13.1 AgentSheets Tutorial 1a

*Introduction to Computational Science:*

*Modeling and Simulation for the Sciences, 2nd Edition*

Angela B. Shiflet and George W. Shiflet

Wofford College

© 2014 by Princeton University Press

Introduction

This first tutorial on *AgentSheets*® gives an introduction to the system and prepares you to understand a *AgentSheets* implementation of the model in Module 11.2, "Agents of Interaction: Steering a Dangerous Course," and to use the software to complete various projects in Chapters 11 and 14*.* The tutorial is in three parts (1a, 1b, 1c), and Tutorial 1a, which is in this document, develops a simulation of unconstrained growth.

One technique of modeling the movement of individuals and the spread of disease amongst the animals is a cellular automaton simulation. A related alternative is a grid-based, **agent-based** (**individual-based**) **simulation**.

For *a cellular automaton simulation*, the state of a grid cell might indicate the number of cattle at that location as well as attributes, such as weight(s), associated with the animal(s). Transition rules that specify the relationship of a cell with its neighbors determine the state of the cell at the next time step. For each time step, a cellular automaton simulation sweeps through every cell of the grid, updating its state.

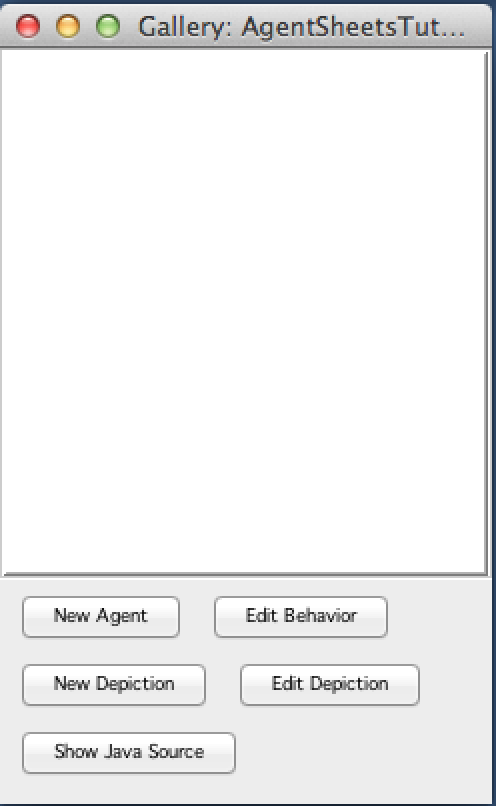
With an *agent-based simulation*, each animal is modeled as an autonomous, decision-making **agent** that has a **state**, which is represented by a set of **state variables,** or attribute values, and **behaviors**, which control its actions. A **method** or **procedure**, which is associated with a class, or breed or group, of agents, is a function that captures some or all of an agent's behavior. A simulation frequently includes several **global simulation variables**, which all agents can access. Agents often operate in an **environment** that arranges cells in a rectangular **grid**. (Individually based models exist that are not grid-based.) The environment, its neighboring agents, and the states and behavior of an agent determine the agent's new state. For each time step, instead of iterating through each grid cell, an agent-based simulation proceeds through each agent, revising its state.

Creation of a Model of Unconstrained Growth

In this section, we will create a model of unconstrained growth of bacteria on a petri dish. We start with one bacterium, allow it to move to a neighboring patch with a 15% probability, else have it divide with a 10% chance.

Start *AgentSheets*. If you have been working with another model in *AgentSheets*, from the *File* menu, select ***New Project*** or use the indicated shortcut to begin a new model. Save the project with a meaningful name, such as *AgentSheetsTutorial1a*. The project will be saved in a folder with that name. After subsequent work, save by selecting *Save* or using the indicated shortcut. **SAVE OFTEN**, particularly before you print or run a simulation. In a separate document, type the answers to all quick review questions.

After saving the file, we are prompted to define the agent size in pixels, or points on the screen. For the current tutorial accept the default size of 32 × 32 by clicking *OK*. Subsequently, we are presented with a ***Gallery*** window for creating agents.



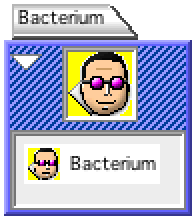
**Quick Review Question 1**

**a.** In the *Gallery* window, click *New Agent*. What prompt appears? Name the agent "Bacterium."

**b.** Define a new agent called "Background," and give the steps to do so in your answer document.

Currently, the two agents have the same depiction (appearance), so let's make them different.

**Quick Review Question 2**

**a.** Either double-click the default depiction for *Bacterium*, , or select the *Bacterium* box, , and click *Edit Depiction*, . What button at the bottom of the popup depiction window should we click to clear the default picture?

