

Coding Guide

A complete example of the base Eco Data program can be viewed and played here:

<https://scratch.mit.edu/projects/472504058>

Introduction

We are going to use Scratch to create a model of interactions between a consumer (rabbits) and a producer (plants). The program will demonstrate a simplified relationship between the number of consumers in a habitat and the number of plants available for consumption.

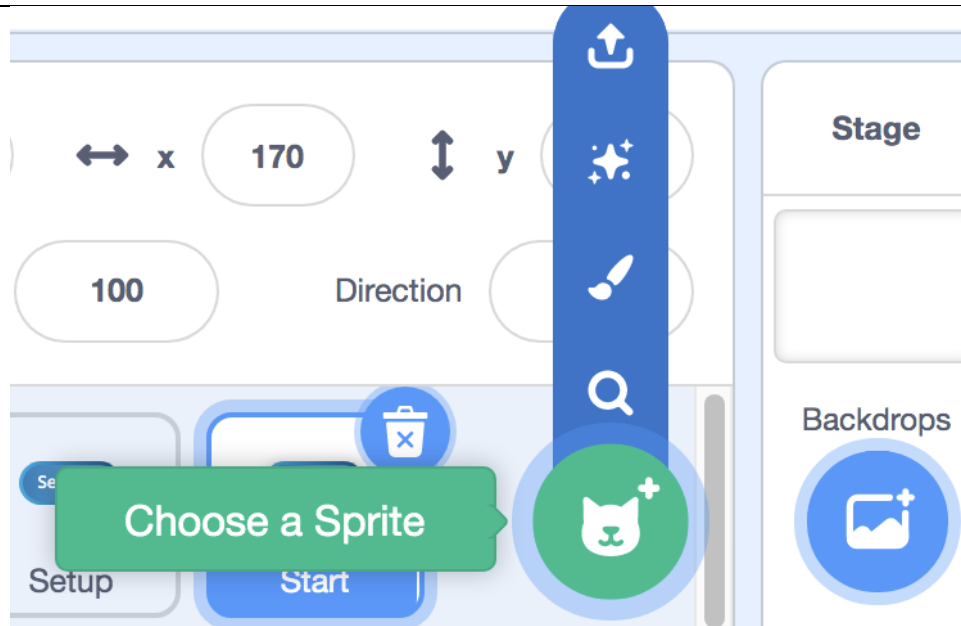
Let's break down the basics of what we want our program to do:

- Set the number of plants that exist in our ecosystem and generate these plants in random positions;
- A number counter reflects the number of plants on the screen;
- Set the number of rabbits that exist in our ecosystem and generate these rabbits in random positions, pointing in random directions;
- A number counter reflects the number of plants on the screen;
- Have the rabbits move continuously;
- When a rabbit sprite touches a plant sprite, the plant sprite disappears (is consumed by the rabbit);
- When a plant sprite is removed, the plant counter value decreases by 1;
- A visible timer counts the number of seconds that pass from the start of the program until all of the plant sprites are consumed by rabbit sprites.

Create Sprites

A sprite is a visual element — like a character or interactive object — that is often programmable.

We are going to create four sprites for our program: one rabbit, one tree, and two buttons. To create a new sprite and select art for that sprite, click the **Choose a Sprite** button located at the bottom right of the screen (shown below). This button will open a library of stock sprite images. In the search bar, search “Rabbit” and select the image of a rabbit.



Repeat these steps three more times, but search “tree” in the sprite library to select a tree sprite, and “button” to select two button sprites (This lesson uses “button1” for both button sprites).

Position the button sprites on the stage and rename each button but changing the name in the sprite field within the Sprite Settings panel (shown below) to distinguish them. In our example code, we have named one button “Setup” and another button “Start”.

Resize the rabbit and trees sprites by changing the value in the size field in the Sprites Settings Panel (shown below). In our example code, we have set the Rabbit sprites to 30 px, and the Trees sprite to 15 px. The buttons will remain 100 px.

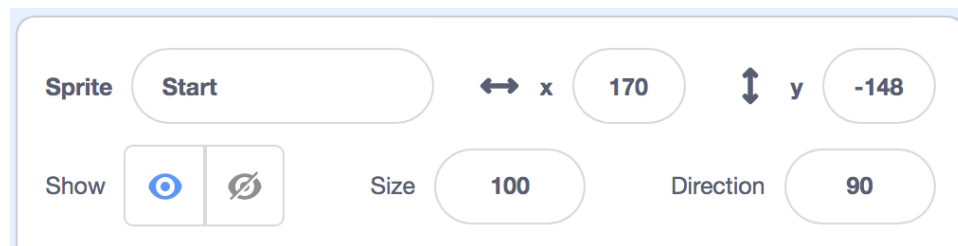
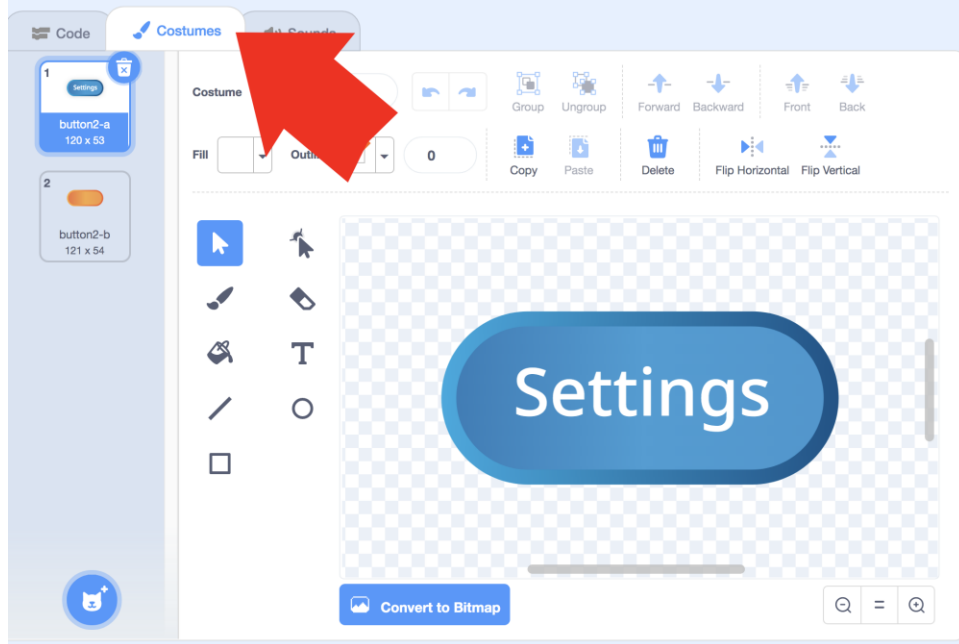


