Lesson 2 - Student Activity #2 Guide

Flower Turtles: Have your turtles paint a masterpiece!

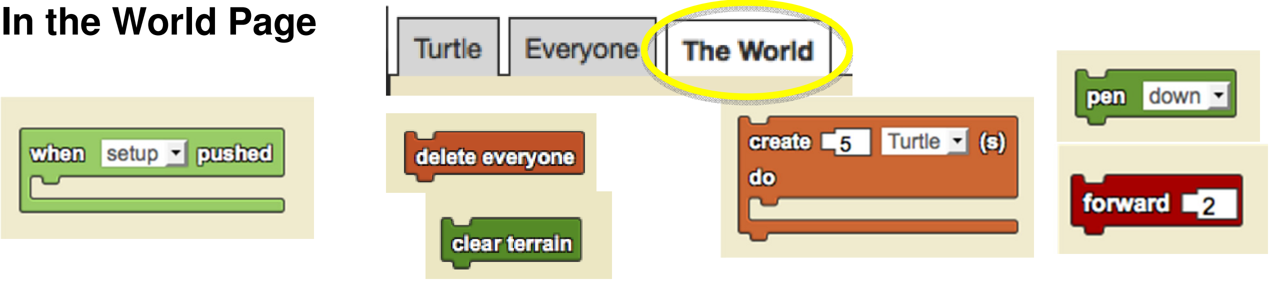
Your challenge is to make the turtles draw a flower pattern on Spaceland and to experiment with different kinds of turtle movement.

Note: Make sure to REMIX the project if you do any of the extensions.

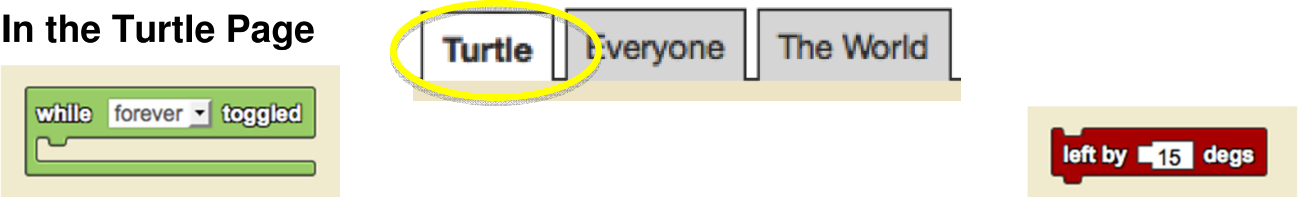
Guidelines: What your project needs to do

1. Start a new blank project.
2. Name it: Flower Turtles. If working in pairs, don't forget to put both partners' names in the project title.
3. **Create 5 turtles** that separate by 2 steps when the setup button is pressed.
4. **Have the turtles move and leave trails with their pens down** when the forever button is toggled.
5. When you are done, save and share your project.
6. If time, try some extensions.

In the World Page



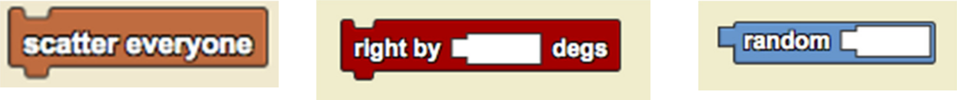
In the Turtle Page



Extensions:

1- Try changing the number of turtles created and scattering them.

2- Experiment with adding a slight wiggle to the turtle's walk using the random command block. The random block can be used in place of a number anywhere a number could fit.