**b.** Use the **Pencil** tool to create a drawing of a bacterial. You may click a color to change the color of the pencil, the eraser () to remove part of the figure, the paint bucket () to fill an area with a color, and filled and unfilled rectangles and ovals. Create a brown filled circle by selecting brown, clicking the filled oval (), and holding down the shift key while dragging from the top left to the bottom right. If dissatisfied with the results, what to we click to erase the depiction? When through, click *Done*.

**c.** Let us now change the depiction of the *Background* agent. What steps do you follow to make the agent all yellow ()?

We are now ready to create a **worksheet**, or simulation world.

**Quick Review Question 3**

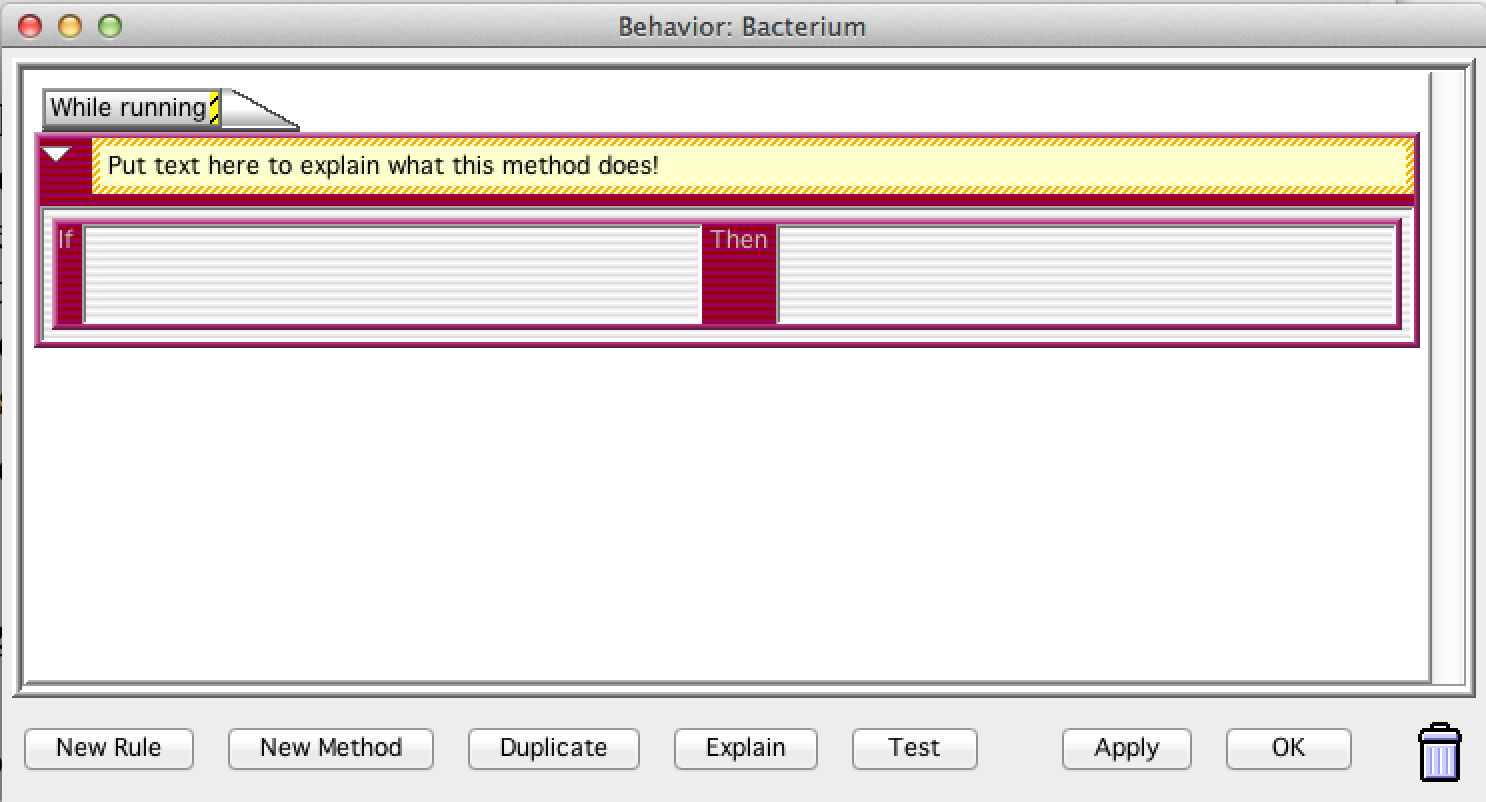
**a.** From the *File* menu, select *New Worksheet*. What appears?

**b.** Let's fill the worksheet with *Background*agents. First, select the *Background*agent from the gallery; second, select the **Draw Rectangle** () tool from the left of the worksheet; and then click and drag over the entire worksheet area. The worksheet is now covered with yellow *Background* agents. From the *File* menu, select *Save* and name the worksheet *Petri*.

