Lesson 2 - Student Activity #2 Guide

Flower Turtles: Have your turtles paint a masterpiece!

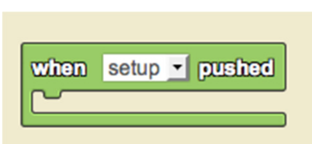
Your challenge is to make the turtles draw a flower pattern on Spaceland and to experiment with different kinds of turtle movement.

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

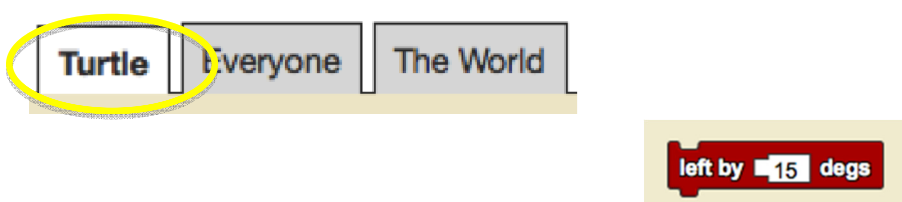
<u>**Guidelines**</u>**:** What your project needs to do

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



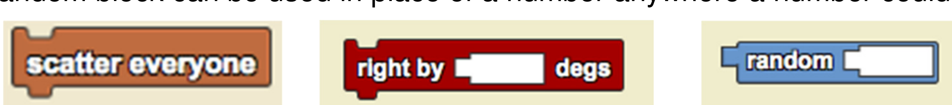
In the Turtle Page



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: ___________

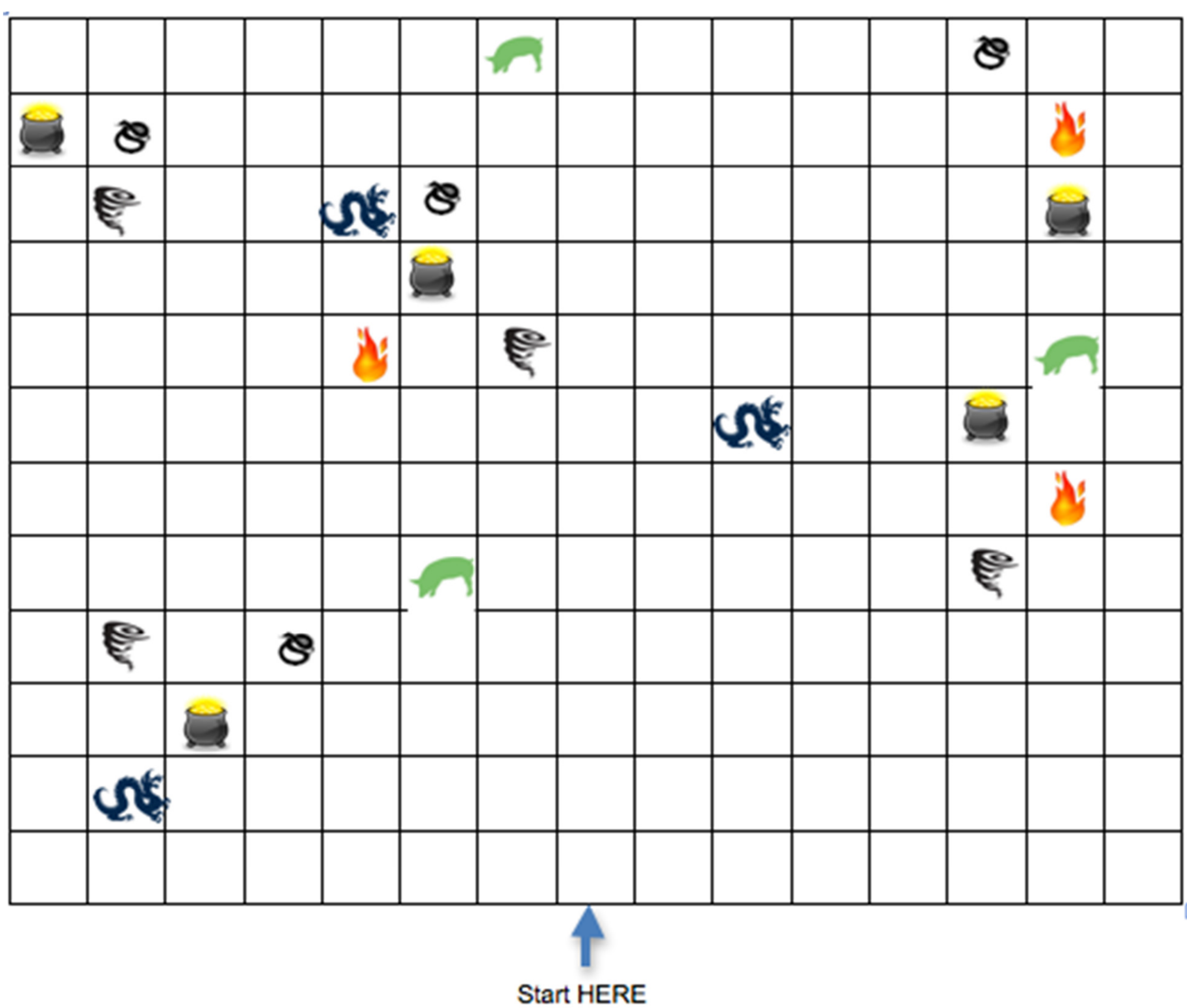
Trailblazer

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, <u>ending at START</u>. 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.

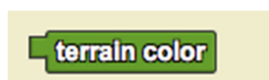
Your challenge is to make the turtles react to the landmarks created by the “Paint Landmarks” procedure, following the rules laid out in the Trailblazer activity.

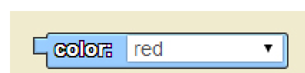
Note: DO NOT make changes to the “Paint Landmarks” procedure.