Lesson 3 - Student Activity #1 Guide

Name: ___________

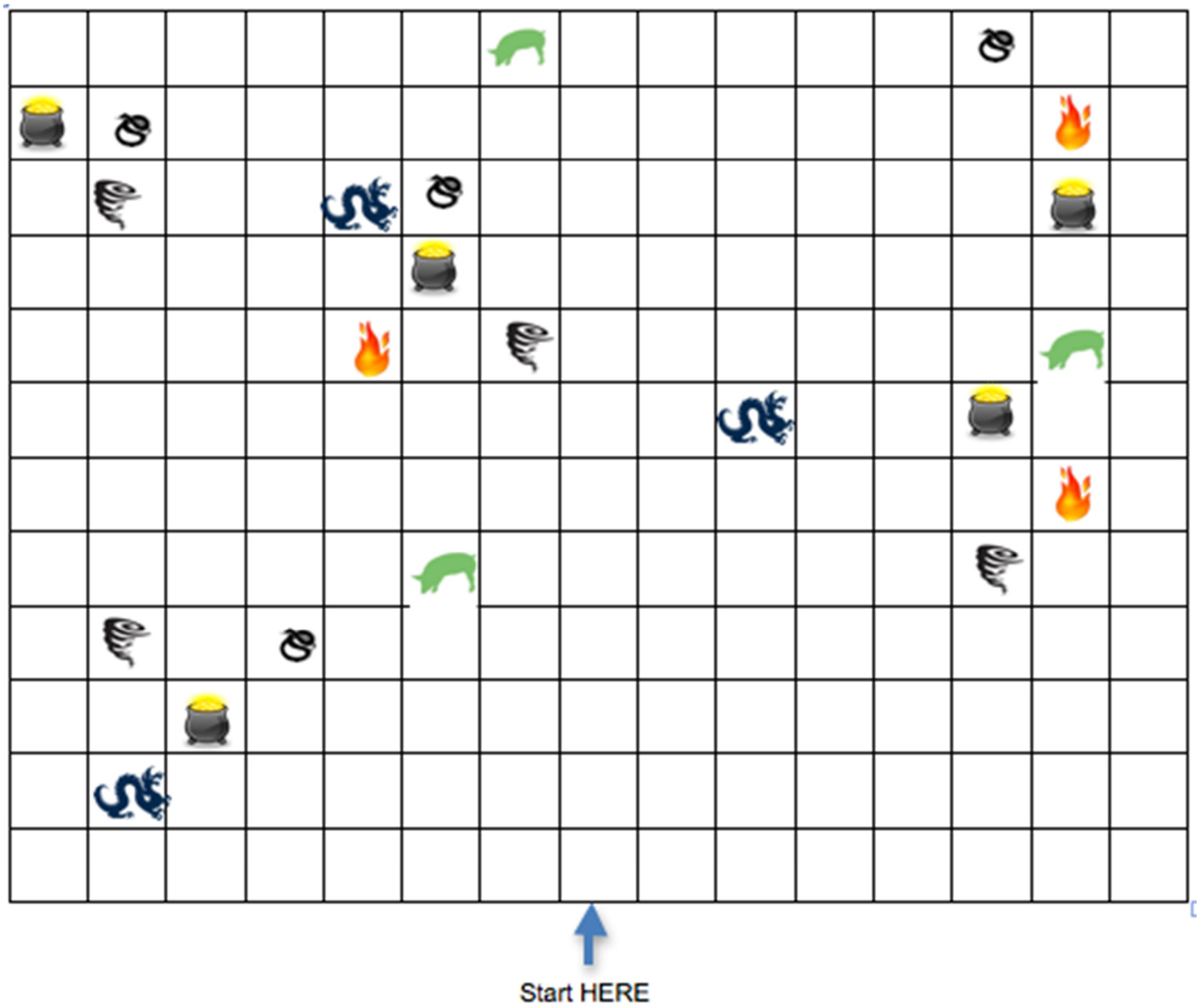
Trailblazer

Instructions and playing board

How to play: On the map, START at the designated position and heading in the direction of the arrow. Using a pencil, draw the path in the CENTER of squares. Pick up ALL the gold while avoiding the hazards, ending at START. Color the squares as necessary according to the following rules:

- **Take a step forward.**
- **If you are standing on a RED square, then turn right by 90 degrees**
- **If you are standing on a BLUE square, then turn left by 90 degrees**
- **If you are standing on a BLACK square, then turn right by 180 degrees**

Trade your map with a partner and figure out if following the landmarks lead you along the path to collect all of the gold while avoiding the hazards.



Lesson 3 - Student Activity #2 Guide

Bumper Turtles

- Start with the model “Bumper Turtles starter”. (Teacher provides the link.)
- This starter model already has a button called “Paint Landmarks” and some coding associated with it.
- Click on the “Paint Landmarks” push button and see the program execute the code provided.

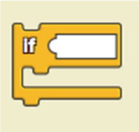
Your challenge is to make the turtles react to the landmarks created by the “Paint Landmarks” procedure, following the rules laid out in the Trailblazer activity.

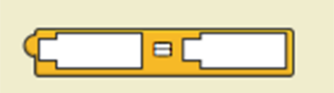
Note: DO NOT make changes to the “Paint Landmarks” procedure.