Image Description: The Sprite Settings panel as shown on the Scratch website.

Edit Button Appearance

To add text to the buttons (to better identify them to the user), we will edit them in the **Costumes tab** at the top left of the web page.



Add Text using the **Text** button. Position the text on the button by selecting the text box with your mouse cursor and dragging the text box into position. Change the text colour by selecting the desired colour using the **Fill** dropdown menu.

In this example, we have placed the text “Settings” on the setup button sprite, and “Start Sim” on the start button sprite.



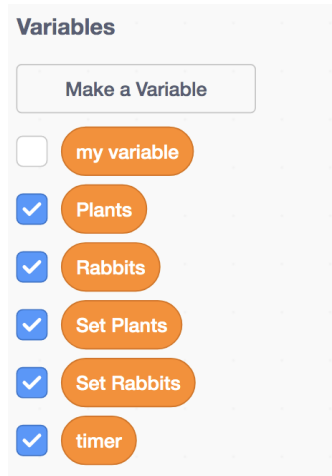
Set Up Variables

Variables store data that we can use in our code. We are going to create five different variables for our program.

In the **Code** tab, select the dark orange **Variables** menu. Click the **Make a Variable** button to create a custom variable block. Type the variable’s name in the **new variable name** text field and select the radio button for **for all sprites**.

- Create a variable named **Plants**. This will be used as a counter to track the number of plants visible on our screen.
- Create a variable named **Rabbits**. This will be used as a counter to track the number of **rabbits** visible on our screen.
- Create a variable named **Set Plants**. This will be used to allow the user to manipulate

- the number of plants that generate at the beginning of the simulation.
- Create a variable named **Set Rabbits**. This will be used to allow the user to manipulate the number of rabbits that generate at the beginning of the simulation.
 - Create a variable named **Timer**. This will be used to measure the number of seconds that pass from when the simulations starts until all plants are consumed.



Position the variable displays on the screen by clicking and dragging them into position using your mouse.

Right click the variable displays for **Set Plants** and **Set Rabbits** and select **slider**. This option will show a slider that allows a user to change the **Set Plants** or **Set Rabbits** variables to a value between 0 and 100.



Program the Setup button sprite

The **setup** button is used to perform two main functions when the button is clicked: to reset the timer to 0 seconds, and to reveal the **Set Rabbits** and **Set Plants** variable slides so that the user can select how many plants and how many rabbits to generate for their simulation.

****Make sure the Setup button sprite is selected in the sprite panel.**

In the yellow **Events** menu, choose the **when this sprite clicked** block and drag it onto coding canvas.

To reset the timer:

- In the dark orange **Variables** menu, choose the **set my variable to 0** block and connect it directly below the **when this sprite clicked** block. Using the drop-down menu on the **set my variable to 0** block, select the variable **Timer**.

To show the sliders:

- In the dark orange **Variables** menu, choose the **show variable [my variable]** and connect it below the **set my variable to 0** block. Using the drop-down menu on the **show variable** block, select the variable **Set Plants**.
- In the dark orange **Variables** menu, choose the **show variable [my variable]** and connect it below the previous **show variable** block. Using the drop-down menu on the new **show variable** block, select the variable **Set Rabbits**.

The complete program for the Setup button sprite should look like this:



Program the Start button sprite:

****Make sure that you have selected the Start sprite in the sprite panel.**

The **start** button is used to launch the timer when the simulation starts. We also want the **Set Plants** and **Set Rabbits** sliders to disappear when the **start** button is pressed.

To hide the sliders:

- In the yellow **Events** menu, choose the **when this sprite clicked** block and drag it onto the coding canvas.
- In the dark orange **Variables** menu, choose the **hide variable [my variable]** and connect it below the **when this sprite clicked** block. Using the drop-down menu on the **hide variable** block, select the variable **Set Plants**.
- In the dark orange **Variables** menu, choose the **hide variable [my variable]** and connect it below the previous **hide variable** block. Using the drop-down menu on the new **hide variable** block, select the variable **Set Rabbits**.
- In the yellow **Events** menu, choose the **broadcast new message** block and place it directly below the **hide variable [Set Rabbits]** block. In the drop-down menu, select **new message** and in the **write new message** field, type “Place Plants”. The broadcast function allows one program to send a signal that triggers another program to run. We will revisit the “Place Plants” broadcast when we build the program for the Trees sprite.