<u>Guidelines</u>: 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.

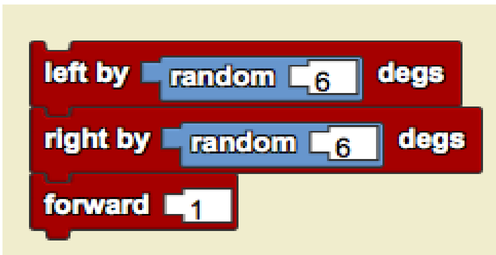
Right turn

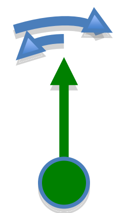
Left turn

--	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?

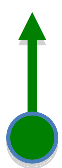
Finally, consider rolling two dice where one represents the degrees to turn to the left and the other die represents the degrees to turn to the right. That's similar to using a left turn random 6 followed by a right turn random 6, then taking a step forward.



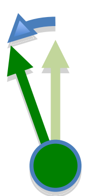


Turtle seen from above with initial heading.

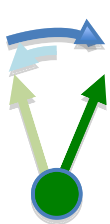
What is actually going on?



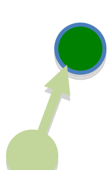
1. Agent has an initial heading.



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)

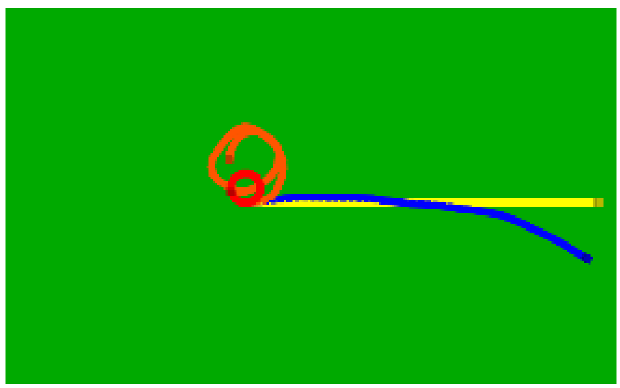


4. Agent takes one step forward at new heading. (Image not to scale)

In StarLogo Nova:

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:



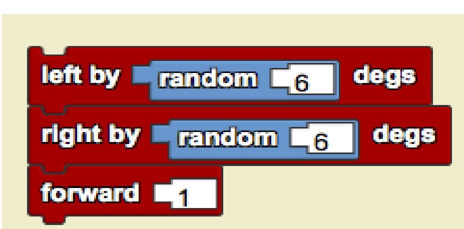
Match the command blocks a, b, c, and d with the path created by an agent following those blocks.

a. 