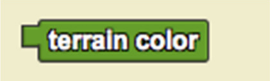
Guidelines: Have the turtles check the terrain color they are standing on.

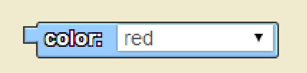
1. Remix the “Bumper Turtles starter” model; add your name(s) to the title of the project.
    a. If working in pairs, don’t forget to put both partners’ names in the project title.
2. In the World Page, instruct your turtles to react to the landmarks according to the rules in the Trailblazer activity.
    a. Use logic blocks that evaluate the color of the terrain and tell the turtle how to turn.
3. When you are done, save and share your project.

The new command blocks, to be used in addition to the blocks you used in Lesson 1 and 2, are:









Extensions:
Change how the turtles react to the colors (not just a turn, but a new shape or color for the agent).
Change the colors of the terrain stamps and make your turtle instructions match this new information.

Lesson 4 - Student Activity #1 Guide

Name ___________________

Dice and Data: Chances Are and Wiggle Walk

Introduction

Probability plays a large role in models of complex adaptive systems. We'll be programming our agents to mimic the movement of creatures in the real world. There are also chance events that occur when agents interact, such as the passing of a contagion from one person to the other.

PART 1: "Chances Are"

Directions

- In your group, roll 1 die 50 times in a cup, while the partner marks down the results for all 50 rolls using the chart provided with the possible rolls (1 through 6).

For each roll of the die, record the outcome with a tic mark below next to the number rolled. After 50 rolls, sum up each row and record the sum in the column on the right.

1		Sum:
2		Sum:
3		Sum:
4		Sum:
5		Sum:
6		Sum:

Discussion:

What was the distribution of the results?
Was one number more common?
Why might that be?

Extension:

If time allows, pool all of the data from the class and see what the sums are across groups.
If you graphed this data as a bar chart, what would the graph look like? (Draw it below.)

PART 2: Wiggle Walk

Next imagine that you were going to roll two dice instead of one die and you will subtract the second number from the first number. Do you think we will still get a flat distribution (in which the probability of rolling each outcome would be the same)?

Directions

- Each group should designate a Student 1 and a Student 2, each with their own die.
- This time, roll 2 dice simultaneously and collect the data from 50 rolls.
- Students are to SUBTRACT the Student 1 die number from the Student 2 die number.
- Record the outcome on the activity sheet with a mark next to the sum of the dice rolled (some will be negative numbers). After 50 rolls, sum up each row of tic marks and record the sum in the column on the right.
- For instance, if student 1 rolls a three and student 2 rolls a six, 3 - 6 = -3, so you would put a tic mark in the -3 row.

Area for recording tic marks of subtracted numbers

5		Sum:
4		Sum:
3		Sum:
2		Sum:
1		Sum:
0		Sum:
-1		Sum:
-2		Sum:
-3		Sum:
-4		Sum:
-5		Sum:

Discussion:

What number gets created most often from the rolling and subtraction? Why?
Given this chart, what should be the most commonly created number? Why?

Extension:

If time allows, graph this data as a bar chart. What shape does the graph look like? (Draw it below.)

To understand this phenomenon, let's look at the underlying statistics:
We know that with one die we have equal probability of rolling a 1, 2, 3, 4, 5, or 6.

Fill in the chart below that shows the outcome of rolling two dice. The numbers down the first column represent the number rolled by Student 1; the numbers across the top represent the numbers rolled on the other die, by Student 2. Subtract the second number from the first number.

Student 2

Student 1

--	1	2	3	4	5	6
1	0					
2						
3						
4						
5						
6						

Discussion:
How many ways are there to make 2?
How many ways can you create a 0?
What about other numbers?
What is the most often-seen number in the matrix?
How many times is it seen?

Next, fill in the result of a left turn of random 6 followed by a right turn of random 6.

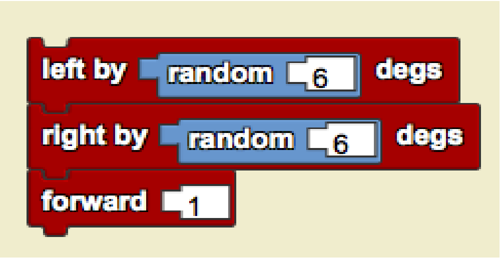
Right turn

Left turn

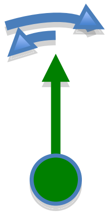
--	1	2	3	4	5	6
1	0					
2						
3						
4						
5						
6						

Discussion:
How many ways are there to make 0?
How many ways can you roll a +2?
How many ways can you roll a -2?
What about other numbers?
What is the most often-seen number in the matrix?
How many times is it seen?