The complete program should look like this:



To program the timer:

- In the yellow **Events** menu, choose the **when I receive new message** block and drag it onto the coding canvas. Using the drop-down menu on the block, select **new message** and in the **write new message** field, type “Start Sim”.

This block is another block that uses the broadcast message function. Instead of sending out the signal, it waits to receive a signal before running the rest of the program connected to this block. We will build the program that sends out the “Start Sim” signal in the code for the Rabbit sprite.

- In the dark orange **Variables** menu, choose the **set [my variable to 0]** block and place it directly under the **When I receive [Start Sim]** block. Using the dropdown menu on this block, select the **timer** variable. This will reset the timer to 0 seconds every time we start a new simulation.
- In the light orange **Control** menu, choose the **repeat until** block and connect it directly below the **set [my variable to 0]** block. This block will act as a loop.
- We want the timer to run until the Plants counter equals 0 (there are no more plants

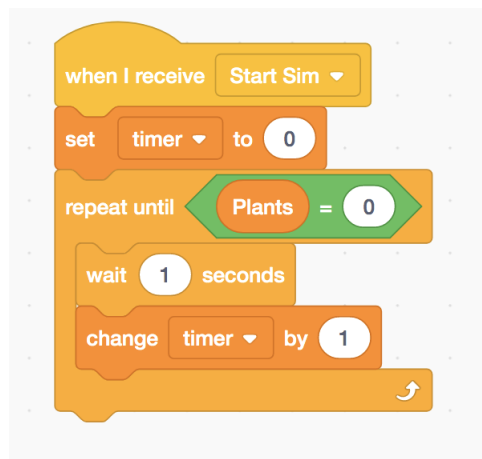
visible on the screen). To do this:

- find the **[blank] = [blank]** block in the green **Operators** menu and drag it on top of the shaded hexagonal space on the **repeat until** block. The shaded space will expand to accommodate the green block.
- In the dark orange **variables** menu, choose the **Plants** block and drag it on top of the first blank space of the **[blank] = [blank]** block. In the second space, type the value 0.

Every block that is placed inside the open “mouth” of this loop will repeat until the **Plants** variable counter reaches 0.

- In the light orange **Control** menu, choose the **wait 1 seconds** block and place it inside the “mouth” of the **repeat until [Plants] = 0** block.
- In the dark orange **Variables** menu, choose the **change my variable by 1** block and place it inside the “mouth” of the **repeat until [Plants] = 0** block, connected directly below the **wait 1 seconds** block. Use the drop-down menu on the **change my variable by 1** block to choose the **timer** variable. This loop will increase the timer counter value by 1 (second) every time 1 second passes.

The completed program should look like this:



Now we’re ready to code our Rabbit and Trees sprites.

Program Trees sprite

*Make sure that you have selected the **Trees** sprite in the sprite panel.

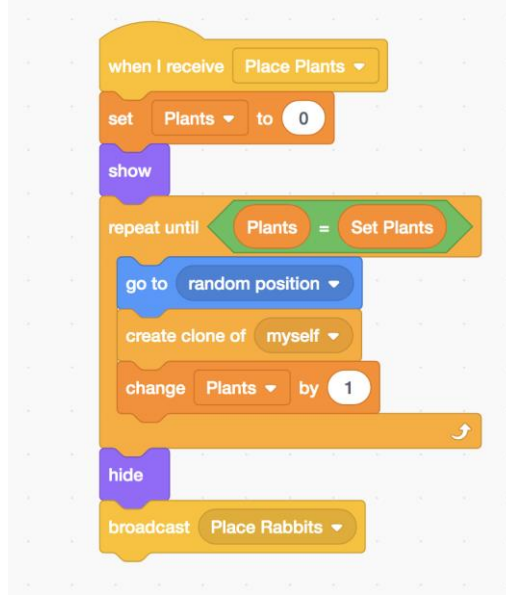
The program for the trees sprite has two main functions:

- To generate and place as many visible plants in random positions on-screen as the user indicates using the **Set Plants** slider (e.g., if the user sets the slider to 10, the program should generate 10 visible plants on-screen); and
- To make plants disappear when touched by a rabbit sprite (and for the corresponding **Plants** counter to update to reflect the number of visible plants left on-screen).

