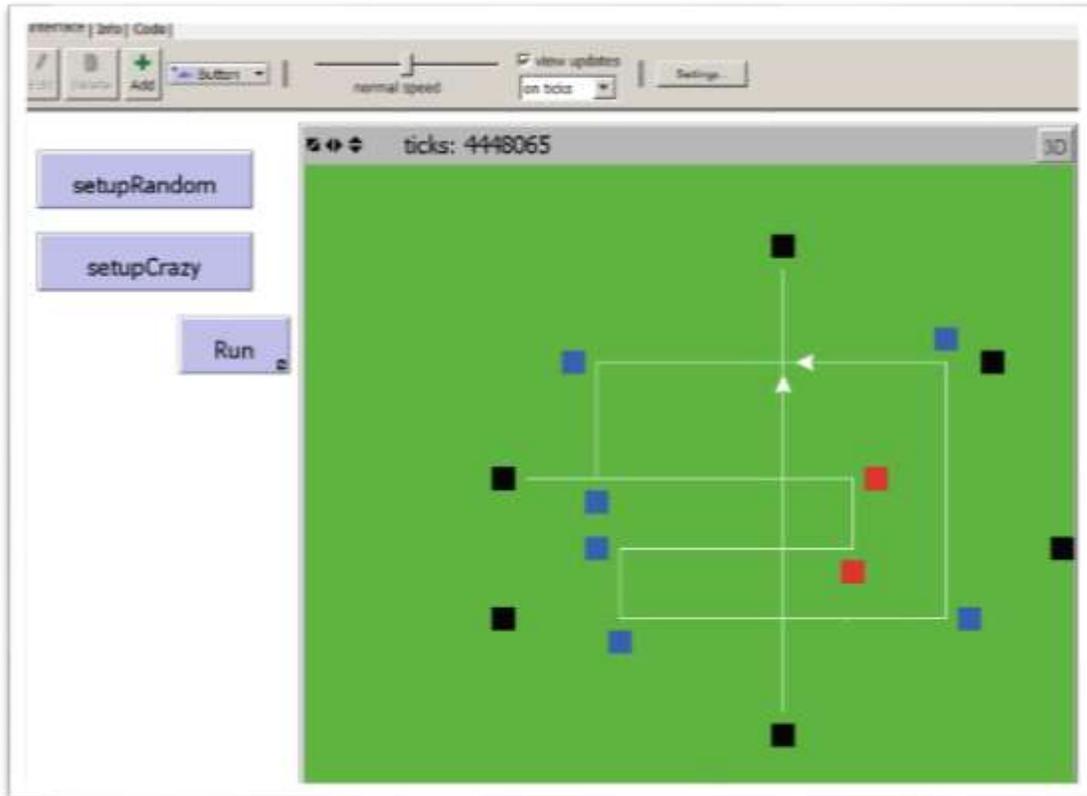


CS108L Computer Science for All Module 5: Bumper Turtles



$\text{min-pxcor} = -16, \text{max-pxcor} = 16, \text{min-pycor} = -16, \text{max-pycor} = 16$

The Bumper Turtles model created in this lab requires the use of *Boolean logic* and *conditional control flow*. The basic rules are:

- 1) Each turtle starts in the middle of a patch. This could be any patch, but not part in one patch and part in another.
- 2) Each tick, each turtle “looks” ahead one patch in its current heading.
 - a. If the patch ahead is black then the turtle makes a U-Turn.
 - b. If the patch ahead is blue, then the turtle makes a 90° left turn.
 - c. If the patch ahead is red, then the turtle makes a 90° right turn.
 - d. If the patch ahead is beautiful green grass, then there are two cases to deal with: If there is another turtle in that patch, then the turtle makes a U-Turn, otherwise, the turtle runs one step forward in the turf.

Note: the turtle should ONLY move when the patch ahead is green. This is because after the turtle turns, there might be another block in its new forward direction.



Setup Button:

Your program must have a setup button that when pressed must:

- 1) Clear the 2D world view.
- 2) Remove all the turtles (clear-all does both 1 and 2).
- 3) Set all patches to green. Then set specific, hardcoded patches to black, blue or red. You choose which specific patches to set black, blue or red such that the turtles you create in step (4) follow run around on a cool, creative, crisscrossing track. The screen capture on the first page shows an example of such a setup after the turtles have run around a bit. You may create more turtles and more interesting paths than that shown. Hint: Draw your pattern on graph paper before coding it.
- 4) Create at least 2 turtles each with a specific location and heading so that someplace along the path or that will enter the path created in step (3).

Hint: Setting a Particular Patch to a Particular Color

The NetLogo command:

```
ask patch 2 -4 [ set pcolor blue ]
```

will set the color of the patch with coordinates (2, -4)

Hint: Setting a Particular Turtle to a Location and Heading

Each turtle in NetLogo has a unique identification number. The identification numbers always start with 0 and count up to one less than the total number of turtles. The NetLogo **who** command reports a turtle's identification number.

These facts can be used to set properties of a particular turtle. For example:

- 1) **create-turtles 2**
- 2) [
- 3) **if (who = 0)**
- 4) [**setxy -3 0** ;; Lines 4 & 5 only execute for the
- 5) **set heading 180** ;; turtle with ID number = 0.
- 6)]
- 7) **if (who = 1)**
- 8) [**setxy 4 0** ;; Lines 8 & 9 only execute for the
- 9) **set heading 0** ;; turtle with ID number = 1.
- 10)]
- 11)]

Hint: Getting the Color of the Patch Ahead

See "Bumper Turtles" video for more details

We have seen before that within a turtle context, **pcolor** is the color of the patch the turtle is on. In this lab, we need to look at the color of the patch one ahead of the current turtle location. This can be done with the **patch-ahead** function as shown below:

```
ask turtles  
[  
  let colorOfPatchAhead green
```



```
ask patch-ahead 1 ;;1 is the number of patches ahead to look
```

```
[  
  set colorOfPatchAhead pcolor  
]
```

```
if (colorOfPatchAhead == blue)
```

```
[  
  ;; In this code block, do what you want to happen when  
  ;; the color of the patch ahead is blue.
```

```
]  
;; also have an if statement for each of the other possible  
;; colors
```

```
]
```

Hint: Finding out Whether there is a Turtle Ahead

See “Bumper Turtles” video for more details

To determine whether there is another turtle on the patch ahead of the current turtle, we build a triple compound command :)

Within a turtle context, **patch-ahead 1**, reports the patch that is one patch ahead of the turtle’s current location.

The NetLogo function, **turtles-on patch**, reports the set of all turtles that are on *patch*.

The NetLogo function, **any? agentset**, reports true if *agentset* contains at least one agent. Otherwise, it reports false.

Putting this all together, the statement:

```
any? turtles-on patch-ahead 1
```

Reports true if and only if there is at least one turtle on the patch that is one ahead of the current turtle’s current location.

Since this function reports a Boolean (true or false) it can be used in an **if** or in an **ifelse** statement.