Finally, consider rolling two dice where one represents the degrees to turn to the left and the other die represents the degrees to turn to the right. That's similar to using a left turn random 6 followed by a right turn random 6, then taking a step forward.

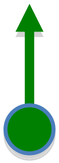


```
left by random 6 degs
right by random 6 degs
forward 1
```

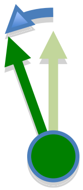


Turtle seen from above with initial heading.

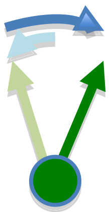
What is actually going on?



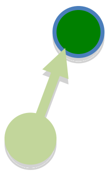
1. Agent has an initial heading.



2. Agent turns left by some random number of degrees between 1 and 6. In this case, it is 2 degrees. (Image not to scale)

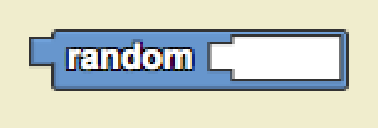


3. Agent turns right by some random number of degrees between 0 and 5. In this case, it is 5 degrees, so the final heading is 3 degrees to the right of the original heading. (Image not to scale)

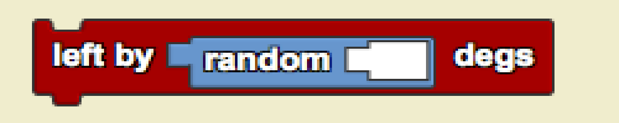


4. Agent takes one step forward at new heading. (Image not to scale)

In StarLogo Nova:



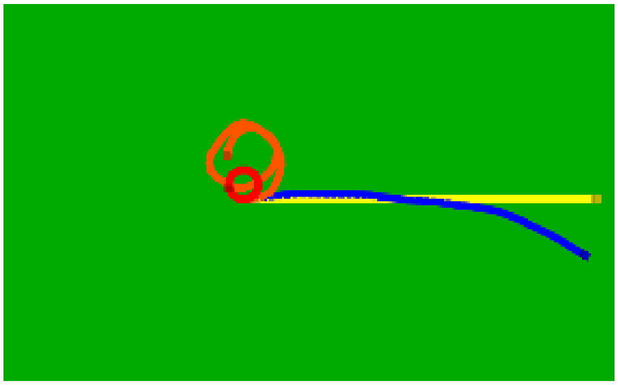
```
random
```



```
left by random degs
```

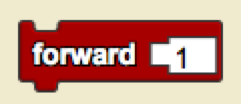
In StarLogo Nova, we use the random command to simulate the roll of a die.   Random 6 would give me the result of rolling a 60-sided die with numbers 1 through 6 on the sides. Random functions can also be used within other commands to implement random behavior.

Self-test:



Match the command blocks a, b, c, and d with the path created by an agent following those blocks.

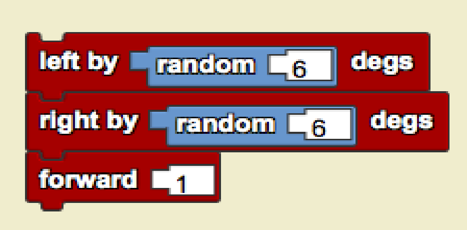
a.



```
forward 1
```

color: __________

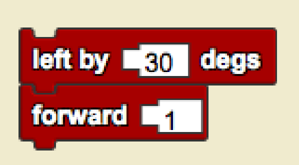
b.



```
left by random 6 degs
right by random 6 degs
forward 1
```

color: __________

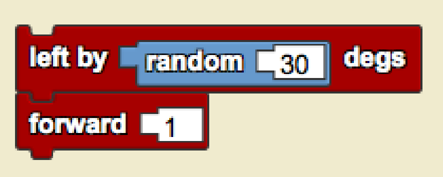
c.



```
left by 30 degs
forward 1
```

color: __________

d.



```
left by random 30 degs
forward 1
```

color: __________

Lesson 4 - Student Activity #2 Guide

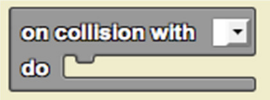
Name ___________________

Colliding Turtles

Introduction

In this activity we are going to implement agents interacting with other agents upon colliding. Collisions occur when two agents bump into one another. (They do not need to be centered on the same patch, just touching.) This is different from Bumper Turtles; in Bumper Turtles, *agents were responding to colored patches in their environment*, not other agents.

