

## Exercise 1.1 – A First NetLogo Session

NetLogo is an interpreted language meaning you can type commands directly into a command line and see the results. In order to introduce NetLogo we will first type some commands directly into the command line. Later we will move on to composing and examining programs.

Open Netlogo

You will see the NetLogo main window. Inside this you will see another window in the form of a black square which is the 2D environment.

Notice at the top-middle of the screen are three tabs “Interface”, “Info” and “Code”. If you click on these you are taken to a different screen. “Interface” is where all input and output happens. “Info” is an editable page (like a wiki) that contains text information about the model. “Code” is where you write code directly into the built-in editor.

Ensure you are on the “Interface” screen (which is the default screen when NetLogo is started).

At the bottom of the screen you’ll see the “command center”.

At the very bottom of the screen you’ll see an “observer>” prompt.

This is where you can type commands directly into NetLogo to be executed immediately.

### Turtle commands and properties

At the “observer>” prompt type:

```
create-turtles 10
```

This will create 10 turtles and put them in the centre of the 2D environment. You should see some coloured dots in the centre. Since all 10 turtles are in the same place they are displayed over each other. Type:

```
ask turtles [forward 10]
```

This tells each turtle to execute the command “forward 10”. You should now be able to see each turtle more clearly since they have moved out from the centre of the environment. Turtles holds what is known as an “agentset” that contains all currently existing turtles.

Note: the colours and facing directions of each turtle are initialised randomly by NetLogo. In general NetLogo will initialise agents with default values if you do not explicitly set them. Also note that when you ask turtles to perform an action each turtle is selected in random order to perform that action.

Type:

```
ask turtles [pen-down forward 5]
```

This asks each turtle to execute two commands: put a pen down to leave a trace and move forward 5.

Hover the mouse over a turtle on the screen and right click. You should see a context menu with “turtle n” at the bottom (where n is some number). Select this and then select “inspect turtle n”. This will open a window showing the properties of the selected turtle (which are termed agent variables). Note the value in the “who” slot which should be the same as the n value of the turtle.

Back in the command center type:

```
ask turtle n [set shape "circle"]
```

Note: replace **n** above with the who value from the turtle.

Notice the change in shape of the turtle. Type:

```
ask turtle n [set label "hello world"]
```

Note: replace **n** above with the who value from the turtle.

Notice a text label is now attached to the turtle.

Close the turtle inspection window. Type:

```
clear-all
```

Clear-all removes all the turtles and resets the environment to a blank state.

### Using “with” for conditional operations

In the command center type:

```
create-turtles 10 [forward 10 set label who]
```

Placing commands after a create-turtles command sends those commands to each turtle just like an ask-turtles command. Here we send two: move forward 10 and set the turtle label to it's who value (which is a unique number for each turtle).

We can also attach a condition to commands sent to turtles. Type:

```
ask turtles with [who > 4] [forward 5]
```

This asks each turtle that satisfies the condition “who > 4” to execute the command “forward 5”. Note: the “with” reporter creates a new agentset composed of the agents from a given agentset (in this case turtles) that produce true for the condition (in this case who > 4).

Let's remove the labels and make each turtle display as a circle. Type:

```
ask turtles [set label "" set shape "circle"]
```

To relocate the turtles so they form a circle around the centre of the environment of radius 5, type:

```
layout-circle turtles 5
```

### Simple Network Operations

NetLogo has a number of commands to create and process links between turtles to form a network. Type:

```
ask turtles [create-links-with n-of 2 other turtles]
```

This tells each turtle to create two undirected links to two other turtles (which are selected at random from the population). You should see the links displayed in the environment.

Note: the “other” reporter returns all turtles apart from the current turtle. The “n-of” reporter returns some number (in this case 2) of randomly selected turtles.

We can layout the network using a “spring loaded” algorithm based on the links. This attempts to minimise the length of the links but keep the nodes from overlapping. Type:

```
repeat 1000 [layout-spring turtles links 0.2 5 1]
```

This applies the layout-spring command to turtles using the links and various parameters. Notice that we repeat this command 1000 times since each call makes a small adjustment. You should see the network layout in a form that better evidences the structure.

Links like turtles are agents meaning you can send them commands with the ask command. Type:

```
ask links [set color red]
```

In order to see the variables associated with links right-click over a link in the environment. You should see an “inspect link x y” option in a context menu (where x and y are some numbers). Select the inspect link option. Notice that links are identified by, and store, the turtles they connect in the link variables end1 and end2.

Close the link inspection window.

