagion

You will be modeling or simulating an epidemic. We will begin with a very simple model and adjust it to make it more accurate.

Tasks:

1. Create a new project called Contagion
2. Create agents for the simulation. You will need the following agents:
 - Ground
 - Person (two depictions, healthy and sick)
3. Create a basic worksheet that is covered in ground, with many healthy people and one sick person.
4. Program the agents using a method to make decisions:
 - Every 0.1 seconds DECIDE what happens next
 - If you are healthy and next to a sick person, determine if you get ill using the method TEST_GET_SICK.
 - Create a rule so that when a healthy person is next to a sick person, he SOMETIMES gets sick. Consider what percent of the time that should be.
 - If you are healthy and not next to a sick person, you should just move randomly on the ground.
 - If you are sick, you should move randomly on the ground.
5. Once your simulation is programmed, run it to see what happens. Work with a partner to answer these questions:



- What happens when the percent is changed to zero? What would that mean in real life?
- What happens when the percent is changed to 100. What would that mean in real life.
- What percent is realistic for this simulation if it were the common cold? What percent is realistic for this simulation if it were Ebola?
- What causes the simulation to stop? Why?

Student Handout 2:

Part 2 – Getting better!

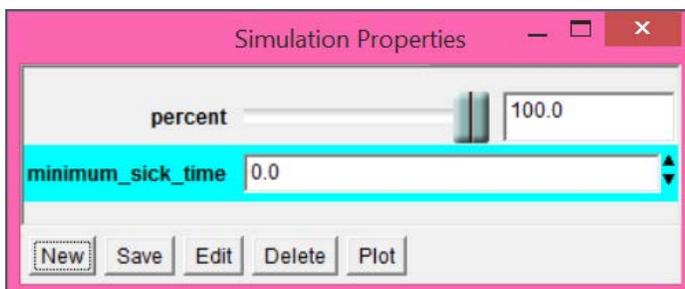
As you discussed with your class, you don't stay sick forever – even though you might feel miserable with a cold, you will get better, generally in a week or so. We are going to change the simulation to reflect that people do get better, and it takes 7 – 10 days.

We are going to designate that all sick people will take at least 7 days to get better (using a simulation property `MINIMUM_SICK_TIME`), and some might take a bit longer (as many as 10 days) using a randomly generated value of the variable (`EXTRA_SICK_TIME`). We will use a variable called as `SICK_CLOCK` to count down how long someone has been sick. We will use the `TEST_RECOVERED` method to check how many days are left of being sick.

Confused? Don't be – we'll take this step by step.

First, let's create some variables. Open the simulation properties (Tools>>Simulation Properties)

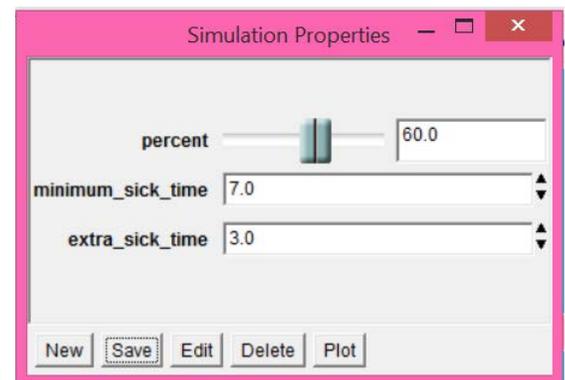
Click on **NEW**, and title it `minimum_sick_time`. Click **OK**. You should now see this:



Set this to 7 (to represent the 7 days) and click on **SAVE**.

Now, repeat this process and create the variable `extra_sick_time`. Set it to 3 (to represent the three extra days someone could be sick) and click **SAVE**.

Your simulation properties should now look like this:



To do the remaining programming, we will do it in steps...