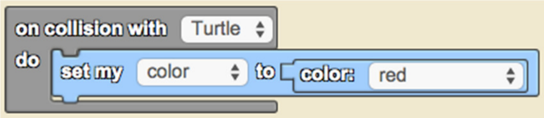
We will use a collision block that looks like this. Use the pull-down arrow to select what type of object to collide with. Then put the commands that should run or execute when the collision occurs in the area below the notch.



```
on collision with [ ▾]
do
```

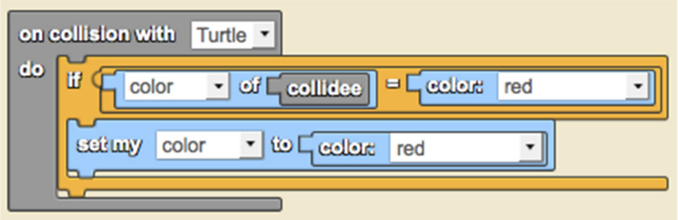
Select the object to collide with

Here's an example:



```
on collision with [Turtle ⬍]
do
    set my [color ⬍] to [color: [red ⬍]]
```

Here's another example with a conditional instruction. **The set color command only executes if the condition is true.**



```
on collision with [Turtle ▾]
do
    if [color ▾] of [collidee] = [color: [red ▾]]
        set my [color ▾] to [color: [red ▾]]
```

Your challenge is to make the turtles react to another agent upon collision.

<u>Guidelines:</u>

1. Start from a new blank project or remix the "Colliding Turtles starter" model in the Project GUTS gallery.
2. Don't forget to put both partners' names in the project title.
3. Create 50 blue turtles (refer to Lesson 2) and 5 red turtles. (Will need 2 'create do' blocks.)
4. Have the turtles move around with a wiggle walk. (Work in the turtle page with a forever block.)
5. Use a collision block and have the turtles change a trait like color after colliding with another turtle.
6. When you are done, save and share your project.

Lesson 5 - Student Activity #2 Guide

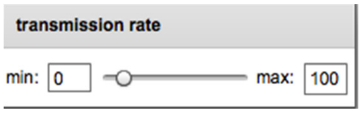
Modeling the Spread of Disease

Part 1: Altering Colliding Turtles to create an Epidemic Model

Your challenge is to make the turtles spread disease to one another when they collide. You will learn how to use sliders to change a variable in a computer model.

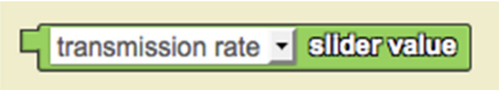
Guidelines:

1. Remix your Colliding Turtles project or use the ‘Epidemic Starter’ model.
2. Name it ‘Epidemic Model: *your name(s)*’ *Don’t forget to put both partners’ names in the project title.*
3. Create 300 blue turtles and 5 red turtles
4. Create a transmission rate slider; set the maximum to 100.
5. Use the value in the transmission rate slider as the probability of passing on the disease.
   Note: you will need to use the random function to mimic rolling the 100-sided die.
6. Save and test your model.



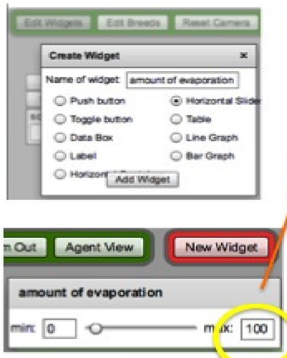
transmission rate

min: 0 max: 100



How to add a slider

Step 1- Edit Widgets, Select Horizontal Slider to add a slider, give it a name.



Create Widget

Name of widget: amount of evaporation

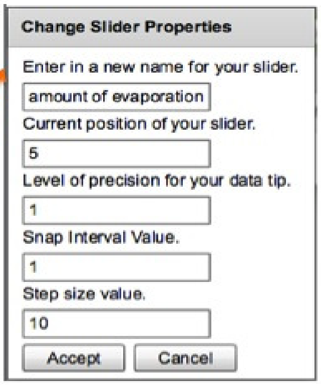
○ Push button ● Horizontal Slider
○ Toggle button ○ Table
○ Data Box ○ Line Graph
○ Label ○ Bar Graph
○ Horizont...

amount of evaporation

min: 0 max: 100

Must hit ‘enter’ after the number.