### The 2D environment

Hover the mouse over a blank part of the 2D environment and right click. You should see a context menu with an option “inspect patch x y” (where x and y are numbers representing the coordinates of the patch over which your mouse is positioned). Select the inspect patch option. Notice the variable values for the patch. Leave the patch inspection window open.

In the command center type

```
ask patches [set pcolor blue]
```

Here the command “set pcolor blue” is executed by each patch. The colour of all the patches in the environment should turn blue. Type:

```
ask patch x y [set pcolor black]
```

Replace **x** and **y** above with the values from your open patch inspection window. That patch should change to black.

For common colours in NetLogo you can use names. These map to numbers. The numbers for colours can be found by going to the NetLogo menu bar and selecting Tools > Color Swatches. Type:

```
ask patches [set pcolor abs(pxcor * pycor) / 30]
```

This tells each patch to set its colour based on a function of its location. You should see shades from black to white radiating from the origin at the centre of the environment. Each square is a single patch.

Click on the “settings” button in the top right of the NetLogo window. You should see various model settings including the dimensions and topology of the 2D environment. Here you can change the number of patches and the size (in pixels) in which they are displayed on the screen.

In the model settings widow change max-pxcor to 50 and max-pycor to 50. Change the Patch size to 5. Click OK. The environment is now 101 x 101 patches displayed at 5 pixels per patch.

In the command center type:

```
ask patches [set pcolor abs(pxcor * pycor) / 30]
```

You should see a set of shades and colours radiating from the centre of the environment. Notice the larger number and smaller sizes of the patches.

Right-click anywhere in the environment. Choose the “select” option. Notice the environment now has a border. You can drag or resize the environment by holding down the left mouse button and moving. Click outside the environment to deselect it. You can also select the environment by dragging over it with mouse button down.

Quit NetLogo

## **Exercise 1.2 – Buttons and Sliders**

In addition to typing commands into the command center we can put commands into buttons displayed on the screen. When the button is pressed the commands are executed. In addition we can create sliders that allow the user to set the value of a variable.

In this exercise we will create a slider and two buttons that allow for the creation of networks with different numbers of nodes.

Open NetLogo

### **Create a slider**

On the NetLogo toolbar (top left) click the “+Add” button.

Select “Slider” from the dropdown next to the Add button.

Move the mouse to clear area of the NetLogo window (to the right of the 2D environment) and click.

A slider dialogue should open. In the “Global variable” field type:

```
nodes
```

In the “Minimum” field type:

```
10
```

Leave other values at the default settings and click “OK”.

You should now see a slider labelled “nodes”. You can drag the slider to set nodes to different values.

Note if you select the slider by dragging over it you can move it to different places on the screen. While selected you can double click the slider to reopen the dialogue and change the values that determine the behaviour of the slider. You can also right click of the slider and select “edit” to open the dialogue. The same procedure can be used for any element on the screen. You can also delete elements by right clicking and selecting “delete”.

### **Create a button**

Now we will add a button. Again, from the toolbar click Add and select “Button” from the dropdown next to the Add button.

Move the mouse to a clear area and click.

A button dialogue should open. Enter the following:

For “Commands” enter:

```
clear-all
```

```
create-turtles nodes [set shape "circle"]
layout-circle turtles 5
ask turtles [create-links-with n-of 2 other turtles]
```

For “Display name” field type:

```
new network
```

Click OK.

You should now see a button named “new network”.

We can now test the slider and the button. Select a value on the slider. Click the “new network” button. You should see a network displayed in the environment.

Try clicking “new network” several times to see new networks. Also try different slider settings to get different numbers of nodes.

We will now create a new button that causes the network to be displayed in a spring format (that often looks better).

Create a new button (as we did previously) but this time when the dialogue opens enter the following:

For “Commands” enter:

```
repeat 1000 [layout-spring turtles links 0.2 5 1]
```

For “Display name” type:

```
spring layout
```

Click OK.

Try clicking on the “spring layout” button. You should see the network change layout.

**Task:** Use the slider and button you just created to create some new networks with different numbers of nodes and see how the different networks look in spring layout.

To see the spring layout process in operation do the following:

Set the nodes slider to 25.

In the NetLogo toolbar (at the top middle) move the speed slider to a slower (left) speed.

Click the new network button.

Click the spring layout button.

Note: you can move the speed slider while the network is moving to speed up or slow down.

**Task:** Create a new slider for a variable called “maxlinks”. Set the range from 1 to 10. Modify the code in the “new network” button so that each node makes maxlinks number of random links rather than 2. Experiment with different numbers of maxlinks by using the slider.

Quit NetLogo