

**NetLogo Tutorial Series:
Langton's Ant**

Nicholas Bennett
Grass Roots Consulting
nickbenn@g-r-c.com

July 2010

Copyright



Copyright © 2010, Nicholas Bennett. All rights reserved.

“NetLogo Tutorial Series: Langton's Ant” by Nicholas Bennett is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. Permissions beyond the scope of this license may be available; for more information, contact nickbenn@g-r-c.com.

See the accompanying NetLogo models for the source code copyright and license information.

Acknowledgments

Development of this curricular material and the accompanying NetLogo models was funded in part by:

- Santa Fe Institute Summer Internship/Mentorship (SIM) program for high school students;
- New Mexico Supercomputing Challenge;
- Project GUTS.

Participants in the above programs have also provided invaluable feedback on earlier versions of this material.

Introduction

One condition which sometimes produces complex behavior is high throughput. This throughput can be the product of a large number of agents operating and interacting simultaneously; a large amount of some physical material or property (e.g. electrical charge, fluid, gas molecules) acting in a constrained space or medium; or even a smaller number of agents repeating simple behaviors a large number of times, with feedback.

In this activity, we'll build a NetLogo model inhabited by ants¹ that follow a simple rule. In each time step, an ant will perform the following actions:

1. Move forward one space.
2. If the space on which the ant is now standing is black, then:
 - a. Paint the space the same color as the ant itself.
 - b. Turn left 90°.

Otherwise:

- a. Paint the space black.
- b. Turn right 90°.

This virtual ant is known as *Langton's ant*, after its inventor Christopher Langton, a pioneer in the field of artificial life [1][2].

We'll begin with a single ant, starting at the center of the NetLogo world. Then, we'll extend the model by adding more ants at different initial positions.

1. What pattern will result from a single ant with the behavior described above?
2. How many repetitions of the above steps will it take for a pattern to emerge?
3. How would you characterize the results? Complex? Chaotic?
4. Will the aggregate behavior be different with two or more ants, instead of one?

We could try to answer the above questions by doing the exercise on paper, but it would become very tedious, very quickly – probably long before the result is clear.

¹ Although the agents in this model don't behave like real ants, that's what we'll call them. Logo agents that move are generically called *turtles*; we'll use *ants* and *turtles* interchangeably in this exercise.