Step 1: When a person gets sick, start the sick clock countdown

Step 2: If I am sick, check to see if I'm still sick each 'day'

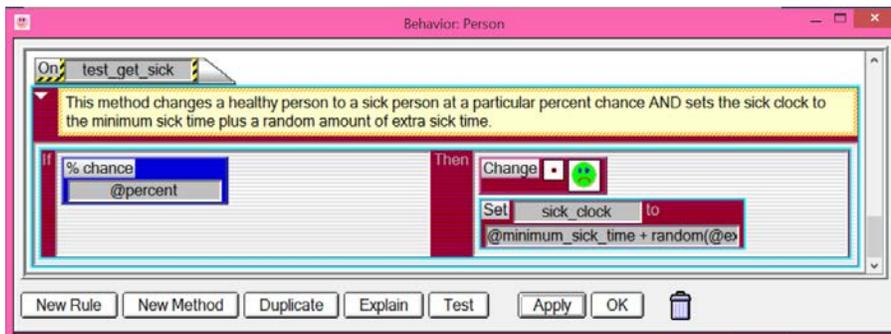
Contagion (Continued)

Step 3: Check to see if the sick clock is down to zero – if it is, change me back to a healthy person. (If not, count down another day - I stay sick).

Step 4: Create a sick clock starting time for your first sick person (already on your worksheet).

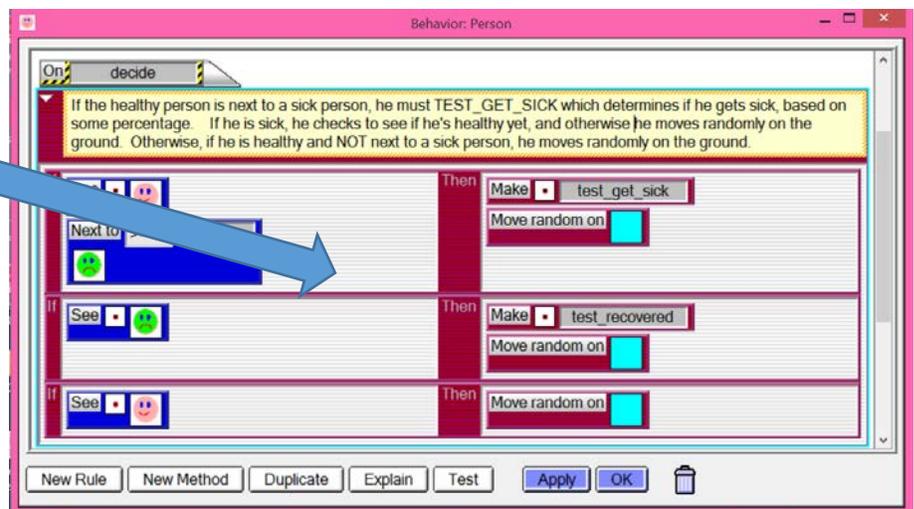
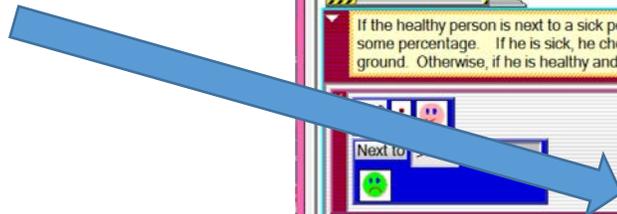
Let's get started!

Step 1: When I become sick, set the sick_clock to the minimum_sick_time plus a random amount of extra_sick_time.



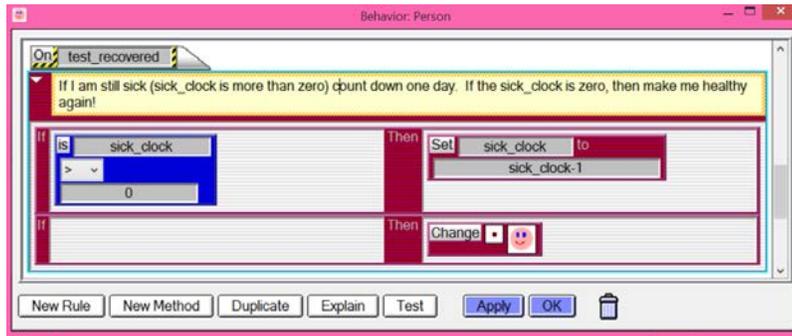
Set sick_clock to @minimum_sick_time + random(@extra_sick_time)

Step 2: Once I'm sick, check to see if I am still sick.