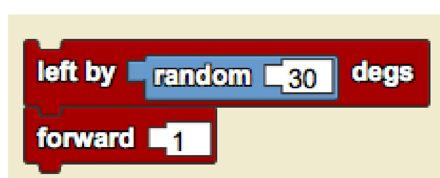
color: __________

c. 

color: __________

b. 

color: __________

d. 

color: __________

Lesson 4 - Student Activity #2 Guide

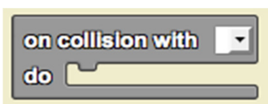
Name ___________________

Colliding Turtles

Introduction

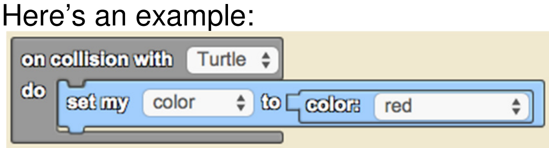
In this activity we are going to implement agents interacting with other agents upon colliding. Collisions occur when two agents bump into one another. (They do not need to be centered on the same patch, just touching.) This is different from Bumper Turtles; in Bumper Turtles, *agents were responding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

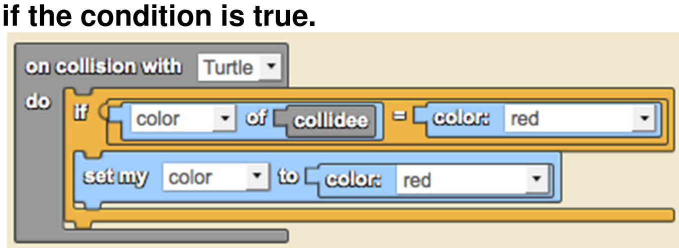


Select the object to collide with

Here's an example:



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



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

Guidelines:

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

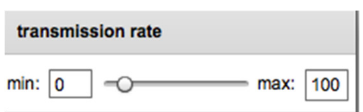
Modeling the Spread of Disease

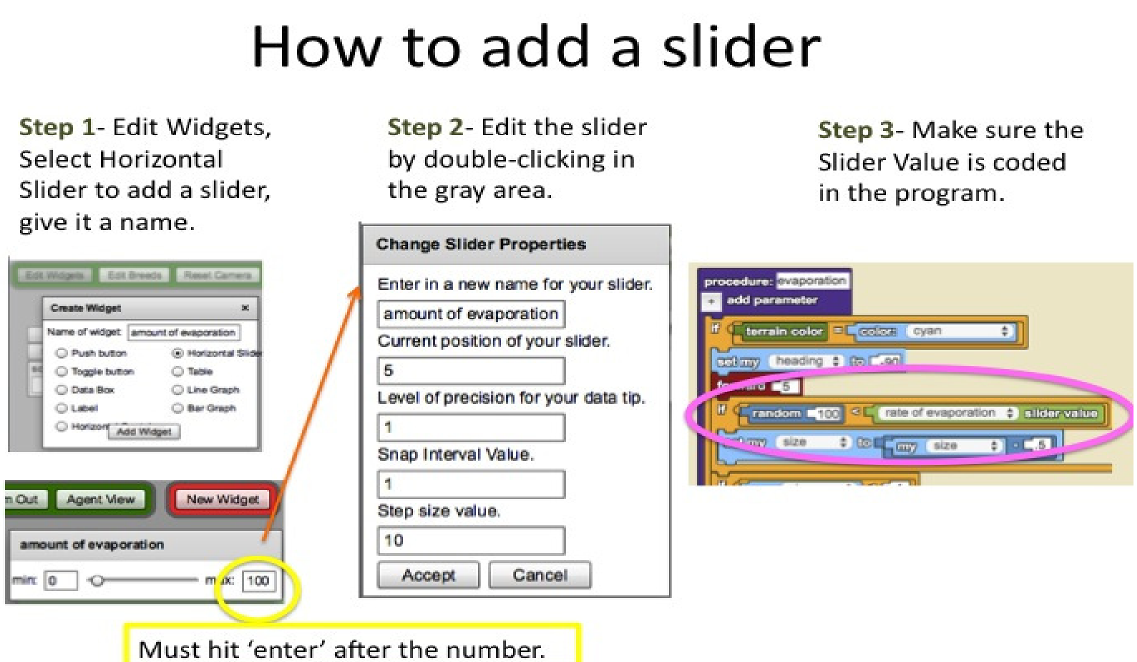
Part 1: Altering Colliding Turtles to create an Epidemic Model