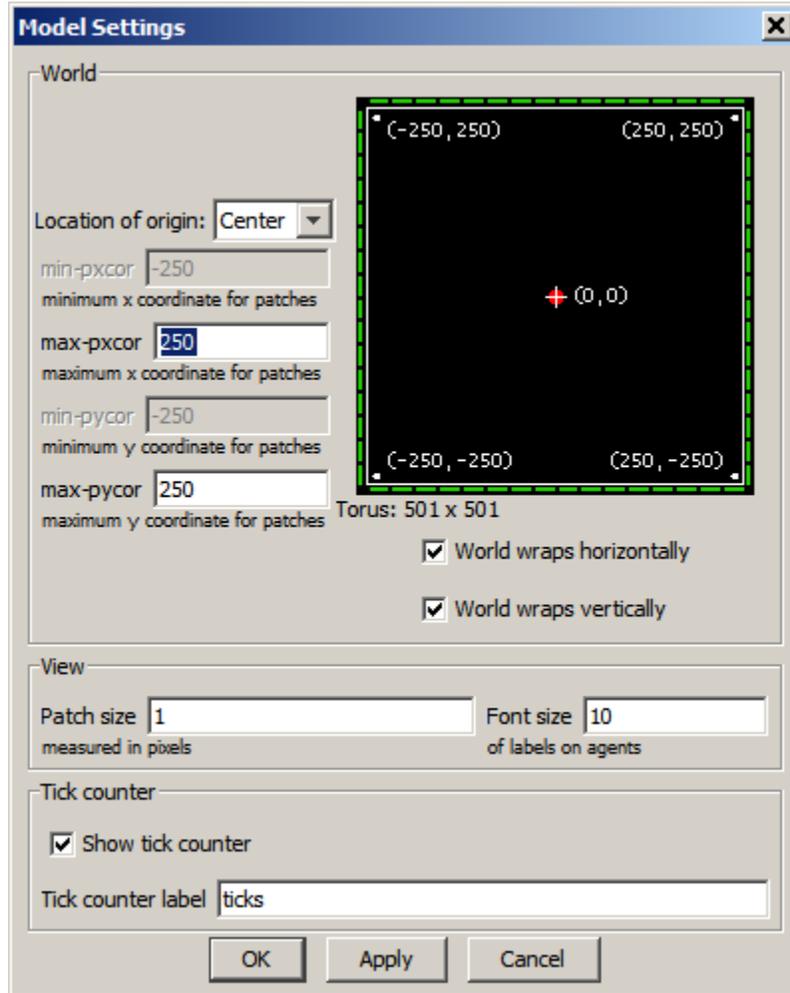
Task 1: Getting Started

(Before following the steps below, consider reading “The NetLogo Coordinate System”. Even if you've read it already, it might be useful to review it.)

1. From the Macintosh **Applications** folder, or from the Windows **Start/All Program/NetLogo** menu, launch the **NetLogo v4.1.1**² application (not **NetLogo 3D**).
2. Because we want to give our ants lots of room to maneuver, and because NetLogo colors an entire patch at once (i.e. no matter how large a patch is, we can't split it into different-colored regions), let's set up the NetLogo world with a large number of very small patches. To do this, click the **Settings...** button, near the top of the NetLogo window, and make these changes:
 - a. Leave **Location of origin** set to **center**.
 - b. Set the **max-pxcor** value to 250.
 - c. Set the **max-pycor** value to 250.
 - d. Make sure the **World wraps horizontally** checkbox is checked.
 - e. Make sure the **World wraps vertically** checkbox is checked.
 - f. Set the **Patch size** value to 1.0.

2 This tutorial was originally written for NetLogo v4.0.4, then updated for and tested with NetLogo v4.1 and v4.1.1. As written, it should work with any 4.x version, though there might be some slight differences between the screen captures and the actual displays.

The **World & View** window should now look like this:



3. Click the **OK** button; we now have a world that's 501 patches tall and 501 patches wide, with each patch displayed on a single pixel. Adjust the size and position of the NetLogo application window, as necessary, to display the entire world.
4. Save your model using the **File/Save** menu command. (In fact, you should do this after completing each task in this activity.)

Task 2: Setting Up the Model

Since we typically run a model several times in a single session, without closing and re-opening the model (or closing and re-opening NetLogo itself), we need a way to “clean up” anything left over from a previous run (if any), and prepare the model for a new run. By convention, when building NetLogo models, we usually refer to this cleanup and preparation operation as the model's *setup* process; our immediate task is to teach NetLogo how to perform this operation for our ant model.

To teach NetLogo to perform an operation, we instruct it in much the same way we would teach a very literal-minded person: “To do X, first do A, then do B, then ...”, with each step spelled out unambiguously. In this case, the first thing we'll tell NetLogo to do is clear the NetLogo world of any previously created ants, and reset all of the patches to the color black. Then, we'll tell it to create a single ant, in the middle of the NetLogo world.

(Before following these steps, we recommend you read “Programming in NetLogo”, if you haven't already done so.)

1. In NetLogo, click on the tab or button labeled **Procedures**. Here you see a blank area for writing the instructions that NetLogo uses for the various parts of the model.
2. Write the following in the space provided:

```
breed [ants ant]

to setup
  clear-all
  set-default-shape ants "bug"
  create-ants 1
  [
    set color red
    set size 15
    set heading 0
  ]
end
```

The code starts by declaring a breed of turtle agents called **ants**. It then tells NetLogo that the following steps make up the operation called **setup**:

- a. Execute the **clear-all** command. This is one of the instructions NetLogo already knows how to do; among other things, it removes any ants that have already been created, and sets the color of all the patches to black.

