

 Projects

## Tree life simulator

Create a simulation that shows the impact of land management and deforestation on trees, wildlife, and the environment



### Step 1 Introduction

---

Use Scratch to create a simulation that shows the impact of land management and deforestation on trees, wildlife, and the environment.

This project focuses on three of the United Nations Sustainable Development Goals:

- Responsible Consumption and Production (<https://www.undp.org/sustainable-development-goals#responsible-consumption-and-production>).
- Climate Action (<https://www.undp.org/sustainable-development-goals#climate-action>).
- Life Below Water (<https://www.undp.org/sustainable-development-goals#below-water>).

Thanks to Ecosia (<https://www.ecosia.org>), for their guidance on tree life simulation.

What you will make



#### What you will need

Hardware

- A computer or tablet capable of running Scratch 3

Software

- Scratch 3 (either online (<https://scratch.mit.edu/>) or offline (<https://scratch.mit.edu/download>))

Downloads

- If you are working offline, download the project starter file (<https://rpf.io/p/en/tree-life-simulator-go>).

#### What you will learn

- How to use **clone** blocks and **random** numbers to simulate a natural environment
- How to use **if** operators to control a simulation
- How to program a simulation to help communicate an environmental issue



### Additional information for educators

You can download the completed project here (<https://rpf.io/p/en/tree-life-simulator-get>).

If you need to print this project, please use the printer-friendly version (<https://projects.raspberrypi.org/en/projects/tree-life-simulator/print>).

## Step 2 Grow some trees

---

Did you know that trees cover 30 percent of the Earth's land surface? The way we look after our local trees now, will have an impact on the future of our planet. In this step, you will add code to create an area of trees with a variety of different tree types.

Online: open the starter project (<http://rpf.io/tree-life-simulator-on>) in Scratch.



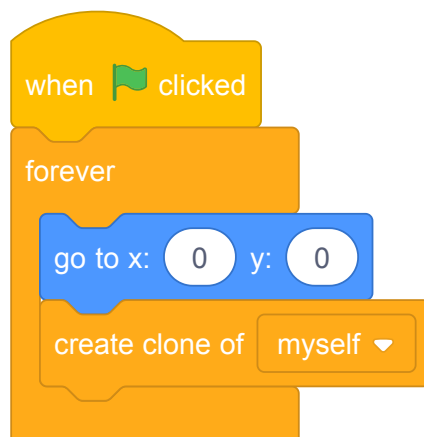
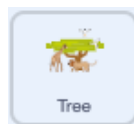
Offline: open the project starter file (<http://rpf.io/p/en/tree-life-simulator-get>) in the Scratch offline editor. If you need to, you can download and install Scratch here (<https://scratch.mit.edu/download>).

You should see a green grassland background.



In Scratch, you can make up to 300 copies of one sprite! This is called cloning. Your task is to add code to the Tree sprite so that it creates clones of itself on the Stage.

Go to the Sprites pane, and click on the Tree sprite. Drag a **when green flag clicked** block and a **forever** block onto your Code area. Within the **forever** block, add a **go to x: 0 y: 0** block and a **create clone of myself** block:



Next, plant the cloned trees in random positions across the Stage.

Add a **pick random** block to both your **x:** and **y:** values to vary where the trees grow. Change the **x:** coordinates to **pick random -150 to 200**. Change the **y:** coordinates to **pick random -120 to 120**:

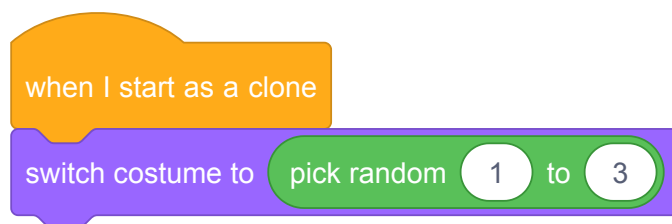
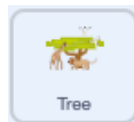


It is important that you plant a variety of trees to house a range of animals, restore the environment, and benefit people. For example, a koala relies on the broadleaf evergreens in Australia, while a lemur in Madagascar needs the deciduous trees that grow on the island.

The Tree sprite has three costumes: tree 1, tree 2, and tree 3. Click on the Costumes tab to see them.

Click back to the Code tab and use the **random** operator to vary the looks of the trees and add variety to the area.