**c.** In the middle of the worksheet, place one *Bacterium* using the Pencil tool. List the steps to do so. If not happy with the placement, use the **Selection** tool () to move the agent. To delete the agent, employ the **Eraser** tool (). Once again, save the worksheet.

We now want to give the *Bacterium* agent behaviors.

**Quick Review Question 4** Double-click the blue texture area of *Bacterium* in the gallery. Alternatively, select the gallery's *Bacterium* and press the *Edit Behavior* button. The following Edit Behavior window appears:



Initially, the behavior contains one method with **trigger** called ***While running***. A trigger indicates when to execute a method, and *AgentSheets* will execute the *While running* method each tick of the simulation clock, or each simulation cycle. A method contains **IF-THEN** **rules**; if a **condition** occurs, then *AgentSheets* should perform an **action** or actions.

The first rule we wish to have for the method has the following pseudocode, or English description of the command:

With a 15% chance, have the bacterium move at random to one of its eight *Background* neighbors.

Expressing the pseudocode in if-then format, we have the following:

If a random number between 0 and 100 is less than 15, then

move to a random *Background* neighbor.

**a.** To start the command, double-click the gray area after "If," which is to contain the condition. What appears? Alternatively, we could click the **Show Conditions Palette** () at the top of the *AgentSheets*window.

**b.** Under **basic** conditions, scroll to the ***% chance*** condition (), which has a default parameter of 50. Drag the condition from the Conditions Palette to the condition area after "If." Then, change 50 to 15. Will *AgentSheets* allow us to drag the condition to the gray area after "Then"?

**c.** Following the same ways to obtain the Show Conditions Palette, besides clicking the **Show Actions Palette** () icon at the top of the screen, list another way to obtain this palette.

**d.** Under the **basic** actions, scroll to the ***move random on*** action (). How do we get the condition to the rule?

**e.** We do not want the *Bacterium* agent to move on another bacterium but to move on a *Background* agent. In the action part of the rule, click and hold on the brown circle representing a *Bacterium*. What do you see? Select the depiction of a *Background* agent.

**f.** Click the **Apply** button at the bottom of the *Bacterium* behavior window to apply the method. Optionally, click OK or close the behavior window.

Although, we are not through with defining a bacterium's behavior, we now want to run the simulation to see if our work is correct.

**Quick Review Question 5** The three buttons () at the bottom left of the *petri* worksheet indicate to **Stop**, **Run**, and **Step** through the simulation one cycle at a time, respectively.

**a.** Click the Run button. The bacterium should move around the screen. Adjust the speed by dragging on the slider bar () to the right of the three buttons. Stop the simulation. Click the Step button several times. Does the bacterium always move?

**b.** Why not?

**c.** What happens when you press the **Reset** button () to the bottom right of the worksheet window?

Let us now complete *Bacterium*'s behavior.

**Quick Review Question 6** When the bacterium does not move, we would like for it to give birth with a 10% probability. Pseudocode for the rule follows:

With a 10% chance, have the bacterium give birth to a new *Bacterium* agent.

In if-then format, we have the following:

If a random number between 0 and 100 is less than 10, then

generate a new *Bacterium* at this location.

**a.** How do we get into the behavior window for *Bacterium*?

**b.** At the bottom of the behavior window, click the **New Rule** () button. What happens?

**c.** Drag the *% chance* condition from the first rule to the new rule's condition area. What happens? In the second rule, change the percent chance from 15 to 10.

**d.**How do we get the ***New*** action (), which is a basic action, to the proper location in the rule?

**e.** Click on the up arrow in the action portion of the rule. How many arrows appear? The arrows indicate the eight neighbors and the dot represents the current location. Select the dot. Apply the results and run the simulation on a medium-speed setting. You should see bacteria moving and increasing. After running for a while, stop the simulation. With the arrow selection tool, drag a *Bacterium* agent to another location. Repeat this process until you observe another *Bacterium* in the original location of the moved agent. Bacteria move on *Background* agents, but they can stack.

It is quite a challenge to remember what various procedures do. Consequently, we should write a comment for each method.

**Quick Review Question 7** Return to the behavior window for *Bacterium*. What does the comment area at the top of the *While running* method say? Add a meaningful comment for the method and apply the change.

The order of the rules is important. As soon as *AgentSheets* finds a condition that applies, the system executes the corresponding *then* clause and does not execute any other rules. Thus, for the current *Bacterium* behavior, approximately 15% of the time, the agent moves. For the remaining 85% of the time, the second rule is executed. Thus, generation of a new *Bacterium* only occurs with a probability of 0.85 × 0.10 = 0.085 = 8.5%: The bacterium must not move (probability 0.85) and then may divide (probability (0.10).