- b. Set the “bug” shape as the default shape for the **ants** breed.
- c. Create a single ant. In the **create-ants** command (like the more generic **create-turtles** command), the numeric value following the command is the number of agents to be created, and the subsequent set of square brackets contains a list of commands that the newly-created agents are asked to execute. In this case, we are asking our single ant to do the following:
 - i. Set its color to red. NetLogo uses a spectrum in which each color is identified by a number from 0 to 139.9; however, there are some symbolic constants – such as **red** – which are preassigned to the correct number value, so we usually don't have to remember the numbers themselves. (To see the entire NetLogo color spectrum, select the **Tools/Color Swatches** menu option.)
 - ii. Set its size to 15 times the default size. Without this, we wouldn't be able to see the bug shape we've selected for our ant, since we've configured the NetLogo world to use only one pixel for each patch, and since each turtle of size 1 fits inside a single patch.
 - iii. Sets its initial heading to 0 (i.e. the direction of increasing Y values).

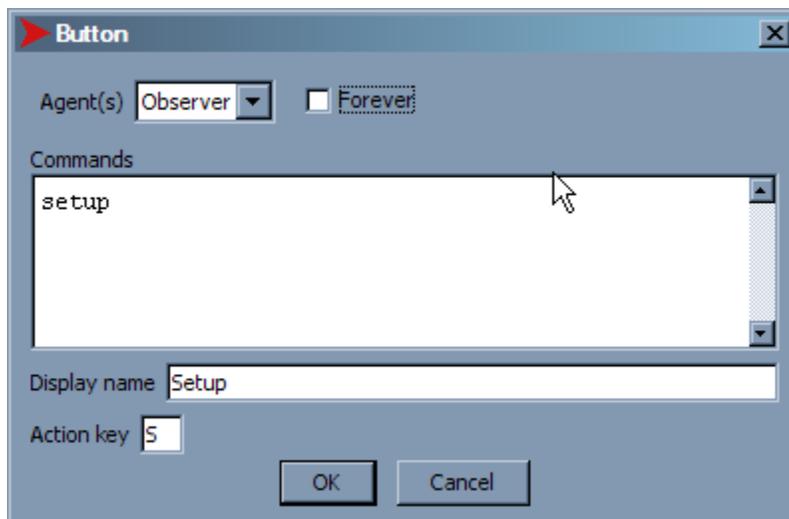
Note that we didn't include any instructions for setting the initial location of the ant. We'll learn how to do that soon enough – for now, we'll take advantage of the fact that NetLogo initially places turtles at the (0, 0) coordinates.

We now need to create a button to execute the new **setup** procedure:

3. Click on the **Interface** tab/button, so that the NetLogo world is visible again.
4. On the toolbar at the top of the **Interface** pane, select **Button** from the pull-down menu (next to **Add**); then click on the white space to the right or left of the NetLogo world.

5. In the **Button** properties dialog that appears, make the following changes (as necessary):
 - a. From the pull-down menu labeled **Agent(s)**, select **Observer**.
 - b. Make sure that the **Forever** checkbox is *not* checked.
 - c. In **Commands**, type **setup**.
 - d. If desired, type the text that you want to appear as the button title in **Display name** (by default, whatever you have typed in **Commands** will be used as a button title).
 - e. If desired, specify in **Action key** a shortcut letter that can be used to invoke the button commands with the keyboard.

Except for possible differences in the optional items, you should now have



something like this:

6. Click the **OK** button; you now see your new button in the NetLogo interface.
7. To test your work so far, click the button you created; you should see a single ant in the middle of the NetLogo world.

Task 3: Teaching Our Ant How to Behave

Let's write a procedure containing the ant behavior described at start of this document. By convention, we often call the procedure containing or invoking the main behavior(s) of our turtles the **go** procedure.