Program Plant Generator

- In the yellow **Control** menu, select the **when I receive new message** and drag it onto the programming canvas. Using the drop-down menu on this block, select the message **Place Plants**. This is the program that will wait to receive the **Place Plants** signal that was broadcast by the program that we built for the Start button sprite. This program will not be triggered to begin until it has received that signal.
- In the dark orange **Variables** menu, choose the **set my variable to 0** and connect it directly below the **when I receive Place Plants** block. Use the drop-down menu to select the variable **Plants**. This block will ensure that we always reset our **Plants** counter to 0 before we generate our plants.
- In the purple **Looks** menu, choose the **Show** block and connect it directly below the **set Plants to 0** block. This will ensure that the Tree sprite is not hidden.
- In the light orange **Control** menu, select the **repeat until** block and connect it directly below the **Show** block.
- In the green **Operators** menu, choose the **[blank] = [blank]** block and drag it on top of the shaded hexagon on the **repeat until** block.
- In the dark orange **Variables** menu, choose the **Plants** block and place it in the first white space on the green **[blank] = [blank]** block. Choose the **Set Plants** block and place it in the second white space on the green **[blank] = [blank]** block.
 - This loop will repeat a program that will generate copies of our tree sprite (called **clones**) until the number of plants visible on-screen (represented by the variable **Plants**) matches the number of plants selected by the user (using the **Set Plants** variable).
- For the inside of our **repeat until Plants = Set Plants** loop:
 - In the blue **Motion** menu, choose the **go to random position** block and place it inside the loop. This will make sure that every new plant that is generated on-screen is placed in a random position.
 - In the light orange **Control** menu, choose the **create clone of myself** block and place it inside the loop beneath the **go to random position** block. This block will make copies (clones) of the sprite.
 - In the dark orange **Variables** menu, choose the **change my variable by 1**. Use the drop-down menu to select **Plants**.
- In the purple **Looks** menu, choose the **Hide** block and connect it below the **repeat until Plants = Set Plants** loop (i.e., this part of the program exists *outside* of the loop). This block will hide the original tree sprite (otherwise our count will be one more than the number of plants that we selected with the **Set Plants** slider).
- In the yellow **Events** menu, choose the **broadcast new message** block. Using the dropdown menu, select “new message”. In the **write new message** field, type “Place Rabbits”. This message will trigger a program that we will build to generate rabbits in the Rabbits sprite code.

Your program should look like this:

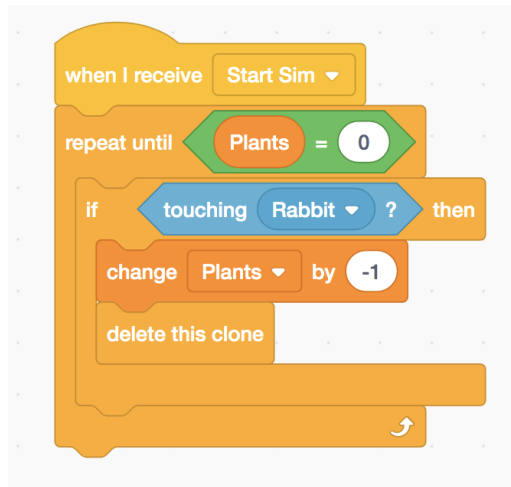


Program Plants to Disappear When Consumed

- In the yellow **Events** menu, choose the **when I receive new message** block and drag it onto the programming canvas to start a new program. Use the drop-down menu to select “Start Sim” from the available options. This will run the program when the “Start Sim” signal is broadcast.
- In the light orange **Control** menu, choose the **repeat until** loop block. Connect it directly below the **when I receive Start Sim** block.
 - In the green **Operators** menu, choose the **[blank] = [blank]** block and drag it on top of the shaded hexagon on the **repeat until** block.
 - In the dark orange **Variables** menu, choose the **Plants** block and place it in the first white space on the green **[blank] = [blank]** block. In the second blank space, type the value 0. This means that this loop will repeat until the **Plants** counter reaches zero.
- In the light orange **Control** menu, choose the **if [blank] then** conditional block. Nest it inside the **repeat until Plants = 0** loop.
 - In the light blue **sensing** menu, choose the **touching mouse pointer?** block and drag it onto the shaded hexagonal blank space on the **if then** conditional block. Use the drop-down menu on the **touching mouse pointer?** block to select the option “Rabbits”. Any code blocks that we place inside this **if touching Rabbits? Then** conditional block will be executed *only* when a Tree sprite is touching a Rabbit sprite.
 - In the dark orange **Variables** menu, choose the **change my variable by 1** block and place it inside the conditional block. Use the dropdown menu to select **Plants** and change the value on this block to -1. This will decrease the value on the Plants counter by 1 every time a plant is consumed.
 - In the light orange **Control** menu, choose the **delete this clone** block and place it inside the conditional block, directly below the **change Plants by -1** block. This will remove the image of the plant that has been

consumed.

The completed program should look like this:

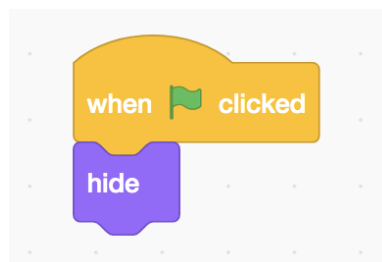


Hide Trees Sprite at Program Start

You'll notice that with our current program, we always have a single visible tree sprite on-screen, even when we are starting a fresh simulation. To correct this, we will create one more program that will hide the original, non-clone sprite.

- In the yellow **Events** menu, choose the **when green flag clicked** block and drag it into the programming canvas to start a new program.
- In the purple **Looks** menu, choose the **hide** block and connect it to the **when green flag clicked** block.

The completed program should look like this:



We are now ready to move on to programming our final sprite: the Rabbit.

Program Rabbit Sprite

*Make sure that you have selected the **Rabbits** sprite in the sprite panel.

