Notes

Sierpinski Triangle Simulation

What is it?
Sierpinski Triangle is a simulation that shows an initial green cell with a nucleus (a small white circle) that in the next generation produces two similar daughter cells, one at its left, the other at its right. The daughter cells repeat this for another 20 or so generations, leading in the end to the structure of a Sierpinski triangle.

How it works
The simulation starts with an initial patch in the middle of the top-row of the canvas, which has been colored green by a single turtle, who itself was created in a "tosetup" procedure. Besides marking the initial patch, the parent turtle also replicates by hatching a "child" on that patch. The hatched child itself also leaves a "copy" of itself behind (a grand-child, so to speak).

The "to go" procedure asks the "child" to run a "to move" procedure. The "to move" instructs the child to go to a patch below and at the left of the original cell, followed by going to a patch below and at the right of the original cell. Before it makes that second move, it calls the "tomark-and-breed" procedure. It does so again after completing the second move.

The "to go" runs for a defined number of generations. To do this, a counter called "generationstep" has been set at 1 in the setup and is increased by 1 every time the simulation goes through a next round of "to go". The "to go" stops after it has been checked if generationstep = #generations. Note: that each cell contains just one "copy". This means that the parent turtle and the "children", but not the "copies", must have "died".

How to use it
- First, click the "setup" button to set up the initial agent.
- Then, click the "go" button to run the program.
- Move the #generations slider to specify the number of generations outputted.
- Note: The "to go" stops after it has been checked if generationstep = #generations.

Credits
This script was created by Anthony Constant (AC). If you have any questions or suggestions, you can contact him at anthonyconstant.co.uk/

License

GitHub

Share Link: https://github.com/Anthony-Constant/Sierpinski-Triangle-Simulation

NETLOGO COPY & PASTED LOCAL SOURCE CODE

```
# Sierpinski Triangle Simulation
# Created Sierpinski Triangle Simulation in NetLogo
# Author: Anthony Constant (AC)

breed [children child]
breed [copies copy]
globals [generationstep]


to setup
  clear-all ; clear all patches
  set generationstep 0 ; sets generationstep to 0
    create-turtles 1 ; create 1 new turtle
  [

    setxy (max-pxcor / 2) max-pycor ; set the coords for intial setup
    set size 0.75 ; set size 0.75 for turtle
    set color white ; set color white for turtle
    set shape "circle" ; set circle shape for turtle

  ]
end


to move
  ask turtles with [color = white] [ ; if turtle = white
    fd sqrt 2 ; move forward square 2 and iterrate
    ifelse count turtles-here with [color = brown] > 0 [die][ ; if turtles = brown use die command
    ask patch-here [
      ifelse pcolor = black [set pcolor green][set pcolor red] ; if patch color is equal to black set patch color to green/red
    ]
  ]
  ]
```

```
end


to make-copy
  ask patches [
    if count turtles-here with [color = brown] > 1 [ ; if the turtle color equal to brown
         set pcolor red ; and set color to red to represent non division
         ask one-of turtles-here [die] ; use die command
    ]
  ]
  ask turtles [set color brown] ; set turtle color to brown
end

to mark-and-breed
  ifelse count turtles = 0 [ ; if there are no turtles on the patch
    ask patch 0 0 [
      set pcolor green ;sets patch color to green
      hatch 1 [ ; creates a turtle
    set shape "circle" ; sets the shape to circle
    set size 0.6 ; sets the size to 0.5
    set color white ; sets color to white
      ]
    ]
  ][
    ask turtles with [color = brown][ ; if turtle color is brown
      if [pcolor] of patch-here = green [ ; if patch color is green
      hatch 1 [ ; create turtle
    set shape "circle" ; set shape to circle
    set size 0.6 ; set size to 0.5
    set color white ; set color to white
    set heading 135 ; set heading (angle) to 135
      ]

      hatch 1 [ ; create turtle 1
    set shape "circle" ; set shape to circle
    set size 0.6 ; set size to 0.5
    set color white ; set color to white
    set heading -135 ; set heading (angle) to -135
```

```
      ]

    ]
    ]
  ]
end

to go
  if generationstep < #generations [ ; if #generations slider is less than generation step then...Run
    set generationstep generationstep + 1 ; add 1 to the generation step each time you run go function
    mark-and-breed
    move
    make-copy;
  ]

end
```