1. Click the **Procedures** tab/button.
2. *After* the **end** that marks the end of the **setup** procedure, add these lines to your code:

```
to go
  forward 1
  ifelse (pcolor = black)
  [
    set pcolor color
    left 90
  ]
  [
    set pcolor black
    right 90
  ]
end
```

Be sure to put a space before and after the equals sign in the line that begins with **ifelse** – those spaces, like many in NetLogo, are essential!

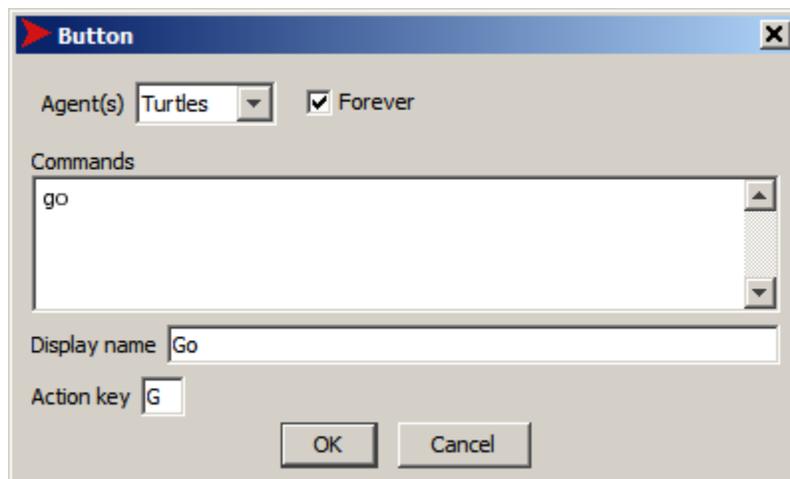
This code tells NetLogo that in order to **go**, a turtle must do the following:

- a. Move forward a distance of 1.
 - b. Test to see if the color of the patch (**pcolor**) where the turtle is standing is black. If so, then:
 - i. Set the color of the patch to be the same as the color of the turtle (**color**).
 - ii. Turn left 90°.
- Otherwise:
- i. Set the color of the patch to black.
 - ii. Turn right 90°.

(In Langton's original description of his ant behaviors, the order of steps is slightly different than that described here; the result, however, is identical.)

Now that we have a procedure containing the steps that we want our ant to follow, let's create a button to tell the ant to execute that procedure repeatedly:

3. Switch back to the **Interface** pane.
4. Follow the same steps as those you previously followed to create the **Setup** button; this time, we'll create a **Go** button.
5. In the button definition dialog that appears, make the following changes (as necessary):
 - a. Select **Turtles** from the **Agent(s)** pull-down menu.
 - b. Select/check the **Forever** checkbox. (This tells NetLogo to hold the button down when it's pressed, and to ask the specified agents to execute the specified commands repeatedly, as long as the button is down.)
 - c. In **Commands**, type **go**.
 - d. If desired, specify a **Display name**.
 - e. If desired, specify in **Action key** a shortcut key that can be used to execute the button commands with the keyboard.



You should now have something like this (again, the **Display name** and **Action key** values are optional; the important parts are the **Agent(s)** selection, the **Forever** checkbox, and the **Commands**):

6. Click the **OK** button to place your **Go** button in the NetLogo interface.

Task 4: Check For Errors – and Fix Them

If the titles of either of the buttons you've created are displayed in red text, that means that NetLogo's telling you there's an error in the button definition, or that the commands specified for the button aren't valid for the type of agent specified. If you Similarly, if there's a yellow bar displayed at the top of the **Procedures** pane, this means that one or more procedures has an error. Pay very close attention to spelling: computers aren't very good at recognizing that you meant “turtles”, when you wrote “trutles” (for example). Also, note that square brackets and parentheses are not interchangeable in NetLogo; the same goes for dashes, underscores, and spaces (e.g. NetLogo understands **clear-all**, but not **clear_all**, nor **clear all**).