Start a new script with a **when I start as a clone** block. Add a **switch costume to** block below it. Drag a **pick random 1 to 10** block into the **switch costume to** block. Change the values from **1** and **10** to **1** and **3**:

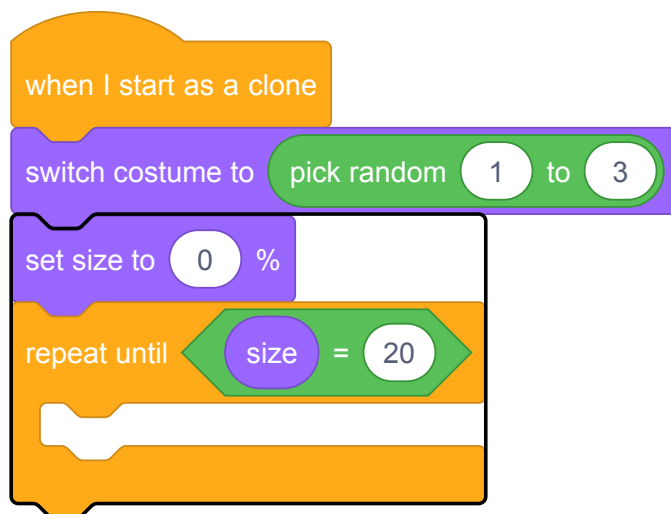


Test your simulation by clicking on the green flag. Make sure you have a variety of trees.



Trees don't just appear full grown, they get bigger over time. You need to set up a **repeat until** loop so you can change the size of the tree as it grows, until its size equals 20 percent.

Get a **set size to 100** (percent) block, but change the value to **0** so that the Tree sprite starts from nothing. Add a **repeat until** block, and drag an equals **=** block inside. Add the condition **size = 20**:



Get the tree to resize and wait as it grows.

Add a **change size by 10** block within the loop and change the value to **1**. Add a **wait 1 seconds** block and change the value to **0.1** so that it changes quickly:



```
when I start as a clone
  switch costume to pick random 1 to 3
  set size to 0 %
  repeat until size = 20
    change size by 1
    wait 0.1 seconds
```

Test your simulation again. Your trees will grow, but a full-grown tree also appears wherever your next clone is growing.



Hide the tree until it starts a new clone.

Add a **hide** block to the start of your **when green flag clicked** script, and a **show** block to the start of your **when I start as a clone** script.

The image shows two Scratch scripts. The first script, titled "when green flag clicked", consists of a "hide" block followed by a "forever" loop. Inside the loop, there is a "go to x: pick random -150 to 200 y: pick random -120 to 120" block and a "create clone of myself" block. The second script, titled "when I start as a clone", consists of a "show" block, a "switch costume to pick random 1 to 3" block, a "set size to 0 %" block, a "repeat until size = 20" loop, and a "change size by 1" block inside the loop. Below the loop is a "wait 0.1 seconds" block.

Test your simulation again. Your trees should now grow like they would in real life.




Save your project

## Step 3 Tree management

---

The rate at which new trees grow depends on a number of factors, such as climate conditions, soil quality, disease, sunlight, and water. In this step, you'll add a slider to demonstrate how managing these conditions can impact the number of trees in the area.

The growing speed will be held in a **variable** called **tree management**.

To create a new variable, click on the **variables** blocks menu. 

Then click on the Make a Variable button.

You can give your **variable** a name. Call it **tree management**.

Your new **variable** is visible on the Stage. Next, create a slider to control the speed in your simulation.

On the Stage, right-click on the **tree management** variable, and a menu will appear. 

Select slider in the menu.

At the moment, the **tree management** range is too wide.

On the Stage, right-click on the tree management slider and select change slider range. 

Change the range to between **0** and **5**.

In the simulation, the tree management slider controls the speed that new trees grow. If you move the slider to the right, it speeds up the growth; if you move it to the left, it slows down the growth.

When the slider is at the right (5), the simulation waits one second to plant a tree, and when it is at the left (0) it waits six seconds to plant a tree.