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.

<u>Guidelines:</u>

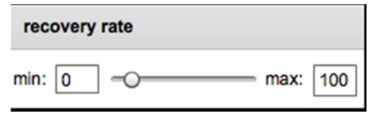
1. Remix your Colliding Turtles project or use the 'Epidemic Starter' model.
2. Name it 'Epidemic Model: *your name(s)*' *Don't forget to put both partners' names in the project title.*
3. Create 300 blue turtles and 5 red turtles
4. Create a transmission rate slider; set the maximum to 100.
5. Use the 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.





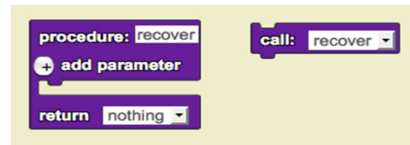
Part 2: Customizing your model [adding in recovery]

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 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.



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:

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

Bonus:

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

Lesson 2 - Student Activity #2 Guide

Inspecting the Water Pumping Model

Look under the Hood
Now we are going to get to know the code that makes up the base model!

1) Open your saved StarLogo Nova Water Pumping base model.

2) Navigate to the code section.

3) Use the **Model Observation Form** as you and your programming partner take turns looking at the code. (Remember to use your **driver** and **navigator** roles and switch roles from time to time.) Complete the form by running the model and looking at the code.

4) Which part of the code have you and your partner been assigned?

5) Write down what the code in your assigned section does.

6) Diagram the program's execution loop.

Here is a tip:

- You can refer to your StarLogo Nova Command Blocks and CS Concepts reference sheets from Module 1.

Lesson 2 - Student Activity #3 Guide

Adding a Slider for Evaporation Rate

Adding a Slider

In Activity 2 you edited the code to change the evaporation rate. In this activity you will learn a better way to change the evaporation rate.

1) **REMIX** your model and edit the name to "Water Pumping base model ***your name your partner's name*** **mod1**"
2) Add a slider for the evaporation rate and all necessary code.
3) Write down the slider settings you set.
4) Run an experiment using the evaporation rate slider. Use the **Experimental Design Form** to design your experiment first.
5) Record the data from your experiment and summarize your results.

Here is a tip:

- You can write up your results on a separate piece of paper. You can use graphs and/or tables to help you.

When you are done, upload and share your project.
Don't forget to put both partners' names in the project title.

Lesson 3 - Student Activity #1 Guide

Adding a Water Pump

In this activity, you will be adding a new pump that pulls water from the aquifer. Review what you know about how the first pump was created.

1. Open up your version of the base model. **REMIX** and rename the project with ***your name your partner's name*** **mod3**"

2. Use the **Model Design Form** to plan your modification.

3. Get coding!

4. Test your model to make sure it is working correctly.

Here are a few tips:

- Remember to use the **driver** and **navigator** roles and switch with your programming partner regularly.
- Ask for help if you need it.

When you are done, upload and share your project.
Don't forget to put both partners' names in the project title.

Lesson 3 - Student Activity #2 Guide

Running an Experiment

In this activity you will use your new model to run an experiment.

1. Use the **Experimental Design Form** to plan your experiment.

2. Record your data and analyze your results.

Here is a tip:

- You can write up your results on a separate piece of paper. You can use graphs and/or tables to help you.