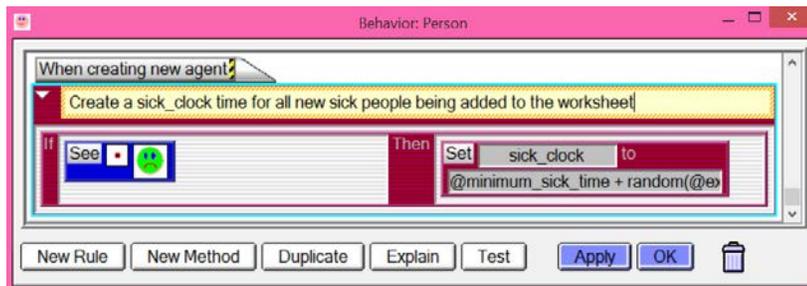
Contagion (Continued)

Step 3: Add the method `test_recovered` to count down the sick clock and then change me back to healthy if the clock is zero.

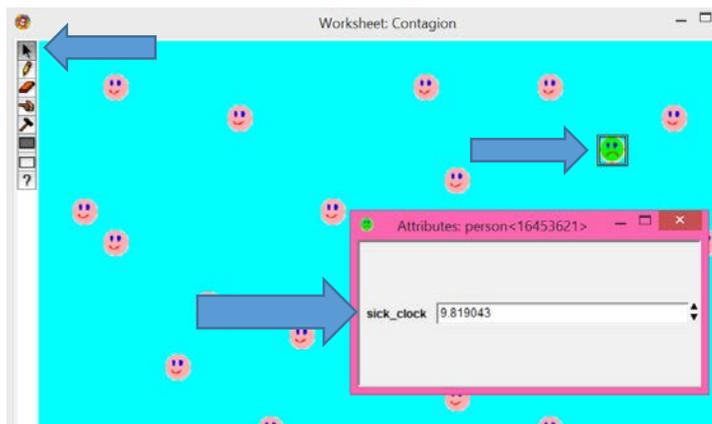


Step 4: Before you test this, you need to set the sick clock for your very first sick person. To do this, add this code to your person:

If I am sick, Then, **Set sick_clock to** `@minimum_sick_time + random(@extra_sick_time`



Delete your original sick person from your worksheet and add a new sick person. **SAVE YOUR WORKSHEET.** You can check to see the sick_clock value for your sick person by clicking on the arrow on the worksheet, and then clicking on the sick person (highlighting him). Then select (Tools>>Agent Attributes). It should show the sick clock for your agent.

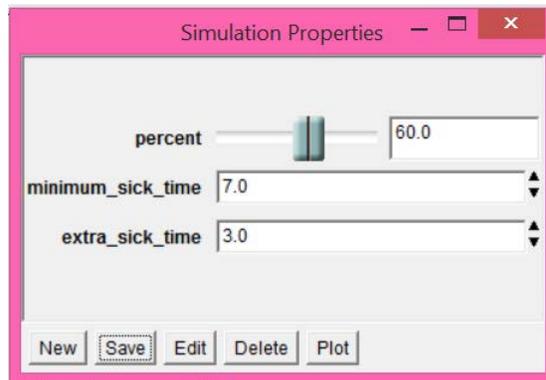


Contagion (Continued)

Run your simulation several times to see what happens. (Remember that it will operate as fast as your computer processor – you can use the slider at the bottom of the worksheet to slow it down, or you can add a ‘once every’ command to the first line of your code to slow it down.)

What happens in your simulation? Is it what you expected?

Open the simulation properties (Tools>>Simulation Properties). Try changing the variables to determine how they affect the simulation.



- What happens if you make the percent who get sick higher? Lower?
- What happens if the cold lasts longer?
- What happens if some people already have a weakened immune system and stay sick longer?
- What happens if there are more sick people in the simulation?
- What happens if there are more healthy people in the simulation?
- In what ways does this simulation reflect the real world? In what ways does the simulation not reflect what really happens?

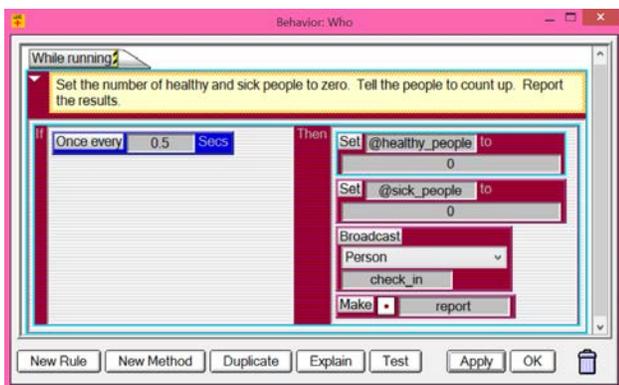
Student Handout 3:

Part 3 – Who’s sick – Who’s not? Gathering and Plotting the data

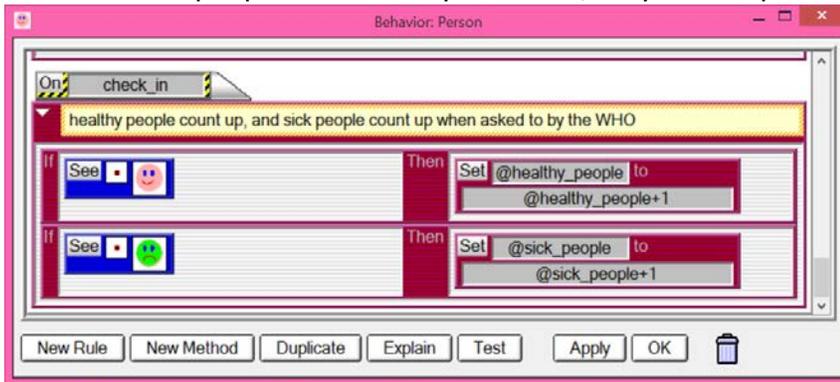
In this next lesson, you will learn to use simulation properties to track the number of healthy and sick people in their ‘world’ and then how to plot those properties.

Because epidemics can affect huge numbers of people all over the world, the United Nations created the World Health Organization to provide leadership on global health issues. This organization is responsible for the eradication of smallpox, and is working on eradicating several communicable diseases including HIV, malaria and tuberculosis. This organization is also very involved in trying to mitigate the current Ebola outbreaks. As part of its responsibilities, this organization surveys huge numbers of people every year to determine who has what illness. In our simulation, the WHO will become the controller (like the controller in Journey or PacMan) and will survey all of the people on the worksheet to determine who is healthy and who is sick.

Step 1: The WHO will set the number of healthy people and sick people to zero. The WHO will then survey the people by broadcasting a request for them to count themselves.

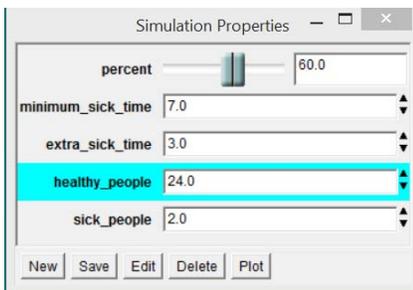


When the sick people are called by the WHO, they count up.

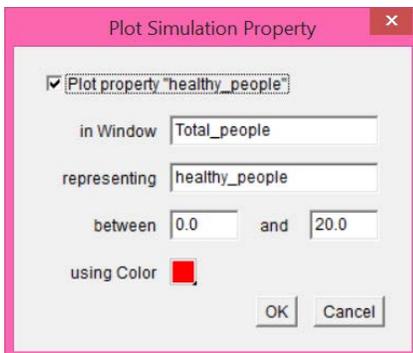


To watch the data, click on Tools>>Simulation properties. You will see the numbers of sick and healthy people change.

To plot the data, use the option to plot them in the Simulation Properties box.

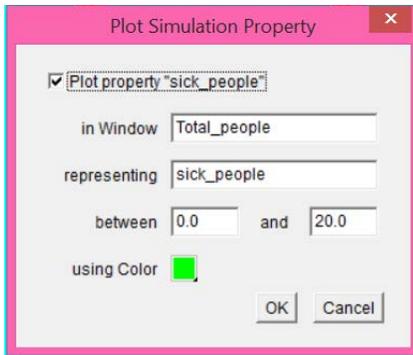


Highlight the property you wish to plot. Click on plot.



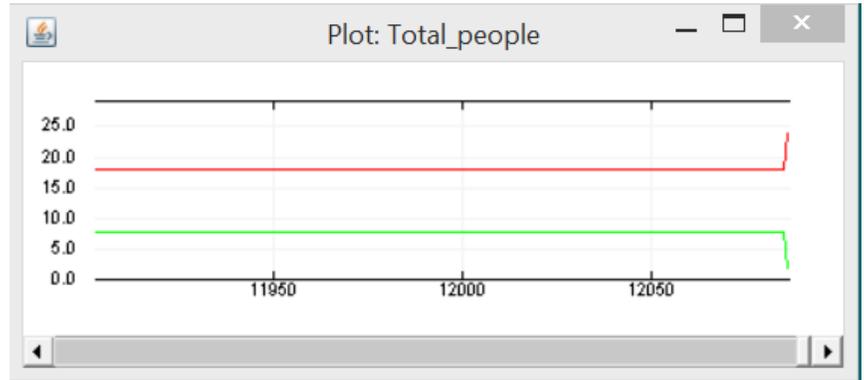
Create a window name (I chose Total_people) and change the min/max values to reasonable values for your worksheet.