The program for the Rabbits sprite has two main functions:

- To generate and place as many visible rabbits in random positions and oriented toward random directions on-screen as the user indicates using the **Set Rabbits** slider (e.g., if

the user sets the slider to 10, the program should generate 10 visible rabbits on-screen); and

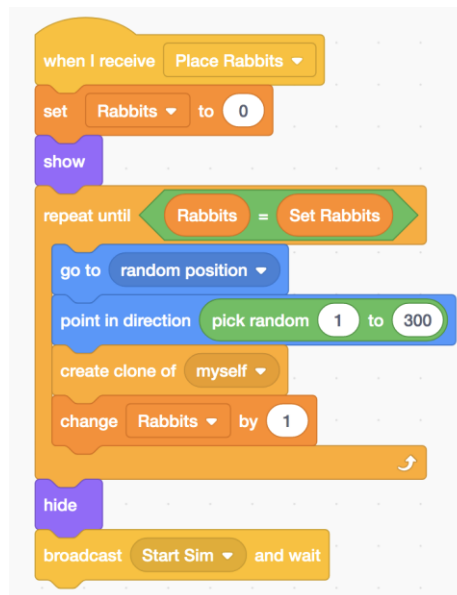
- To make rabbits move continuously.

Program Rabbits Sprite Generator

- In the yellow **Control** menu, select the **when I receive new message** and drag it onto the programming canvas. Using the drop-down menu on this block, select the message **Place Rabbits**. This is the program that will wait to receive the **Place Rabbits** signal that was broadcast by the program that we built for the Start button sprite. This program will not be triggered to begin until it has received that signal.
- In the dark orange **Variables** menu, choose the **set my variable to 0** and connect it directly below the **when I receive Place Rabbits** block. Use the drop-down menu to select the variable Rabbits. This block will ensure that we always reset our Rabbits counter to 0 before we generate our plants.
- In the purple **Looks** menu, choose the **Show** block and connect it directly below the **set Rabbits to 0** block. This will ensure that the Rabbit sprite is not hidden.
- In the light orange **Control** menu, select the **repeat until** block and connect it directly below the **Show** block.
- In the green **Operators** menu, choose the **[blank] = [blank]** block and drag it on top of the shaded hexagon on the **repeat until** block.
- In the dark orange **Variables** menu, choose the **Rabbits** block and place it in the first white space on the green **[blank] = [blank]** block. Choose the **Set Rabbits** block and place it in the second white space on the green **[blank] = [blank]** block.
 - This loop will repeat a program that will generate copies of our rabbit sprite (called **clones**) until the number of rabbits visible on-screen (represented by the variable **Rabbits**) matches the number of rabbits selected by the user (using the **Set Rabbits** variable).
- For the inside of our **repeat until Rabbits = Set Rabbits** loop:
 - In the blue **Motion** menu, choose the **go to random position** block and place it inside the loop. This will make sure that every new plant that is generated on-screen is placed in a random position.
 - In the blue **Motion** menu, choose the **point in direction 90** block and place it directly below the **go to random position** block.
 - In the green **Operators** menu, choose the **pick random 1 to 10** block and drag it on top of the value (90) on the **point in direction 90** block. Adjust the values on the **pick random 1 to 10** to instead read **pick random 1 to 300**. This will generate rabbits oriented toward directions anywhere between 1 and 300°.
 - In the light orange **Control** menu, choose the **create clone of myself** block and place it inside the loop beneath the **point in direction pick random 1 to 300** block. This block will make copies (clones) of the sprite.
 - In the dark orange **Variables** menu, choose the **change my variable by 1**. Use the drop-down menu to select Rabbits.
- In the purple **Looks** menu, choose the **Hide** block and connect it below the **repeat until Rabbits = Set Rabbits** loop (i.e., this part of the program exists *outside* of the loop). This block will hide the original rabbit sprite (otherwise our count will be one more than the number of rabbits that we selected with the **Set Rabbits** slider).
- In the yellow **Events** menu, choose the **broadcast new message and wait** block. Using

the dropdown menu, select “Start Sim”.

Your program should look like this:



Program Rabbit Movement

This program will keep the rabbits moving continuously forever using a loop.

- In the yellow **Events** menu, choose the **when I receive new message** block and drag it onto the programming canvas to start a new program. Use the drop-down menu to select “Start Sim” from the available options. This will run the program when the “Start Sim” signal is broadcast.
- In the light orange **Control** menu, choose the **repeat forever** loop block and connect it directly below the **when I receive “Start Sim”** block.
 - In the blue **Motion** menu, choose the **move 10 steps** block and nest it inside the **repeat forever** loop.
 - Change the value on the **move 10 steps** block to 3 steps so that the rabbits move more slowly (you can adjust this value later while testing the program if you would like to see how having the rabbits move more quickly or more slowly affects the program).
 - In the blue **Motion** menu, choose the **if on edge, bounce** block and nest it inside the repeat forever loop, below the **move 3 steps** block. This block will bounce the rabbit sprite back onto the screen if it reaches the edge (instead of traveling off the visible screen).

The completed program should look like this:

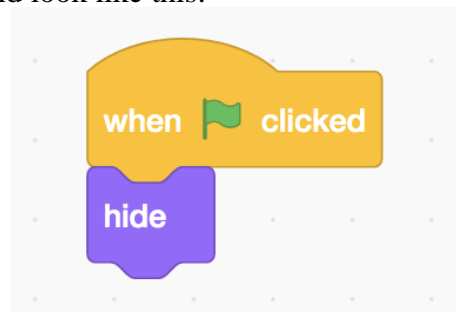


Hide Rabbits Sprite at Program Start

As with the Trees sprite, you'll notice that with our current program, we always have a single visible rabbit sprite on-screen, even when we are starting a fresh simulation. To correct this, we will create one more program that will hide the original, non-clone sprite.