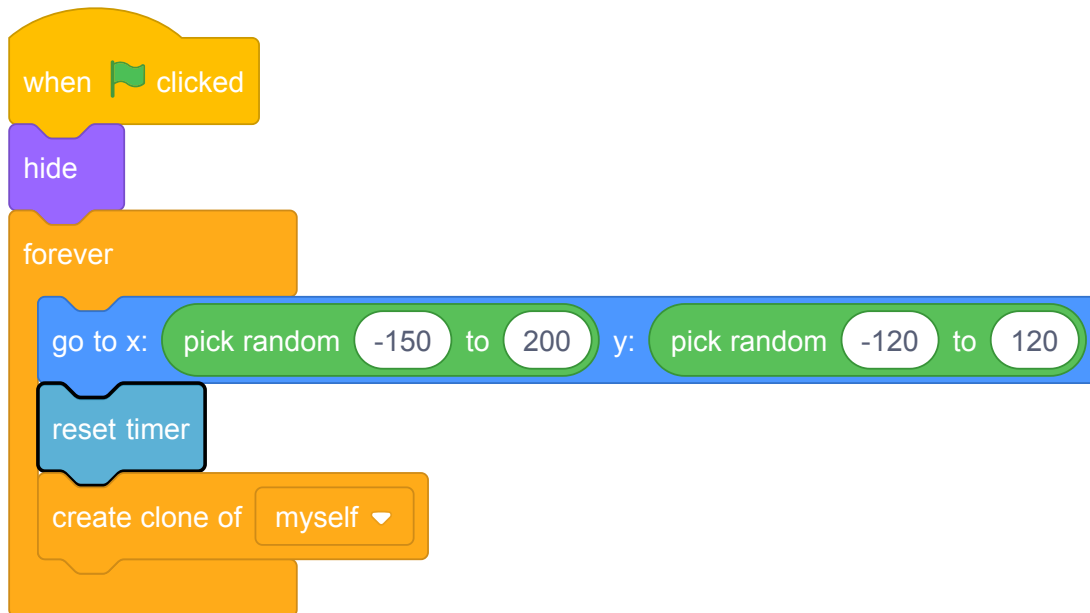
You will add a set of blocks to make sure that the speed that new trees grow is in reaction to a change in the tree management slider.

In this task, you'll see how the **timer** block works. Go to the **Sensing** blocks menu, and click on the checkbox next to the **timer** block to show the timer on the Stage. 

Click on the green flag and you'll notice the timer starts to count up immediately. Click on the checkbox again to hide the timer from view.

Reset the timer to zero each time the flag is clicked.

Insert a `reset timer` block into your `when green flag clicked` script for the Tree sprite, so that the timer resets each time a new tree is cloned: ✓



Create a quick response to the slider so that the user immediately sees the impact of their interaction. Set up a loop that checks the tree management slider value frequently and adjusts the speed before cloning another tree.

Insert a **repeat until** block with the condition **timer > 6 - tree management**. Within the **repeat until** block, add a **wait 1 seconds** block:



```

when green flag clicked
  hide
  forever loop
    go to x: pick random -150 to 200 y: pick random -120 to 120
    reset timer
    repeat until (timer > 6 - tree management)
      wait 1 seconds
    create clone of myself
  
```

Test your simulation again. The tree management slider should control the speed that new trees grow.



Save your project

## Step 4 Count the trees

Scientists monitor trees and wildlife. They record the cover (size), variety, and health of the trees and surrounding areas. By doing this, they can see the true impact of deforestation. Observations can be taken from planes and satellites in the air, as well as sensors and people on the ground. People can use their findings to see where extra measures are needed to protect trees. For example, the construction of fences around a forest to allow trees to thrive, or the employment of more firefighters to help find and stop fires.

In this step, you'll monitor the trees in your simulation through a variable that stores the number of trees and reflects changes to the environment.

Create a new **variable** to store the number of full-grown trees. Call this variable **mature trees**.



Reset the number of trees when the simulation starts to 0.

Underneath the **when green flag clicked** block, set the **mature trees** variable to 0:



```

when green flag clicked
  set mature trees to 0
  hide
  forever loop
    go to x: pick random -150 to 200 y: pick random -120 to 120
    reset timer
    repeat until timer > 6 - tree management
      wait 1 seconds
      create clone of myself
  
```

Make sure your counter records how many trees are created while the simulation runs.

Find your **when I start as a clone** script and add a **change mature trees by 1** block at the end:



```
when I start as a clone
  show
  switch costume to pick random 1 to 3
  set size to 0 %
  repeat until size = 20
    change size by 1
    wait 0.1 seconds
  change mature trees by 1
```

Now you can count your new trees, however not all those trees will survive. The survival rate for new trees is estimated to be 80 percent, which means that 2 trees out of every 10 planted will die of natural causes before they are full grown. You aren't going to reflect this in your simulation, but it's important to know that not all new trees survive naturally.

Mature trees can also be chopped down through human deforestation. To show this, reduce the count of mature trees if a tree has been touched by the Tree Feller sprite.