Contagion (Continued)



Do the same for the second variable. Again, be sure to rename the window to match the first same so that they are plotted on the same graph. Change the min/max values to reasonable values for your worksheet. Change the color to a different color.

Run the simulation. Hint: the plot is sometimes hiding behind the worksheet. You may have to do some rearranging.



Student Handout:

Challenge 1.0: WHO Reporting

Before you start this challenge:

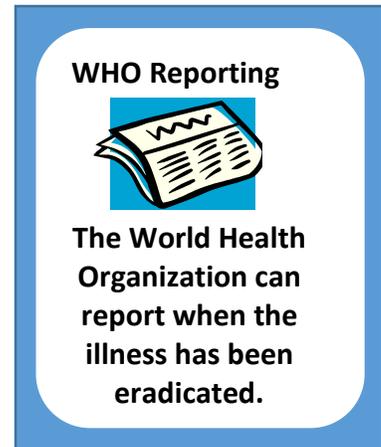
- You must have a complete basic Contagion simulation that enables people to get better after a certain amount of time.
- You have a World Health Organization (WHO) that polls people, asking them to count who is sick and who is healthy.

Design Challenge:

For serious diseases (like Ebola), the World Health organization is responsible for tracking the illness and reporting on the spread of that illness.

Add a method to notify the player when the simulation must end, either because:

- There are no sick people left
- There are no healthy people left.



Student Handout:

Challenge 2.0: Deaths

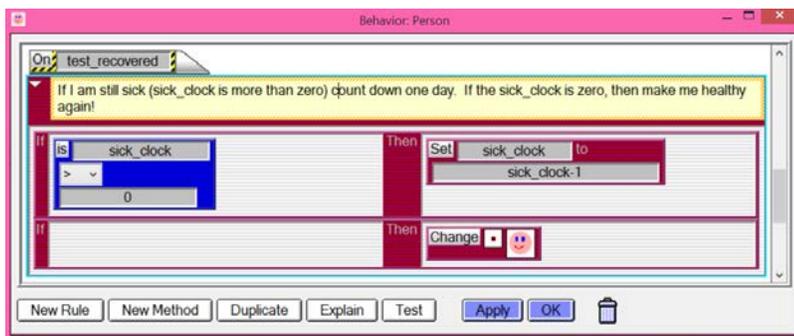
Before you start this challenge:

You must have a complete basic Contagion simulation that enables people to get better after a certain amount of time.

Design Challenge:

For serious diseases (like Ebola), not everyone gets better. Some people will die. Change your game to have some small percentage of people die. Be sure your percentage of people who die can be changed using a slider.

Hint, to do this, change the second rule in this code:



At the end of the sick time...EITHER, I get better, OR I die. So, create a method that checks to see if I live...At some percent of mortality, I change to a dead person, otherwise I change to a healthy person.

You should be seeing some new actions and a new depiction.

Something to consider...do dead people continue to move randomly?

Be sure to plot the deaths.



Student Handout:

Challenge 3.0: Quarantine

Before you start this challenge:

You must have a complete basic Contagion simulation that enables people to get better after a certain amount of time.

Design Challenge:

For serious diseases (like Ebola), not everyone gets better. To prevent a spread of the disease (so that it doesn't become an epidemic), sometimes people are quarantined.

Create a hospital area where sick people go. They remain in quarantine until they are healthy.

Again, you should see a new agent and new actions.

Hint: You will need to use diffusion and hill climbing to get the sick people to search out the quarantine area. Look back at Journey if you need a reminder of how to do that coding.

Questions to answer once you are done...

- Does a quarantine area prevent diseases from spreading?
- How long must someone remain quarantined for it to be effective?
- When is quarantining necessary? Should we quarantine people with the flu? The common cold? Cancer?