- In the yellow **Events** menu, choose the **when green flag clicked** block and drag it into the programming canvas to start a new program.
- In the purple **Looks** menu, choose the **hide** block and connect it to the **when green flag clicked** block.

The completed program should look like this:



Encourage students to test this completed simulation program by selecting different **Set Plants** and **Set Rabbits** values on the slider and observing how long it takes (in seconds on the **timer**) it takes for the rabbits to consume all of the plants. Students may also choose to alter the parameters of some blocks, such as the distance (number of steps) that the rabbits travel — higher values will make the rabbits travel faster.

Opportunities for Extension: Add Complexity

Once we have our basic ecosystem, we can introduce other factors to increase complexity of relationships and interactions. These factors can include:

- Plant growth — have plant sprites that haven't been consumed produce more plant sprites over time
- Introducing a continuously moving predator sprite that consumes rabbit sprites;

Note that the sample code below describes one method for adding each of these factors to the model. Your students may find other ways to achieve coding objectives, using different blocks or connecting the blocks in different ways. Some programs may be more efficient than others, but as long as the end goal is accomplished, there is no wrong way to get there.

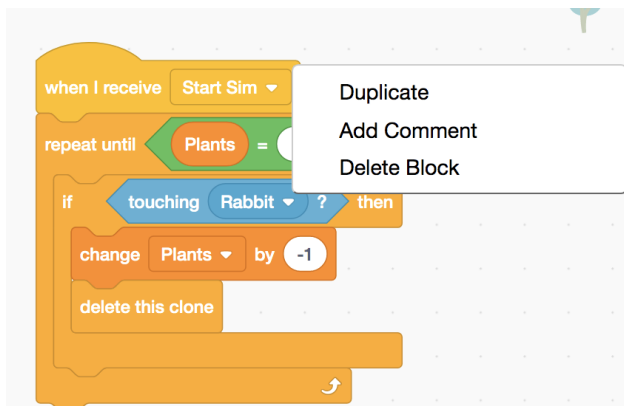
1) Add Plant Growth:

You can view this version of the program in Scratch here: <https://scratch.mit.edu/projects/475366790/>

****Make sure that you have selected the Plants sprite selected in the sprites panel.**

We will create two new programs for the Plants sprite: one that will generate new plant clones every three seconds, and one that stop the program when our Plants counter reaches zero.

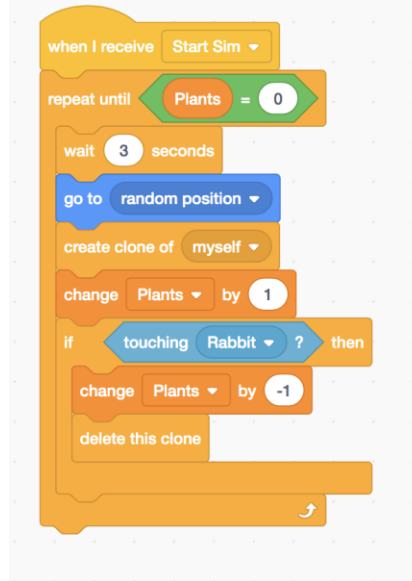
To create the new plant generator, right-click the existing program that includes the **if touching Rabbits?** Conditional statement block (which codes what happens when a plant is consumed by a rabbit) and select **Duplicate**. We will need our new plant clones to behave in the same when when they are touched by a rabbit sprite.



Now let's add code to grow new plants.

- In the orange **Control** menu, select the **wait 1 seconds** block and place it inside the **repeat until Plants = 0** loop block but above the **if touching Rabbit? then** conditional statement block. Change the value of this block to 3 seconds.
- Duplicate three blocks (**go to random position**, **create clone of myself**, and **change plants by 1**) from the existing plant generation program (which begins with the **When I receive [Place Plants]** block). Place these three blocks directly below the **wait 3 seconds** block (above the **if touching Rabbit? then** conditional statement block.)

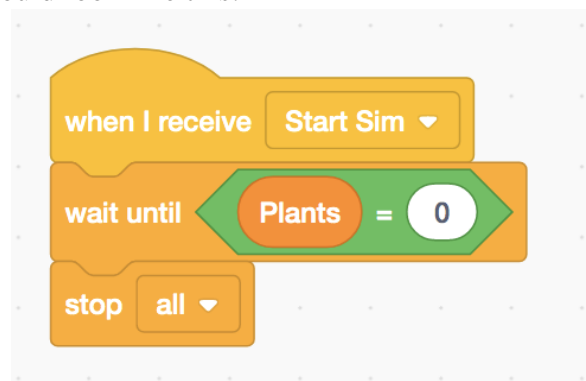
Once these changes are made, the new program should look like this:



Now let's create a simple program that will stop our simulation when all of the plants are consumed.

- In the yellow **Events** menu, select the **When I receive new message** block and drag it into the programming canvas to start a new program. Use the drop-down menu to select the message **Start Sim**.
- In the orange **Control** menu, choose the **wait until [blank]** block and connect it to the new **When I receive Start Sim** block.
- Duplicate the green **Plants = 0** blocks by right-clicking the blocks in another program and selecting **Duplicate**. Drag these duplicated blocks onto the shaded hexagon of the **wait until [blank]** block.
- In the orange **Control** menu, choose the **stop all** block and connect it to the **wait until Plants = 0** block.