Step 2- Edit the slider by double-clicking in the gray area.



Change Slider Properties

Enter in a new name for your slider.
amount of evaporation

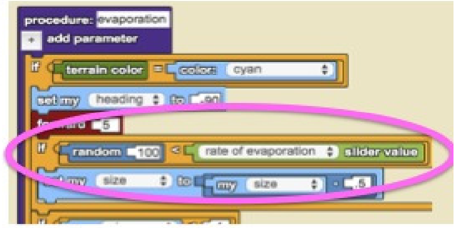
Current position of your slider.
5

Level of precision for your data tip.
1

Snap Interval Value.
1

Step size value.
10

Step 3- Make sure the Slider Value is coded in the program.

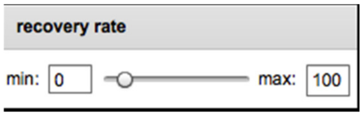


```
procedure: evaporation
+ add parameter
if terrain color = color: cyan
  set my heading to .90
  forward 5
if random 100 < rate of evaporation slider value
  set my size to my size - .5
```

Part 2: Customizing your model [adding in recovery]

Your challenge is to make the turtles recover from the disease.

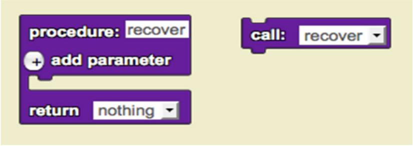
1. Create a recovery rate slider.
2. Create a recovery procedure.
3. Use the value in the recovery rate slider as the probability of recovering from the disease at each step.



recovery rate

min: 0 max: 100

Recovery Procedure: At each step, a sick person has a chance of recovering so we will need recovery to be called when the 'forever' button is toggled. You will need to use the random function to mimic rolling the 100-sided die as we did in the transmission case.



```
procedure: recover
  add parameter
return nothing
```

```
call: recover
```

Note: the "call recover" block needs to be within the "when forever toggled" loop.

Testing your model

- Save and test your model.
- Try changing the recovery rate. Did you see any new outcomes or patterns?
- Notice that even when the recovery rate is really low, the disease goes away. Why do you think that is? Is this realistic?

When you are done, save and share your project.

Progress Monitors for Coding Projects in Module 1

Flower/Painting Turtles Progress Monitor (Agent Affecting Environment)

Goals:

- Create Turtles that are separated by 5 steps. *[hint: Setup]*
- Have them draw a flower. *[hint: Forever with pendown. forward and left by]*

Bonus:

- Use the random block in turtle movement commands. *[hint: "right by" random amount]*
- Use one block you haven't used before.

Bumper Turtles Progress Monitor (Agent-Environment Interactions)

Goals:

- Add logic so turtles react to red squares by turning right by 90 degrees
- Add logic so turtles react to blue squares by turning left by 90 degrees
- Add logic so turtles react to black squares by turning around 180 degrees

Bonus:

- Change the number of landmarks drawn in the "Paint Landmarks" procedure.
- Use one block you haven't used before.

Colliding Turtles Progress Monitor (Agent-Agent Interactions)

Goals:

- Create turtles of two different colors, red and blue, and do not have them leave trails.
- Have the turtles move forward with a little wiggle in their walk.
- Upon colliding with a red turtle, have blue turtles react by changing their color to red.
- Add logic so turtles react to each other.
- Change a turtle's trait after a collision. [size, color, shape, ?]
- Save your project, upload, and share:

Bonus:

- Create a new turtle after a collision.
- Use one block you haven't used before.

Epidemic Model Progress Monitor

Goals:

- Create several hundred blue turtles and a few red turtles. *[hint: Setup]*
- Make them wiggle: *[hint: Forever]*
- Create a collision block in which blue turtles turn red when they encounter red turtles. *[hint: Lesson 4 Student Activity #2 Guide]*
- Create a slider and an if-then for the probability of transmission rate. *[hint: Lesson 4 Student Activity #2 Guide]*
- Create a slider and an if-then for the recovery rate. *[hint: Lesson 5 Student Activity #2 Guide]*

Bonus:

- Add a line graph with a line for infected turtles and one for healthy turtles. [*hint: Lesson 6 Student Activity #1]*
- Add a slider (or more) for one (or more) of the following:
  - Number of original healthy population
  - Number of original sick population
- Use one block you haven’t used before.