

## 13.1 NetLogo Tutorial 1b

*Introduction to Computational Science:  
Modeling and Simulation for the Sciences, 2<sup>nd</sup> Edition*

Angela B. Shiflet and George W. Shiflet  
Wofford College  
© 2014 by Princeton University Press

### Introduction

This document contains *NetLogo* Tutorial 1b, the second of a three-part tutorial. Tutorials 1a-1c give an introduction to the system and prepare you to understand a *NetLogo* 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. Tutorial 1a is a prerequisite to this tutorial, which develops a random walk simulation.

### Random Walk Simulation

In this tutorial, we will create a random walk simulation with from 1 to 10 walkers and to plot their average distance walked from the starting point. Create the simulation as directed, and in a separate document, type the answers to all quick review questions.

Start NetLogo and open a new file. Right click on the black world and select **Edit**. We will change the size of the screen to be rectangular and to have smaller size cells, or **patches**.

**Quick Review Question 1** Change the maximum x-coordinate for a patch to be 40, the maximum y-coordinate for a patch to be 30, and the patch size to be 8. With the origin patch being in the center, at (0, 0), give numbers of patches in the x and y directions. In each direction, be sure to add 1 for the patch at (0, 0). The patch size does not affect the number of patches in each direction.

Add a *setup* button. Go to the code section and type comments with your name and a description of the project. Save the program using a name, such as *Tutorial1b.nlogo*. Save your work after each quick review question. Start writing the *setup* procedure, and in the procedure give commands to clear everything and reset the ticks. The command to reset the ticks should always be the last command in *setup*.

In Tutorial 1a, we created one red turtle with the following command:

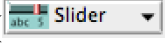
```
crt 1 [ set color red ]
```

We use *color* for turtles but ***pcolor***, or patch color, for patches. Whether for turtles or patches, colors can be designated by name (such as ***red***, ***brown***, ***yellow***, ***green***, ***blue***, ***pink***), number (such as 15, 35, 45, 105, 135, respectively), or RGB lists with amounts of

red, green, and blue light between 0 and 255 (such as [255 0 0], [205 133 63], [255 255 0], [0 255 0], [0 0 255], [219 112 147]).

**Quick Review Question 2** Under the *Help* menu, select the NetLogo dictionary and look up *color* or *pcolor*. Click the link for "Color section."

- Give the range of numbers associated with colors.
- Before *reset-ticks* in the *setup* procedure, write the command for *setup* to ask the *patches* agents to set their color to be **blue**. Clicking the *Check*, check that your commands are correct. On the interface level, click *setup* and observe the new background.

We would like for the user to have the option of varying the number of walkers. Thus, on the interface level in the drop-down menu that currently has *Button*, select *Slider* (). Click below the *setup* button to place the slider.

**Quick Review Question 3** In the resulting popup menu, give the name of the text field to establish each of the following. Also, make the changes and click *OK*.

- a global variable called *number-of-walkers*
- have at least one walker
- have at most 10 walkers
- have a default value of 1

**Quick Review Question 4** In *setup* before the command to reset the ticks, call a new procedure, called *setup-walkers*. Start the definition of *setup-walkers*.

- In *setup-walkers*, write the command to create *number-of-walkers* walkers, each with a random integer color between 0 and 99, i.e., less than 100. On the interface level, click the *setup* button several times and note the different colors for the walker.
- By default, a walker begins at the origin. Besides having a random color, we can have each walker start at a random location. The *x*- and *y*-coordinates of a walker are *xcor* and *ycor*, respectively. In the same creation of walkers after setting the color, set the *x*- and *y*-coordinates to be random integers between 0 and 9, inclusive. Click *Check* to check your work.

On the interface level, click *setup* several times to observe to observe the placement and color of the walker. Then, adjust the *number-of-walkers* slider bar, click *setup* again, and observe the results.

**Quick Review Question 5** Now, we want to be able to instruct the walkers to move.

- On the interface level, create a *go* button that will trigger execution of a *go* procedure. Where in the popup menu do we tell *NetLogo* the name of the procedure?
- What checkbox do we click to have *NetLogo* continuously run *go*? Also, click the checkbox *Disable until clicks start*, requiring the user to click *setup* before activating *go*. Click *OK*.