Your completed program should look like this:

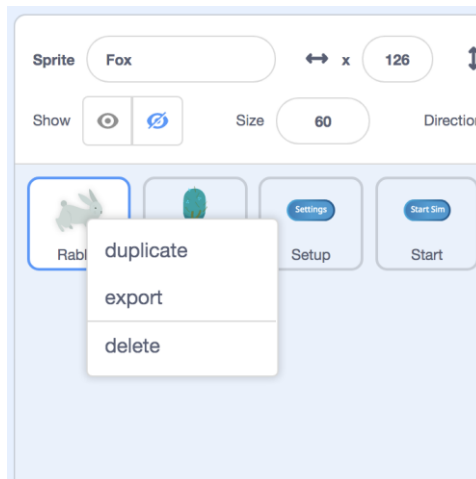


Run the program to test these modifications.

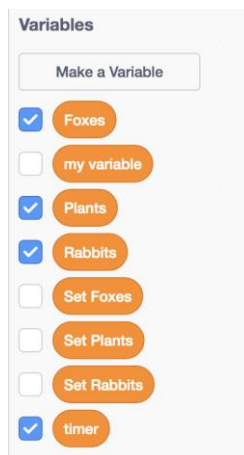
2) Add a Predator:

You can view this version of the program in Scratch here: <https://scratch.mit.edu/projects/475373676>

To add a predator, we will have to create a new sprite to represent the predator. The easiest way to do this, is to right-click the Rabbit sprite in the Sprite Panel and select **Duplicate**. The predator program will be very similar to the program that we built for the Rabbit program, so we can just start with the Rabbit program and make modifications.

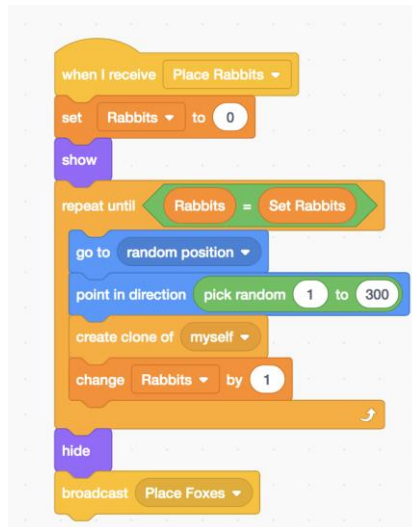


- Rename your new sprite **Foxes**. Apply a new image to this Sprite by opening the **Choose a Sprite** library and searching “Fox” in the search bar.
- Resize the fox sprite by changing the value in the Size field of the sprites panel to 60.
- In the dark orange **Variables** menu, create two new variables for the Fox: Foxes (to build a Fox counter) and Set Foxes (so that the user can set the initial number of foxes with a slider)
- Right-click the Set Foxes counter in the program stage and select **slider**.
- Uncheck the box for Set Foxes variable in the Variables menu to hide the slider.



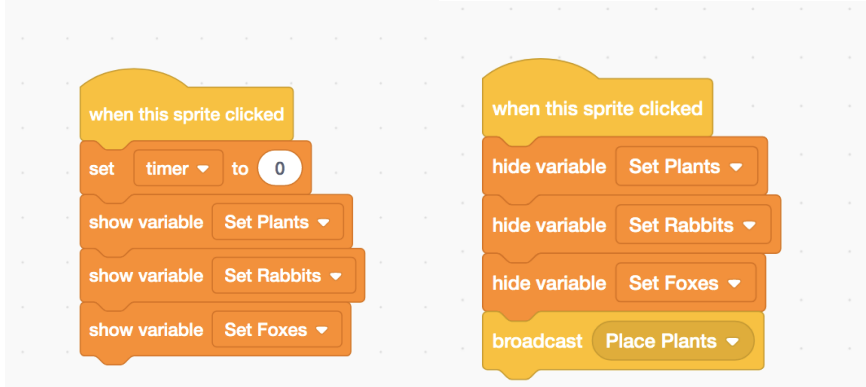
Now we want to set up the sequence of broadcast-triggered programs so that the foxes are generated on-screen after the rabbits finish generating. Select the Rabbits sprite in the sprites panel.

- The last block of the program that begins with the yellow **When I receive Place Rabbits** block is a yellow **broadcast Start Sim** block. Using the dropdown list on the **broadcast Start Sim** block and select **Place Foxes**.



Next, let's make sure our variables for the fox sprite are set up appropriately for when the **Settings** and **Start Sim** buttons are pressed.

- Select the **Setup** sprite in the sprites panel.
- In the dark orange **Variables** menu, select the **show variable** block and connect it directly below the existing **show variable [Set Rabbits]** block. Use the drop-down menu on the block to select the variable **Set Foxes**. This will bring the Set Foxes slider up when the Settings button is clicked (to set the initial number of foxes in the simulation)
- Select the **Start** sprite in the sprites panel.
- In the dark orange **Variables** menu, select the **hide variable** block and connect it directly below the **hide variable [Set Rabbits]** block and above the **broadcast [Place Plants]** block. This will hide the slider when the **Start Sim** button is pressed.