Lesson 4 - Student Activity #1 Guide

Computational Science and Designing Your Project

Design your project
In this activity you and your programming partner will come up with your own model based on the Water Pumping base model.

1. Open up your version of the base model. **REMIX** and rename to "Water Pumping ***your name your partner's name*** **NEW**"

2. Use the **Project Design Form** to plan your modeling.

Here is a tip:

- Try to think of things to put in your model that will help you answer your question, but keep it simple!

When you are done, move on to Activity #2.

Don't forget to put both partners' names in the project title.

Lesson 4 - Student Activity #2 Guide

Designing and Developing Your Model

Code your model
In this activity you and your programming partner will put your planning into practice and you will make your new model.

1. Open up your model ("Water Pumping ***your name your partner's name*** **NEW**")

2. Use the **Project Design Form** to guide you as you take turns **driving** and **navigating**.

Here is a tip:

- You can copy and paste code from other models to help you work more quickly.

When you are done, upload and share your project.
Don't forget to put both partners' names in the project title.

Lesson 5 - Student Activity #1 Guide

Completing and Debugging Your Code

Debug your model
In this activity you and your programming partner trace execution of your model and use **debugging** to fix any issues you find.

1. Open up your model ("Water Pumping ***your name your partner's name*** **NEW**")

2. Test your model out and fix any issues you find.

Here is a tip:

- Try to look at the code and model in sections to avoid getting overwhelmed.

When you are done, move on to Activity 2.
Don't forget to put both partners' names in the project title.

Lesson 5 - Student Activity #2 Guide

Running Experiments

In this activity, you and your programming partner will use your new model to run experiments.

1. Open up your model ("Water Pumping ***your name your partner's name*** **NEW**")

2. Use your **Project Design Form** to guide your experimentation.

3. Record your results and perform **Data Analysis** on them. Record your conclusions.

Here is a tip:

- You may need to modify your experimental design, now that you have finished your model.

When you are done, start preparing your results for a presentation.
Don't forget to put both partners' names in the project title.

Student Activity Guide

Preparing Presentations

In this activity, you and your programming partner will prepare a ten-minute presentation using tools like slide presentation and a projector. Follow the guidelines below:

1. State the question you were seeking to answer or the problem that you were studying.
2. Tell us about any background research you did on the topic.
3. Tell us about your model (what's included and what was left out).
4. Tell us about your experimental design.
5. Show your model running and how you collected data.
6. Show any collected data and analysis.
7. Tell us about any relationships you noticed between variables that help you understand or predict the phenomenon.
8. Summarize your findings; what was the outcome of running your experiments?
9. Do you think you learned anything about the real world?
10. Show us a piece of code you are proud of.
11. Allow time for questions and answers.

Here is a tip:

- Take turns presenting different sections. Plan on going between the presentation (slide presentation, for example) and the model.

When you are done, practice your presentation with your partner.
Don't forget to put both partners' names in the project title.

Lesson 1 - Student Activity #1 Guide

Papercatchers

Name: Date:

Instructions:
1) Copy the data collected during the Papercatchers activity into the table below. Note that you may not be able to complete all 10 generations in class.

Generation	Round 1 – no paper		Round 2 - newspaper		Round 3 – sheet paper
	Population size		Population size		Population size
1	1		1		1
2					
3					
4					
5					
6					
7					
8					
9					
10					

2) In this simulation of an ecosystem, what were the independent and dependent variables?

3) Using different colored pens or markers for each Round, create line graphs of the populations over time in generations. Make note of any carrying capacity or limits on growth in each round.

Remember to label your axes.

4) Predict the population sizes in each round (no paper, newspaper, sheet paper) if we could continue for 10 generations. (Remember that each new generation occurred when the current population did the ball toss and either died or brought another person in.)

Generation	Round 1 – no paper		Round 2 - newspaper		Round 3 – sheet paper
…	Population size		Population size		Population size
10					

5) Describe how you made the prediction using mathematical and scientific reasoning.

Lesson 2 - Student Activity #1 Guide

Looking under the Hood

Now we are going to get to know the code that makes up the base model!

1) Open your saved StarLogo Nova Rabbits and Grass base model.