Add a **wait until** block to the end of your code, and inside this block a **touching Tree Feller** block. Add a **change mature trees by 1** block and change the value to **-1**. Add a **delete this clone** block to remove the tree:



```

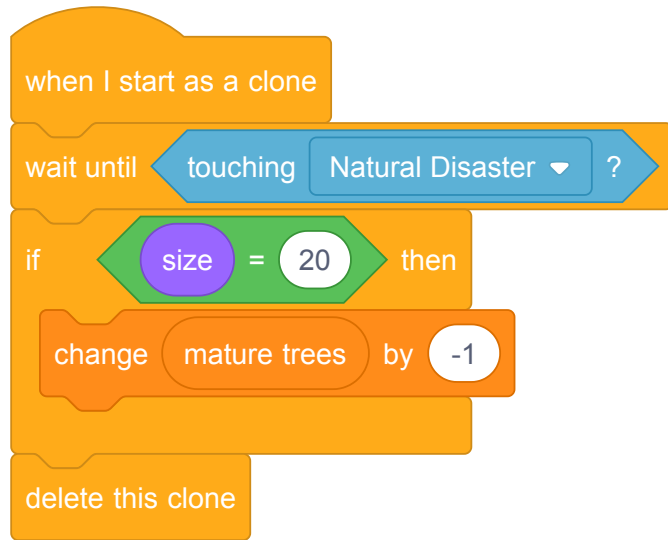
when I start as a clone
  show
  switch costume to pick random 1 to 3
  set size to 0 %
  repeat until size = 20
    change size by 1
    wait 0.1 seconds
  change mature trees by 1
  wait until touching Tree Feller ?
  change mature trees by -1
  delete this clone
  
```

Trees can also die from natural deforestation, whether they are mature or still growing. To show this, reduce your count of trees if a tree has been touched by the Natural Disaster sprite.



Start a new script with a **when I start as a clone** block. Add a **wait until** block to the end of your code, and inside this block a **touching Natural Disaster** block. Add an **if ... then** block with the condition **size = 20**.

Inside the **if ... then** block, add a **change mature trees by 1** block, so that the count of mature trees is only decreased if it were a full-grown tree. Add a **delete this clone** block to remove the tree:



Save your project

## Step 5 Human deforestation

---

There are many reasons why humans chop down trees. The wood might be needed to create furniture, make buildings, or burn as firewood. Alternatively, the land used by the forest might be needed to grow crops, graze cattle, or construct buildings.

In this step, you will code the Tree Feller sprite to get the amount of trees needed, and move through the forest causing human deforestation.

Code the Tree Feller sprite to ask how many trees are needed each time it is clicked.

Click on the Tree Feller sprite and start a new script with a **when this sprite clicked** block. Add an **ask what is your name? and wait** block and change the text to ask **How many trees are needed? (max 20)**: ✓



when this sprite clicked

ask How many trees are needed? (max 20) and wait

Next, you need a variable to hold the input from asking the question **How many trees are needed? (max 20)**.

Create a new **variable** and call it **trees needed**. ✓

Each time someone submits a request for more trees, the Tree Feller sprite should add that number to the amount of trees needed.

To store the answer to your question in the **trees needed** variable, add a **change trees needed by 1** block and drag an **answer** block into where it says **1**: ✓



when this sprite clicked

ask How many trees are needed? and wait

change trees needed ▼ by answer

As trees are big and heavy, the maximum number that can be requested should be set to 20.

Add an **if ... then** block to the end of your code with the condition **trees needed greater than 20**. Inside the **if ... then** block, add **set trees needed to 20**:



```

when this sprite clicked
  ask How many trees are needed? and wait
  change trees needed by answer
  if trees needed > 20 then
    set trees needed to 20
  
```

Reset the number of trees needed every time the green flag is clicked to start the simulation, so it clears the variable ready for a new request.

Start a new script with **when green flag clicked** block, and add a **set trees needed to 0** block:



```

when green flag clicked
  set trees needed to 0
  
```

Run your simulation. Click on the Tree Feller sprite and request a number of trees. Enter different numbers to see what happens to the **trees needed** variable each time.



When the Tree Feller sprite touches a mature tree, the number of trees needed will decrease.

Click on the Tree sprite and add **change trees needed by -1** to the script after it **touches Tree Feller**.