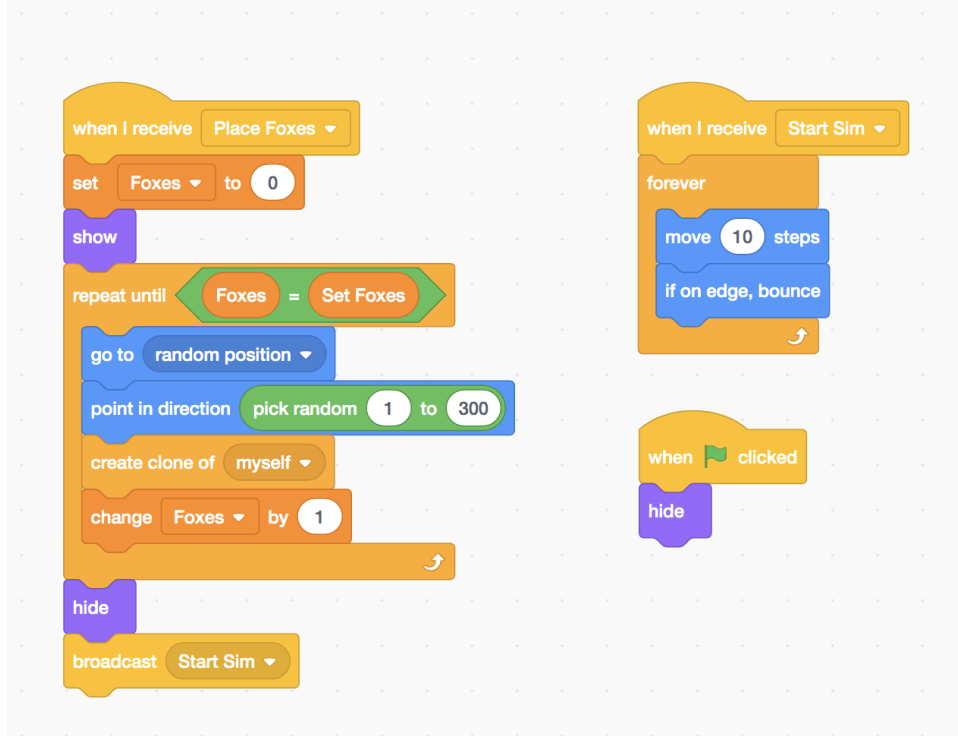
Finally, let's adjust the code that programs the fox sprite's behaviour.

****Make sure that you have selected the Foxes sprite selected in the sprites panel.**

Begin with the program that starts with the **When I receive [Place Rabbits]**. We need to change all references to rabbits to program instead for foxes.

- Using the dropdown menu on this first block (**When I receive [Place Rabbits]**) set the message to **Place Foxes**. Now the foxes will be generated after the rabbits have been generated.
- Change the next block, which reads **Set [Rabbits] to 0** to **Set [Foxes] to 0**, by selecting **Foxes** in that block's dropdown menu.
- In the loop block, **repeat until [Rabbits] = [Set Rabbits]**, we need to change the references to Rabbit variables into Fox variables. In the dark orange **variables** menu, choose the **Foxes** variable block and drag it on top of the **Rabbits** block in the code to replace it. Then, choose the **Set Foxes** variable block and drag it on top of the **Set Rabbits** block in the code to replace it.
- Use the drop-down menu on the **change Rabbits by 1** block to **[Foxes]**.

With these changes, the program should look like this:



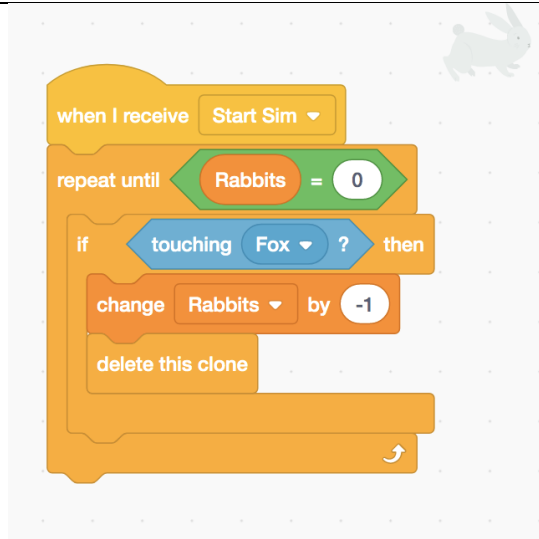
Lastly, we need to add a program that will remove a rabbit when it is consumed by a fox. We will add this program to the Rabbits sprite.

The program that we need to build for this function is the same as the program that we built when a plant is consumed by a rabbit. If you select the Plants variable from the sprites panel, you can right-click the program that we need (the program that contains the **if touching Rabbits then** conditional block) and select **Duplicate** you can drag this duplicated program on top of the **Rabbits** sprite in the sprites panel to transfer the code to the Rabbits sprite.

****Make sure that you have selected the Rabbits sprite selected in the sprites panel.**

- Change the loop that reads **repeat until Plants = 0**. Under the dark orange **variables** menu, select the **rabbits** variable block and drag it on top of the **Plants** variable block to replace it.
- Change the conditional statement so that it reads **if touching [Fox]? then** by using the drop-down menu on the blue **touching ?** Block.
- Change the **Change variable [Plants] by -1** block. Using the drop-down menu on this block, select **Rabbits**.

With these changes, the program should look like this:



Run the program to test these modifications.

Encourage students to experiment with their ecosystem models to introduce different factors than those outlined in these examples, or to experiment with the interactions between these factors.