If you see an error message, read it carefully, then use the message and the highlighted code to locate and correct the problem.

Task 5: The Moment of Truth – Running the Model

1. Click the **Setup** button you created, to prepare your model for execution.
2. Click the **Go** button to execute the model. (You may want to use the speed slider, located at the top of the NetLogo world, to slow execution down, so that you can see what's happening more easily.)

Questions for Discussion

1. Are the results what you expected?
2. How would you characterize the results?
3. Could you identify different kinds of patterns in the result? In other words, did the ant behavior seem to be more orderly at some times than at others?
4. Did knowing exactly what the ant is doing in each iteration help you to predict what the accumulated effect of thousands of iterations would be?
5. We've configured the NetLogo world as a torus; given that, what do you think would change – in the aggregate behavior – if the ant were to start in a different initial position, or face in one of the other three primary directions?
6. Do you think adding a second ant to the model would make the aggregate behavior more predictable, or less so?

Task 6: Adding a Second Ant to the Model

Now that your model's running, let's add a second ant to it. This time, the specific code to write will be left up to you.

Let's begin by placing the new ant as far as possible from the first ant. Clearly, the patches located at the four corners of the NetLogo world are as far as it's possible to get from the center patch. Pick any one of these patches, and figure out its *X* and *Y* coordinates – not as literal numbers, but in terms of NetLogo built-in reporters.

Make the necessary additions to the **setup** procedure to create a second ant, of a different color from the first, facing the opposite direction, and located in the corner you selected. (Hint: NetLogo includes a **setxy** command that instructs a turtle to move to the specified coordinates. Details on that command – as well as all of the other commands, variables, and reporters built in to NetLogo – can be found by selecting the **Help/NetLogo Dictionary** menu item.)

Remember that the rules to be followed by the second ant are the same as those followed by the first. Given that, do you think we need any changes to the **go** procedure? Do we need a second procedure with the same instructions?

Review the definition of the **Go** button by right-clicking (or *ctrl*-clicking) on it, and selecting **Edit...** from the pop-up menu. Do you think that when this button is pressed, only the first ant will execute the **go** procedure, or will all ants do so?

Task 7: Running the Model with Two Ants

Once you've completed the necessary changes to the model (and ensured that two ants are created correctly when you click the **Setup** button), run the modified model. Observe the model for several minutes (if necessary), until you're reasonably confident that you can describe the aggregate behavior you have observed.

Questions for Discussion

1. Were the results what you expected?
2. Did the overall aggregate behavior differ from that observed with only one ant? If so, in what ways did it differ?
3. Do you think the aggregate behavior would be affected by the relative position and facing directions of the two ants?

Task 8: Giving the Ant a Random Location and Direction

Read (in the NetLogo Dictionary) about the NetLogo *reporters* that generate random numbers (a reporter is simply an instruction that returns a value that can be used by your code). Then, incorporate one or more of these reporters in your model, to place the second ant at a random initial position, and facing a direction chosen at random from the four basic compass directions. (For now, don't worry about the possibility that the second ant starts on the same patch as the first. Also, it might simplify your **setup** procedure to place both ants in random positions.)

Task 9: Up to 14 Ants with Random Colors, Locations, Directions

Add a slider to allow the user to set the number of ants to create, from 1 to 14. When the ants are created, give them distinct colors, selected at random from the NetLogo **base-colors** list. Make sure that all of the ants start on distinct, randomly chosen patches, and that each starts out facing one of the four main compass directions (also chosen at random).

References

- [1] A. Gajardo, "The Industrious Ant of Langton", March 2002. [Online]. Available: <http://www.ing-mat.udec.cl/~anahi/langton/>. [Accessed: July 21, 2010].
- [2] "Christoper Langton", Wikipedia, May 2010. [Online]. Available: http://en.wikipedia.org/wiki/Christopher_Langton. [Accessed: July 21, 2010].