Change the condition of the **wait until** block to be **touches Tree Feller** and **trees needed > 0**



```

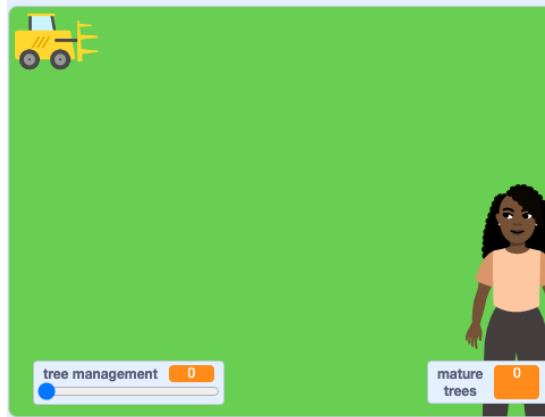
when I start as a clone
  show
  switch costume to pick random 1 to 3
  set size to 0 %
  repeat until size = 20
    change size by 1
    wait 0.1 seconds
  change mature trees by 1
  wait until touching Tree Feller ? and trees needed > 0
  change trees needed by -1
  change mature trees by -1
  delete this clone
  
```

The Stage is looking crowded with so many **variables**. Position them so they appear in better places in your simulation, and hide the ones that don't show values relating to tree management.

Go to the **Variables** blocks menu and untick the **trees needed** variable.

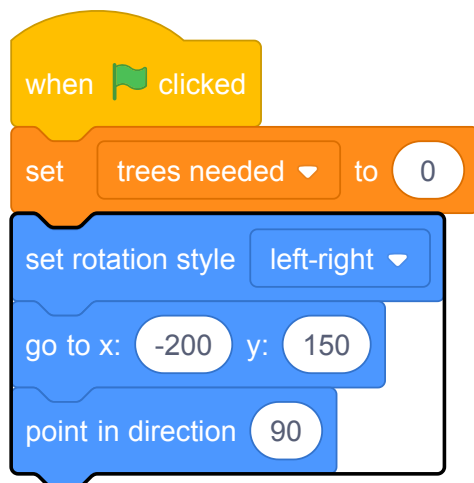
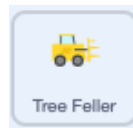


Drag the tree management slider variable and mature trees counter variable to the bottom corners of the Stage.



Set the starting position for the Tree Feller sprite each time the simulation starts.

Click on the Tree Feller sprite and in the **when green flag clicked** script, insert a **set rotation style** **left-right** block so that the Tree Feller sprite does not tip upside down. To get the Tree Feller sprite to start at the top left-hand corner, add a **go to x:0 y:0** block and change the values to **-200** and **150**. Get the sprite to **point in direction 90**:



The Tree Feller sprite will move through the forest from left to right and back again, until it has felled enough trees. Use an operator so the Tree Feller sprite knows it has collected enough trees when the number of trees needed equals zero.

In the `when this sprite clicked` script, add a `repeat until` block and drag an equals `=` block inside. Add the condition `trees needed = 0`.



Within the loop, add a `move 10 steps` block and change the value to 4:



```

when this sprite clicked
  ask How many trees are needed? (max 20) and wait
  change trees needed ▼ by answer
  if trees needed > 20 then
    set trees needed ▼ to 20
  repeat until trees needed = 0
    move 4 steps
  
```

Run your simulation. Click on the Tree Feller sprite and request a number of trees. Does the Tree Feller sprite move through the whole Stage?



At the moment, the Tree Feller sprite only moves across the top of the Stage. You need to make the sprite sense if it is touching the edge of the Stage, so that it knows to change direction.

Add an **if ... then** block with a **touching edge** condition. Inside your **if** loop, add two **Motion** blocks: **turn right by 180 degrees** and **change y by -40**.



```
when this sprite clicked
  ask "How many trees are needed? (max 20)" and wait
  change "trees needed" by answer
  if "trees needed" > 20 then
    set "trees needed" to 20
  repeat until "trees needed" = 0
    move 4 steps
    if "touching edge" ? then
      turn 180 degrees
      change y by -40
```

If the Tree Feller sprite reaches the bottom of the Stage, it needs to start again from the top.

Add another **if ... then** block with a **y position** < -120 condition. Inside your **if ... then** loop, add **go to** **x: -200 y: 150** so that it moves back to the starting position:



```

when this sprite clicked
  ask "How many trees are needed? (max 20)" and wait
  change "trees needed" by "answer"
  if "trees needed" > 20 then
    set "trees needed" to 20
  repeat until "trees needed" = 0
    move 4 steps
    if "touching edge" ? then
      turn 180 degrees
      change y by -40
    if "y position" < -120 then
      go to x: -200 y: 150
  
```

Next, you need to find out when the Tree Feller sprite has felled enough trees. Get the Tree Feller sprite to announce it has finished the requests.