**Quick Review Question 8** To observe the impact of *IF-THEN* command, change the percent chance in the first rule from 15 to 100. Change the speed to fast so that the system resets quickly, and press Reset and then run. How many *Bacterium* agents are present? Change the percent chance back to 15.

When debugging, or finding errors, it is often helpful to follow the behavior of one agent.

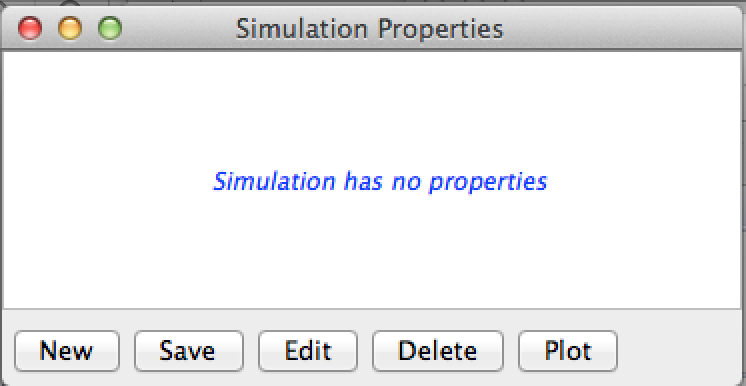
**Quick Review Question 9** Reset the worksheet.

**a.** Click on the one bacterium. Open the behavior window for *Bacterium*, and select the first rule. Click the **Test** button () at the bottom of the window for the command to execute once for that bacterium. Initially the entire rule is outlined in yellow "rope." What is outlined next?

**b.** Click Test several times, observing the behavior of the agent and the outlined parts of the rule. When the bacterium moves, what is outlined?

We would also like to have a plot of the number of bacteria.

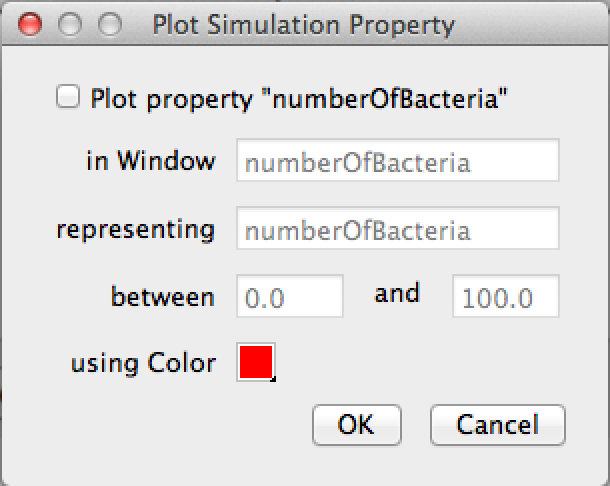
**Quick Review Question 10** First, we need to define a variable to store the number of bacteria. From the **Tools** menu at the top of the screen, select **Simulation Properties** to obtain the following popup window:



**a.** Click **New**. What is the current **Property Name**? Change the name to *numberOfBacteria* and click *OK*.

**b.** What is the default value for *numberOfBacteria*? Because we start the simulation with one bacterium, change the default value appropriately and click the **Save** button at the bottom of the Simulation Properties window.

**c.** Also, click the **Plot** button to obtain the following **Plot Simulation Property** popup window:



Check the **Plot property "numberOfBacteria**" checkbox and indicate what has changed in appearance of the popup window. Click *OK*. Save.

**d.** Run the simulation on a medium speed. Do the number of bacteria on the worksheet increase?

**e.** Stop the simulation. What is the value of *numberOfBacteria* in the Simulation Properties and on the plot?

**f.** The problem is we have not instructed *AgentSheets* to change the value of *numberOfBacteria*. Thus, we must edit the behavior of each *Bacterium* agent to increment *numberOfBacteria* when appropriate. Open the behavior window of *Bacterium*. In the Actions window, scroll to **attributes** and drag the **Set** action () to the appropriate area of the *Bacterium*'s behavior. Do we want to increase the count after moving or after generating a new agent?

**g.** The default variable name is *value*. Change *value* and *value(left)* to *numberOfBacteria*. Thus, we are asking *AgentSheets* to add one to the old value of *numberOfBacteria* and update the value of *numberOfBacteria* to the result. Run the simulation on a medium speed. Does the value in Simulation Properties or on the plot change?

**h.** The problem is we need to reference the **global** simulation property *numberOfBacteria*, not a **local** variable that is only known in *Bacterium*. In the *Set* command, place an @ symbol before each occurrence of the variable name, such as *@numberOfBacteria*, to refer to the global variable. Apply, save, and run the simulation. Do the *numberOfBacteria* change in the Simulation Properties window and the plot?

**i.** What is the appearance of the plot?

xxxReport? It is helpful to have a webpage that documents

xxxHelp – get Reference manual from internet