13.1 *AgentSheets* Tutorial 1c

*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 document contains *AgentSheets* Tutorial 1c, the third of a three-part tutorial. Tutorials 1a-1c give an introduction to the system and prepare 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*.* Tutorials 1a and 1b are prerequisites to this tutorial. Tutorial 1c improves your facility with using the *AgentSheets* Reference Manual and covers some of the commands employed in the implementation of Module 11.2's model. We recommend that you work through the introduction with a copy of *AgentSheets*, answering all Quick Review Questions in a separate document.

Formulas and When Creating a New Agent

**Type all requested commands in an *AgentSheets* project and an answer document. When requested to "look up" a command, consult the** [***AgentSheets* Reference Manual**](http://www.agentsheets.com/Documentation/windows/Reference/index.html)**.**

**Quick Review Question 1** Start a new *AgentSheets* simulation with default agent size and save the project.

**a.** How do we create a *Person* agent? Do so. Also, create a *Background* agent. Edit the depiction of the *Background* agent to be solid black.

**b.** How do we create new depiction for a *Person* agent? If necessary, in the [*AgentSheets* Reference Manual](http://www.agentsheets.com/Documentation/windows/Reference/index.html), look up Gallery Menu | New Depiction…; i.e., click "Gallery Menu" and then "New Depiction…." Create *Person* depictions for *Adult* and *Child*.

**c.** Make the *Adult* depiction have gray hair and a black background. Describe how you start and the fastest tool to use for filling areas.

**d.** Edit the child description to have a red background, no glasses, no brow lines, and no collar. What is the best tool for making the finer alterations to use?

**e.** We wish to initialize various agents to have random weights and ages. Edit a *Person*'s behavior. Start a new method and change the *On* trigger to be ***When creating a new agent***. How do we do this change?

**f.** We will have separate rules for *Adult* and *Child*. What is the condition to check if an agent is a *Child*? Place the condition in the *If* area.

**g.** In the [*AgentSheets* Reference Manual](http://www.agentsheets.com/Documentation/windows/Reference/index.html), look up ***random*** in Actions | Set | formula | Mathematical Operators, which has URL <http://www.agentsheets.com/Documentation/windows/Reference/tm_ti_formulasyntax.html>. In the action area of the rule, set a *Person*'s local *age*variable to a random integer (whole number) between 0 and 17, inclusive. Observe in the documentation's example, that the argument must be one more than the largest integer desired. Give the action in your answer document.

**h.** How do we duplicate the rule? Do so.

**i.** In the second rule's condition, change the depiction to *Adult*. Suppose we want an adult to have a random integer age between 18 and 98. The width of this interval is 98 – 18 = 80. What is the formula to return a random integer between 0 and 80?

**j.** What do we add to the formula in Part i to return a random integer between 18 and 98? Make the necessary changes for the adult's action.

**k.** We also want to initialize adult and child to have random floating-point, or decimal number, weight in kg. To do so, we have an additional action in each *Then* area. For the child, set the local *weight* variable to be 3 times the local *age* variable plus a random floating-point number between 0.0 and 10.0. In the documentation, find the symbol for multiplication and read carefully the example for a random floating-point number. Give the formula.

**l.** For the adult, set the local *weight* variable to be a random floating-point number between 52 kg and 113 kg. Give the formula. Type a comment for the method. Apply.

Make

**Quick Review Question 2**

**a.** Edit the behavior of *Background*. Change the trigger from *While running* to *When creating new agent*. Suppose a *Background* agent is initially to have a person stacked on top of it with the following probabilities: 5% chance of child, 10% chance of adult, 85% chance of no person. Thus, we want to generate a random floating-point between 0.0 and 1.0 and store the value in a local variable, *rand*. The following pseudocode indicates the utility of *rand*:

If *rand* is less than 0.05

have *Background* create a new child

else if *rand* less than 0.15 (and greater than or equal to 0.05)

have *Background* create a new adult

Should *AgentSheets* arrive at the *else-if*, we know that *rand* is great than or equal to 0.05, so we do not need to test for such. Moreover, 10% of the possible values of *rand* are between 0.05 and 0.15, so that approximately 10% of the time, *Background* will create a new adult.