Add a **Looks** block to the end of your code so that the Tree Feller sprite will **say Requests completed. for 2 seconds**:



```

when this sprite clicked
  ask How many trees are needed? (max 20) and wait
  change trees needed by answer
  if trees needed > 20 then
    set trees needed to 20
  repeat until trees needed = 0
    move 4 steps
    if touching edge ? then
      turn 180 degrees
      change y by -40
    if y position < -120 then
      go to x: -200 y: 150
  say Requests completed. for 2 seconds
  
```

Run your simulation. Click on the Tree Feller sprite and request a number of trees. Once the Tree Feller sprite has collected enough trees, it will tell you it has finished.



Once the requests are complete, get the Tree Feller sprite to move to the side of the tree area, ready for the next request.

Add two **Motion** blocks: a **set x to -200** block and a **point in direction 90** block.



```

when this sprite clicked
  ask "How many trees are needed? (max 20)" and wait
  change "trees needed" by "answer"
  if "trees needed" > 20 then
    set "trees needed" to 20
  repeat until "trees needed" = 0
    move 4 steps
    if "touching edge" ? then
      turn 180 degrees
      change y by -40
    if "y position" < -120 then
      go to x: -200 y: 150
  say "Requests completed." for 2 seconds
  set x to -200
  point in direction 90
  
```

Test your simulation. Click on the Tree Feller sprite a few times to request different amounts of wood.



Save your project

## Step 6 Monitor tree levels

---

Trees are extremely important to our quality of life on land. From trees, people get fruits to eat, oils for healthy living, and shelter from the weather. Trees absorb carbon dioxide and help to reduce the rate of CO2 emitted across the globe. They are also home to many different species of animals. Trees help protect our planet!

In this step, you will add code to monitor the number of trees, and add messages to say if the tree area is healthy or too small.

Set the tree management level to zero when the simulation starts.

Click on the Maya sprite. You'll see a couple of blocks are already in the Code area. These tell Maya to start the simulation with some text for the users. Add a **set tree management to 0** block.



```
when green flag clicked
  set tree management to 0
  go to front layer
  say Looking after trees slows down global warming & protects our planet for 4 seconds
```

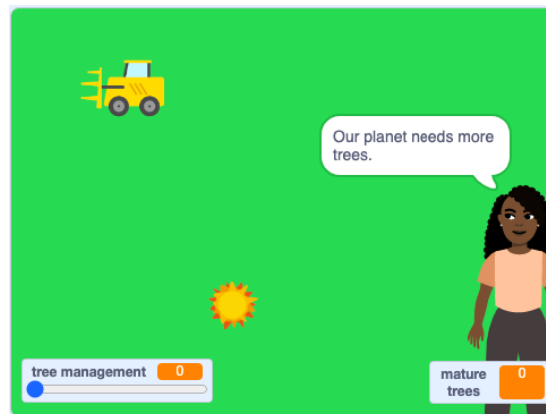
Add code to the simulation so that the Maya sprite says when all the trees have gone.

Add a **forever** block, then insert an **if ... then** block inside. Add the condition, **mature trees = 0**. Add a **say for 2 seconds** block and type: **Our planet needs more trees.**



```

when clicked
  set tree management to 0
  go to front layer
  say Looking after trees slows down global warming & protects our planet. for 4 seconds
  forever
    if mature trees = 0 then
      say Our planet needs more trees. for 2 seconds
  
```



Scientists use their observations to give warnings if CO2 levels are looking dangerous. Add an alert so that the Maya sprite warns if there are not enough trees.

Set up a message to appear when the number of mature trees left is greater than 0 and less than 10.



Add an **if ... then** block inside your **forever** block.

Add some operators for your calculation. First add a **and** block, then a **greater than** block, and a **less than** block. Set up the calculation **mature trees greater than 0 and mature trees less than 10**. Insert **say Global warming is at dangerous levels. for 2 seconds**:



```
when green flag clicked
  set tree management to 0
  go to front layer
  say "Looking after trees slows down global warming & protects our planet." for 4 seconds
  forever loop
    if (mature trees = 0) then
      say "Our planet needs more trees." for 2 seconds
    if (mature trees > 0 and mature trees < 10) then
      say "Global warming is at dangerous levels." for 2 seconds
```

The image shows a Scratch script for a tree life simulator. The script starts with a 'when green flag clicked' event block. It then sets 'tree management' to 0 and goes to the front layer. A 'say' block displays the message 'Looking after trees slows down global warming & protects our planet.' for 4 seconds. A 'forever' loop contains two 'if' blocks. The first 'if' block checks if 'mature trees' is equal to 0, and if true, it says 'Our planet needs more trees.' for 2 seconds. The second 'if' block checks if 'mature trees' is greater than 0 and less than 10, and if true, it says 'Global warming is at dangerous levels.' for 2 seconds.