2) Navigate to the code section.

3) Use the **Model Observation Form** as you and your programming partner take turns looking at the code. (Remember to use your **driver** and **navigator** roles and switch roles from time to time.) Complete the form by running the model and looking at the code.

4) Which part of the code have you and your partner been assigned?

5) Write down what the code in your assigned section does.

6) Diagram the program's execution loop.

Lesson 2 - Student Activity #2 Guide

Designing and Running Experiments

In this activity you will use your new model to run an experiment.

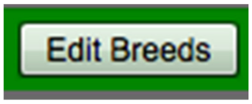
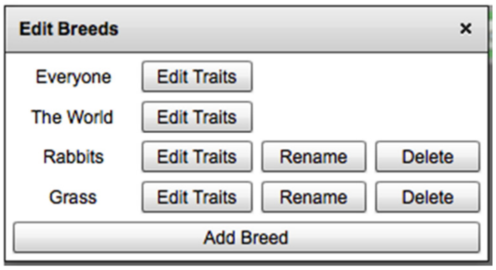
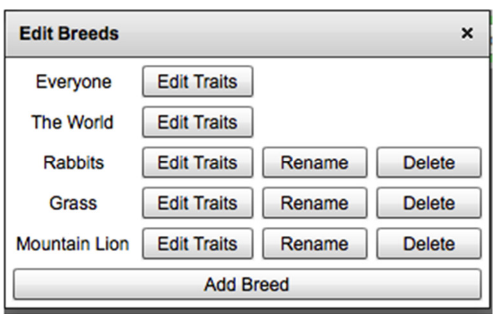
1. Use the **Experimental Design Form** to plan your experiment.

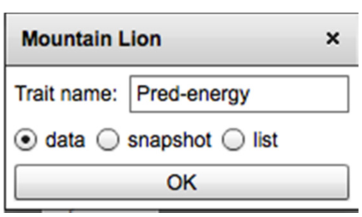
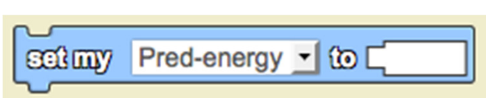
2. Record your data and analyze your results.

Lesson 3 - Student Activity #1 Guide

Adding a Predator

Use the instructions below to add a new breed:

<table>
<tr><td>
</td><td>Use the Edit Breeds button in the Spaceland area to create a new breed.</td></tr>
<tr><td>
</td><td>Click on “Add Breed” to add a new breed.</td></tr>
<tr><td>
</td><td>In the “New Breed” dialog box, give the breed a name and press OK.</td></tr>
<tr><td>
</td><td>You will see the Breed on the next panel. (We added the breed called “Mountain Lion.”)

Next, click Edit Traits next to the new breed label for “Mountain Lion.”</td></tr>
<tr><td>
</td><td>Create a new trait (or variable) for the Mountain Lion breed called “Pred-energy.”</td></tr>
<tr><td>
</td><td>The set my (trait) to (value) block is used to set an agent’s trait to some value. In this case, the Mountain Lion’s Pred-energy is being initialized.</td></tr>
</table>

Lesson 3 - Student Activity #2 Guide

Designing and Running Experiments

In this activity you will use your new model to run an experiment.

1. Use the **Experimental Design Form** to plan your experiment.
2. Record your data and analyze your results.

Lesson 2 - Student Activity #2 Guide

Modifying the Silver Nitrate and Copper Reaction Model

Remix the base model: “silver nitrate and copper reaction” and change its name to include your name and your partner’s name.

Challenge #1:
Modify the code to add or remove some or all of the water molecules in the model.

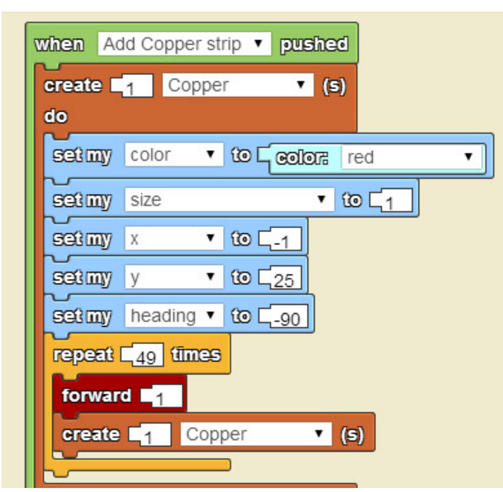
Reminder: Execute your model every time you add a piece of code. Debug as needed. Save often.