Regardless of the situation, we need to define *rand*. Thus, with no condition, for the rule in *When creating new agent*, assign to *rand* a random floating-point between 0.0 and 1.0. Give the command.

**b.** Because this rule has no condition, *AgentSheets* will always execute the *Then* area. However, the system needs to execute the *if-then-else* segment indicated in the pseudocode. The solution is to have this code in another method, say *generate*, which we call after assigning a value to *rand*. To call *generate*, which we will define in the same agent's behavior, we employ the ***make*** action (). We are applying the method for the current *Background* agent. Do we need to change the dot to an arrow? Change the name of the method to *generate*.

**c.** Create a new method, *generate*. What trigger does it have? Change the default name to *generate*.

**d.** Following the pseudocode, give the condition for the first rule.

**e.** Give the action for this rule.

**f.** Duplicate this rule and on the second version make appropriate changes. Give the changes. Type comments for both methods in *Background*'s behavior. Apply.

**g.** Start a new worksheet and fill with *Background* agents. Save as *City*. Give the appearance. When a new *Background* agent is created, its *When creating new agent* method is executed. This method with the assistance of the *generate* method may create a *Person* agent with a *Child* or an *Adult* depiction on top of the *Background* agent. When a *Person* agent is created, its *When creating new agent* method is executed. This method establishes a weight and age for the *Person* agent.

**h.** To observe the local variables associate with an agent, first select the agent with the selection arrow. Then, under the Tools menu, pick **Agent Attributes**. For a *Person* agent, what attributes are presented? Besides examining the attributes in a stopped simulation, by leaving such a window open, we can follow the changing attributes for a particular agent during a simulation. Moreover, we can change the values to investigate certain situations. The ability to do such observations and alterations can be helpful in debugging our code.

Next to and Change

We now will add movement and change of depiction to our program.

**Quick Review Question 3**

**a.** Open the behavior for *Person*. We no longer need to view the contents of *When creating new agent*, so collapse the method by clicking the down-pointing white triangle () to the left of the comment. Where does the triangle point after closing? By clicking this triangle, we can open the method.

**b.** Let's have all *People* agents move at random, aging as they go. Give the two actions for the *Then* clause, or area.

**c.** When a *Child* becomes 18, the depiction should change to *Adult*. How can we handle this *if-then* situation in the *If* area?

**d.** Suppose the name of the method for dealing with such aging is *aging*. Give the command to call *aging* at the end of *Then*.

**e.** Start a new method called *aging*. The **Change** action () changes the current agent to the indicated depiction. Write a rule to handle the first sentence of Part c. Be sure to check if the depiction is for a child and its age is greater than or equal to 18. Apply. Select a *Child* agent and view its attributes. Step through the simulation and note how the agent ages and changes to an adult at age 18.

**f.** Otherwise, have the child gain 3 kg. Write the rule. Apply. As done in Part e, select a *Child* agent and step through the simulation verifying that the weight increases as expected.

**g.** Suppose an adult with a child as one of its eight neighbors loses 0.5 kg weight. (Keeping up with children takes a lot of energy!) The **Next to** condition () checks the neighbors. The adult can be next to one child or more and lose 0.5 kg. What comparison should we make in this condition?

**h.** Give the complete rule. Apply and verify that the weight is changing properly for an adult.

**i.** Otherwise, have an adult gain 0.5 kg. Give the rule. Apply and verify that the weight is changing properly for an adult. Write comments for the *aging* and *While running* methods. Apply.

Clearly, there are many other behaviors we could model. For example, we could have births and deaths. However, completion of Tutorials 1a-c will prepare us for the model in Module 11.2.