Add another observation message.

Right-click on the last **if ... then** block, and choose Duplicate from the menu. Insert your duplicated blocks below the last **if ... then** block, and change the values from **0** and **10** to **50** and **60**. Think of a warning message to display in the **say** block:



when clicked

set tree management ▾ to 0

go to front ▾ layer

say Looking after trees slows down global warming & protects our planet. for 4 seconds

forever

if  $\text{mature trees} = 0$  then

say Our planet needs more trees. for 2 seconds

if  $\text{mature trees} > 0$  and  $\text{mature trees} < 10$  then

say Global warming is at dangerous levels. for 2 seconds

if  $\text{mature trees} > 50$  and  $\text{mature trees} < 60$  then

say Animals have many habitats. for 2 seconds

Test your simulation. When the number of trees gets to the range of your **if ... then** blocks, the Maya sprite will say an observational message.



Save your project

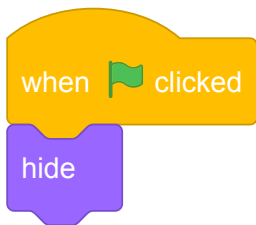
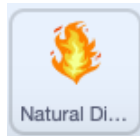
## Step 7 Natural deforestation

---

Deforestation can happen when a natural disaster – such as drought, a forest fire, or a tornado – devastates an area of land. It can also be triggered by human activity, like a campfire, but spread quickly due to natural conditions, like a very dry season. This can be particularly deadly as, in the wide area affected, entire species can disappear along with the destruction of their habitats.

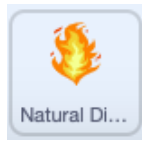
Simulate the impact of natural disasters that randomly move through the forest. Hide the Natural Disaster sprite at the start of the simulation.

Click on the Natural Disaster sprite, then in the Code area add a **when green flag clicked** block, followed by a **hide** block:



The Natural Disaster sprite will move from the top to the bottom of the screen.

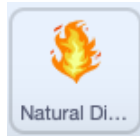
Add a **forever** block to the end of your code. Within it, add a **go to x:0 y:0** block, and change the **y:** value to 200. Add a **show** block, followed by a **glide 1 secs to x:0 y:0**, and change the seconds value to 3 and the **y:** value to -200:



```
when green flag clicked
hide
forever loop
  go to x: 0 y: 200
  show
  glide 3 secs to x: 0 y: -200
```

The Natural Disaster sprite has three costumes: Drought, Forest Fire, and Tornado, as trees can be hit by many types of natural disaster. Add code so that the simulation has multiple disaster types.

Click on the Costumes tab to view the costumes. Click back to the Code tab and add a **hide** block when it reaches the bottom of the screen, followed by a **next costume** block:



```

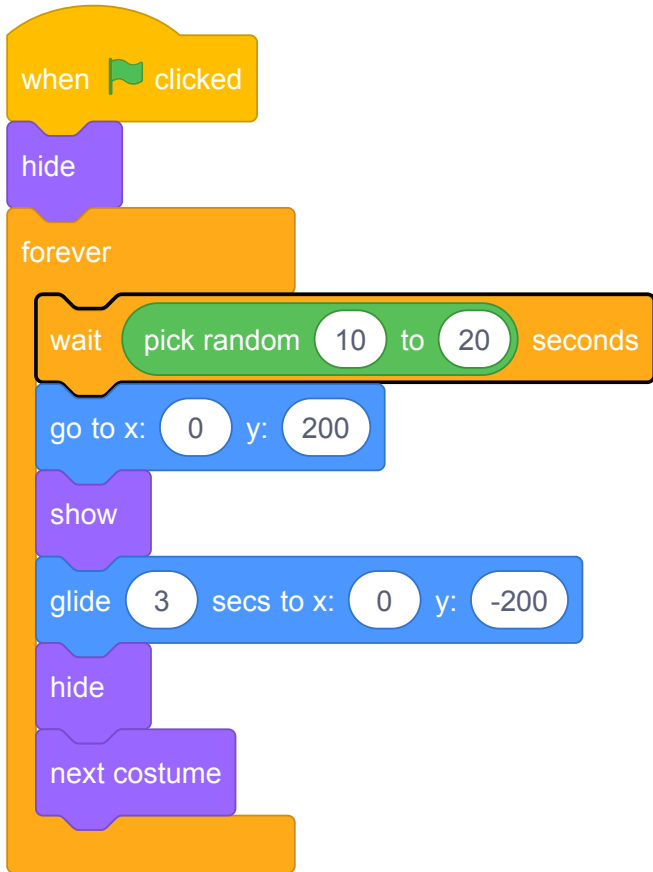
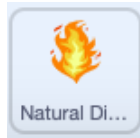
when green flag clicked
  hide
  forever loop
    go to x: 0 y: 200
    show
    glide 3 secs to x: 0 y: -200
    hide
    next costume
  