Hint: Look at the setup instructions for the water agents.
After you change the code, go up to the SpaceLand interface, save the model and click on “Run Code’ before running the model by clicking on the interface buttons. Debug the model as needed.

Q: Does removing the water molecules affect the execution of the model and real-world representation?
If time allows after you are done, think about what is included in this model and what is missing.

Challenge #2:
Modify the code to move the copper rod somewhere else in the solution keeping the number of agents of copper the same. Please note: The base model gives 150 copper agents to represent 150 copper atoms.

Hint: Look at the setup instructions for the copper agents (section shown below)



After you change the code, go up to the interface and hit “Save” and “Run Code” before executing the model. Debug the code as needed.

What does the command “set my x to” do? What does the command “set my heading to” do? What happens if you change the number of times the loop repeats?

Lesson 3 - Student Activity #1 Guide

Modifying the Model

Your Challenge: Change the code to add copper nitrate agents to represent the missing chemical product from the model.

The reaction modeled is:
Aqueous silver nitrate added to solid copper will react to form solid silver and aqueous copper nitrate.
Chemical reaction: $2\ AgNO_3$ (aq) + Cu (s) ---> $Cu(NO_3)_2$ (aq) + 2 Ag (s)

Start with base model - StarLogo Nova base model: “silver nitrate and copper reaction”
Remix the base model and change its name to include your name and your partner’s name.

Reminder: Save and test your model every time you add a piece of code. Debug if needed. Save often.

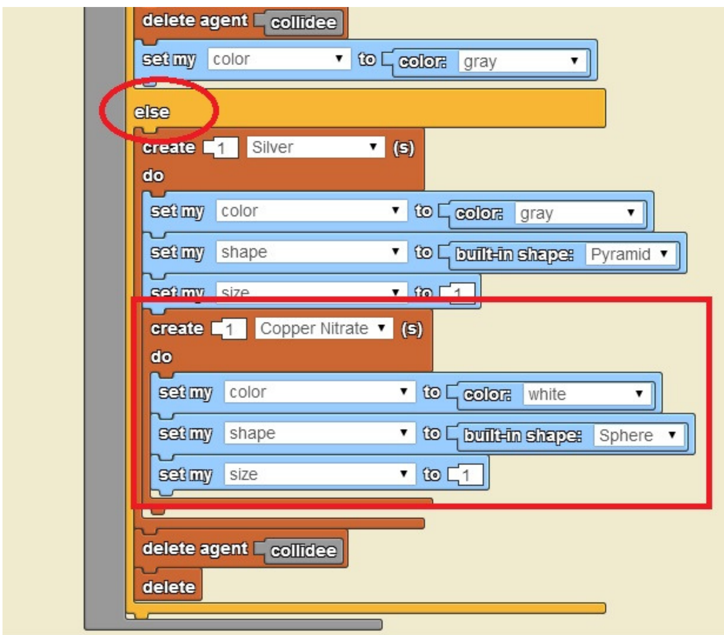
[Hints: (A) Add breed for new agent, (B) create agents inside a collision block, (C) give the agent some movement in water, since it is a soluble ionic compound.]

One solution to the Challenge:

A- Add the copper nitrate agent as a new breed.
Click on edit Breed. Click on add Breed. Name the breed ‘copper nitrate”. Exit edit breed.

Is there a new page/tab in Workspace now that they created copper nitrate breed?

B- When “gray” copper collides with silver nitrate, copper nitrate is created.
Give the copper nitrate agent the following traits:
Shape is sphere, color is white and size is 1.



Where is the correct location of this procedure? Why?