- c. On the *code* level, define a *go* procedure that asks each turtle (walkers) to walk to one of its neighbors.
- d. As the last command in *go*, make the clock tick. *Check* your work. On the interface level, click *setup* and then *go*, perhaps making the simulation *slower*.

Notice that when a walker goes off one edge, it continues its walk on the opposite boundary because of periodic boundary conditions.

**Quick Review Question 6** Right click on the display screen and select *Edit*. What should we do so that we no longer have periodic boundary conditions? Click *OK* and run the program to see how the walkers are now contained within the boundaries.

If we are interested in following the trail of each walker, we can employ a pen. The command ***pen-down***, or ***pd***, changes the mode to drawing a line on top of the patches and below the turtles. Correspondingly, ***pen-up***, or ***pu***, returns the mode to not drawing the lines.

**Quick Review Question 7** We will be adjusting the *setup-walkers* procedure to tell each walker to draw its path.

- a. First, type the *pen-down*, or *pd*, command after the right bracket that created the walkers. Click *Check* and give the error message.
- b. Where should we put the *pd* command? After checking your work, run the simulation.

To keep track of the distance each turtle has walked, we must know its starting point, and each turtle will have its own individual starting *x* and *y* coordinates and distance covered, say *x0*, *y0*, and *turtle-distance*, respectively. Thus, at the top of the code, have the following ***turtles-own*** command, which has no commas separating variables:

```
turtles-own [x0 y0 turtle-distance]
```

**Quick Review Question 8** In *setup-walkers* after establishing *xcor* and *ycor*, write the two commands to make *x0* and *y0* equal to the corresponding starting coordinate.

The following command using ***distancexy*** reports a turtle's distance from its starting point at (*x0*, *y0*):

```
distancexy x0 y0
```

**Quick Review Question 9**

- a. In *go*, after asking each turtle to move, ask it also to store its distance traveled in its variable *turtle-distance*. *Check* your work.
- b. At the end of *go*, make the clock tick.

**Quick Review Question 10** We wish to plot the mean, or average, *turtle-distance*.

- a. In the drop-down menu to the right of *Add*, what do we select? Add a plot.
- b. In the resulting popup menu, change the *X axis label* to "Time" and the *Y axis label* to "Mean Distance." However, we do not want to "plot count turtles," as indicated, but to plot the average *turtle-distance* of all the turtles. Look up **mean** in the *NetLogo Dictionary* to determine the plot command. In the command, we use *plot* instead of *show*. After entering the command, click *OK*.
- c. How do we make the graph window larger?

**Quick Review Question 11** Run the simulation.

- a. Describe the plot of mean *turtle-distance*.
- b. Why does the plot stop consistently rising?

Two projects of Module 9.5, "Random Walk," determined that the average distance walked is proportional to the square root of the number of steps until the walkers approach the boundary. Because the walkers take a step with each tick, the number of ticks is equal to the number of steps. In the next quick review question, we plot the square root of the ticks.

**Quick Review Question 12**

- a. Right click on the graph. What drop-down menu choice do we select to adjust the graph?
- b. Click *Add Pen*. The function *sqr* takes the square root of its argument. Type the *Pen update command* to plot the square root of *ticks*. If necessary, consult *NetLogo Dictionary*.
- c. Change the first *Pen name* to "Mean distance" and the second to "Square root steps." Change the *Y axis label* to "Mean Distance & Square Root Number of Steps." What do we do to have a legend appear? Click *OK*.
- d. Change to a slower speed. After the setup, run the simulation for 10 walkers, but stop the simulation when any walker gets close to the boundary. Does the plot of the square root of the number of steps roughly capture the trend of the plot of the mean distance traveled?

**Quick Review Question 13**

- a. How do we change the maximum number of walkers on the slider bar *number-of-walkers* to be 1000?
- b. By experimenting using 1000 walkers, find a number to multiply by the square root of the number of ticks in the plot command so that the two graphs are approximately the same. Be sure to have a blank on each side of the multiplication symbol, "\*". Thus, you are finding a function (an empirical model) that approximately captures the trend of the mean distance traveled.