```

Test your project. The Natural Disaster sprite now moves from top to bottom, and changes costume each time.



Scientists can sometimes predict when natural disasters will occur, though this can be difficult because their timing isn't always clear, unlike in the simulation. Now, add random timing to the Natural Disaster sprite to make the sprite's appearance unpredictable like in real life.

To create random timing add a **wait 1 seconds** block to the top of your **forever** block. Drag a **pick random** operator into the **wait** block, and change the values to **10** and **20**:



Add randomness to the movement of the Natural Disaster sprite.

Drag a **pick random** operator into the **x:** coordinates of both the **go to** block and the **glide** block. Change the values in each to **-200** and **200**:



```
when green flag clicked
  hide
  forever loop
    wait 10 to 20 seconds
    go to x: pick random -200 to 200 y: 200
    show
    glide 3 secs to x: pick random -200 to 200 y: -200
    hide
    next costume
```

Test your project again. Your Natural Disaster sprite should now wait between 10 and 20 seconds before it falls from a random position to another random position. It also changes costume each time and removes any trees it touches.



Save your project

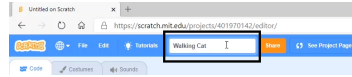
## Step 8 Share your project

You can share your completed Scratch project with your friends and family. To do this, follow the instructions below:

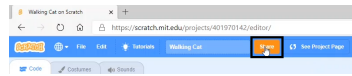
### Share your Scratch project

Please make sure that you do not share any personal information about yourself when sharing your Scratch projects.

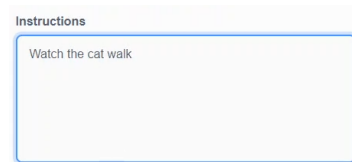
- Give your Scratch project a name.



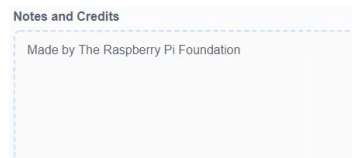
- Click the Share button to make the project public.



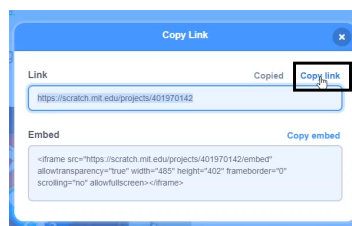
- If you like, you can add instructions in the Instructions box, to tell other people how to use your project.



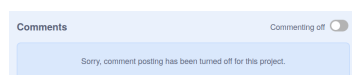
- You can also fill in the Notes and Credits box: if you have made an original project, you can write some short comments, or if you have remixed a project, you can credit the original creator.



- Click the Copy Link button to get the link to your project. You can send this link to other people by email or text, or on social media.



Scratch provides the ability to comment on your own and other people's projects. If you do not want to allow people to comment on your project, you should turn off commenting. To turn off commenting, set the slider above the Comments box to Commenting off.



If you are in a club, why not demonstrate your project to friends and peers?

## Step 9 What next?

---

In this project, you have:

- Used `clone` blocks to generate trees in `random` locations
- Used `say` operators to control the human character to alert the user about the state of the environment
- Programmed a simulation to explore the impact of humans on trees and the surrounding wildlife

If you are following the Protect our planet (<https://projects.raspberrypi.org/en/raspberrypi/protect-our-planet>) projects, then try out the next project, Count the creatures (<https://projects.raspberrypi.org/en/projects/count-the-creatures>). In this project, you will create a program to input real data on the range of natural resources used by countries to generate electricity. Then, you will compare the results using animated and interactive data visualisation.

Did you enjoy the project? Have you spotted a mistake? Please click the Send feedback button below and let us know!

---

Published by Raspberry Pi Foundation (<https://www.raspberrypi.org>) under a Creative Commons license (<https://creativecommons.org/licenses/by-sa/4.0/>).

View project & license on GitHub (<https://github.com/RaspberryPiLearning/tree-life-simulator>).