C- Add movement to copper nitrate: copy and paste Wiggle Walk procedure from water page or silver nitrate page to the copper nitrate page.
To copy, with the mouse clicked, drag a box over the piece of code you want to copy, and click on the copy button under the drawers.

Lesson 3 - Student Activity #2 Guide

Running Experiments

Start with the modified base model.

Your Challenge: Run experiments with different initial quantities of copper or silver nitrate and get the amount of silver produced. How much silver is produced in each case? Note when the copper is in excess and when the silver nitrate is in excess.

Use the Experimental Design Form to guide you.
Multiple trials are needed at every different setting of initial amounts used.

Can you predict what amount of silver nitrate is needed to use up all the copper with no excess left of either?

Lesson 4 - Student Activity #1 Guide

Modifying the Model Further

Your Challenge: Alter your model to include the formation of hydrated copper ions. First you will need to create additional breeds Cu(II).H_2O, Cu(II).$2H_2O$, Cu(II).$3H_2O$

Copper in water forms a complex with 3 water molecules which makes the solution slowly turn blue as the complex Cu(II).$3H_2O$ forms.

1. Start with modified model from Lesson 3
2. Remix the base model and change its name to include your name and your partner's name.

Reminder: Save and test your model every time you add a piece of code. Debug if needed. Save often.

3. In the Interface (Spaceland) add 3 additional breeds:

Cu(II).$1H_2O$
Cu(II).$2H_2O$
Cu(II).$3H_2O$

Hint: Refer to Lesson 3 on how to add new breeds.

Check the Workspace area. Did new pages/tabs appear?

4. Change the code using collision blocks to create the following agents with their own traits:

Cu(II).H_2O: Shape is sphere, color is cyan and size is 1. It is created when one copper nitrate agent collides with one water molecule.
Cu(II).$2H_2O$: Shape is sphere, color is sky blue and size is 1. It is created when one Cu(II).H_2O agent collides with one water molecule.
Cu(II).$3H_2O$: Shape is sphere, color is blue and size is 1. It is created when one Cu(II).$2H_2O$ agent collides with one water molecule.

The complex Cu(II).$3H_2O$ is the final ionic product created in the solution turning the solution blue.

Tip: It is possible to drag a selection box around a group of blocks on one page, and "copy and paste" them to a new page. This is a real time-saver.

Reminder: Execute your model every time you add a piece of code. Debug if needed. Save often.

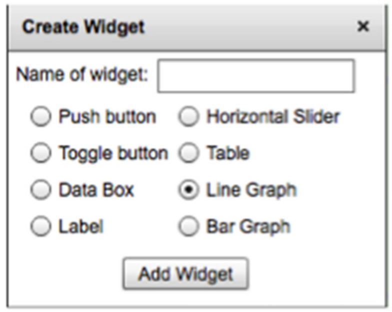
5. These new agents are ions moving about in water.
Set up a Wiggle Walk procedure for each agent on its own page.

Lesson 4 - Student Activity #2 Guide

Running Experiments

Your Challenge: Add instrumentation to your model so you can observe the rate of reaction, then characterize the rate of reaction.

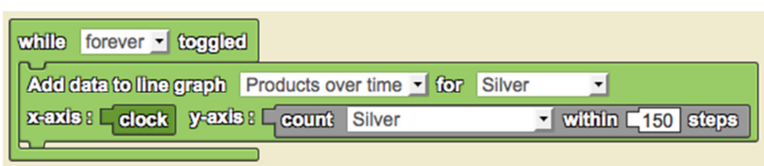
Start with modified model from Activity 1 and add a new widget, a line graph.



On the line graph window, add a new series to the graph by double clicking on New Series and changing the name and line color. For example,

- Create a new series called “Silver” then select gray as its line color.
- Create a new series called “Cu(II).1H2O” then select cyan as its line color.
- Do the same for the other products you want to monitor.

Finally, add the corresponding commands to the World’s page to update the line graph as time advances (clock tick).



Use the experimental design form to design and document your experiment. Try to determine the impact of reactant availability on the rate of reaction (production of products).

What do you observe?
Did the rate of the reaction increase as the number of reactants increased? Or vice versa?