LESSON 8: November 11 - Nov 18 * The Penultimate PDF *
Vocabulary added to our WIki: fractal, recursion

Hello everyone,
As we head to the last two weeks of our course, I am going to offer you the last big topic of this course -- fractals -- and will keep all of the HW problems optional so that there is no guilt, yet for those of you who have "caught up" and are "on top of everything" you still have new tasks and challenges to do.

I am truly hoping that you will by next Monday 11/18 one week from today, you have completed the writing regarding
-- the partner activity and the writing on the WIki
-- the writing on the WIki about your Columbus New World project (even if the project is in the future)
Please, please make it a priority to get these two things done so that I can wake up the morning of Monday 11/18 and see all the writing there.
Yes, **33** "pleases" above -- sbergen**33**@gmail.com

I am also offering a few Hopscotch problems on variables since Jessica and Bridgid are now certified Hopscotch-ers and this will be advantageous to you two! Steve

Bridgid
AHNJaguars


Bryan
MrSark


Jessica
JLanCas2


Matt
MrMel


Maureen
MacTechWitch


Ruth
TechnoTeachRuth


Zhanna
ZhannaC


Team 1 - Bridgid and Zhanna (FL and NY)
Team 2 - Bryan and Ruth (NY and TX)
Team 3 - Jessica, Maureen and Matt (TN, NY and FL)

HW 8.1 due Nov 11th The assignment is to work with your partner(s) and jointly write a paragraph or two on our WIKI about one of the featured projects that you found intriguing. **If you and your partner need an extension, please ask!**
HW 9.1 due Nov 18th The Columbus Day New World project is done and you have shared in a paragraph or two on our WIKI how it went.

Lesson 8: Fill yourself up at the Smorgasbord -- consume as much or as little as you want!

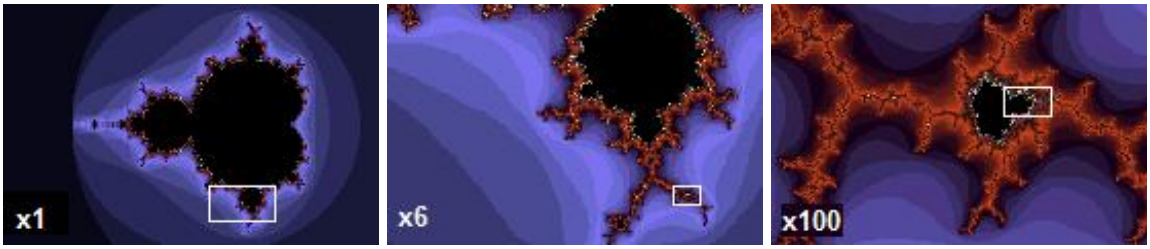
The **first dish** on the smorgasbord is a treatment of recursion -- an important part of computer science and computer programming theory.

Here is the easiest definition I could find from the website of
http://www.techterms.com/definition/recursivefunction
If you look up *Recursion Computer Programming* on Google, you get so much amazing complexity.

"A recursive function is a function that calls itself during its execution. This enables the function to repeat itself several times, outputting the result and the end of each iteration. Recursive functions are common in computer science because they allow programmers to write efficient programs using a minimal amount of code. The downside is that they can cause infinite loops and other unexpected results if not written properly. For example, the function may be terminated if the number is 0 or less or greater than 9. If proper cases are not included in the function to stop the execution, the recursion will repeat forever, causing the program to crash, or worse yet, hang the entire computer system."

Here is my 3 minute video -- http://youtu.be/r40MHcsWxql -- hopefully humorous but informative attempt to explain with my Larry Bird doll what recursion is and why it is different from repetition using the REPEAT command.

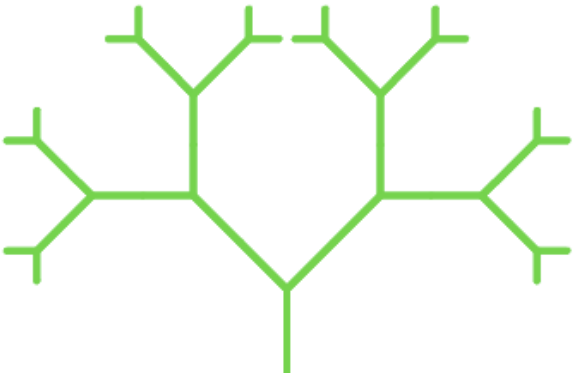
Starting in the 1970s, computer programmers were able to use the techniques of recursion to create fractals. You may have heard of the Mandelbrot fractal on the left. When you zoom in to this fractal on the left you get the graphic in the middle and then when you zoom in again, you get the graphic on the right. Each "zoom" yields a shape that is proportional (similar) to the original one.



This paragraph from http://en.wikipedia.org/wiki/Fractal may be of help:

"The mathematical roots of the idea of fractals have been traced through a formal path of published works, starting in the 17th century with notions of recursion, then moving through increasingly rigorous mathematical treatment of the concept to the study of continuous but not differentiable functions in the 19th century, and on to the coining of the word fractal in the 20th century with a subsequent burgeoning of interest in fractals and computer-based modelling in the 21st century. The term "fractal" was first used by mathematician Benoît Mandelbrot in 1975. Mandelbrot based it on the Latin frāctus meaning "broken" or "fractured", and used it to extend the concept of theoretical fractional dimensions to geometric patterns in nature."

So our 8.2 journey today is make our own fractal.

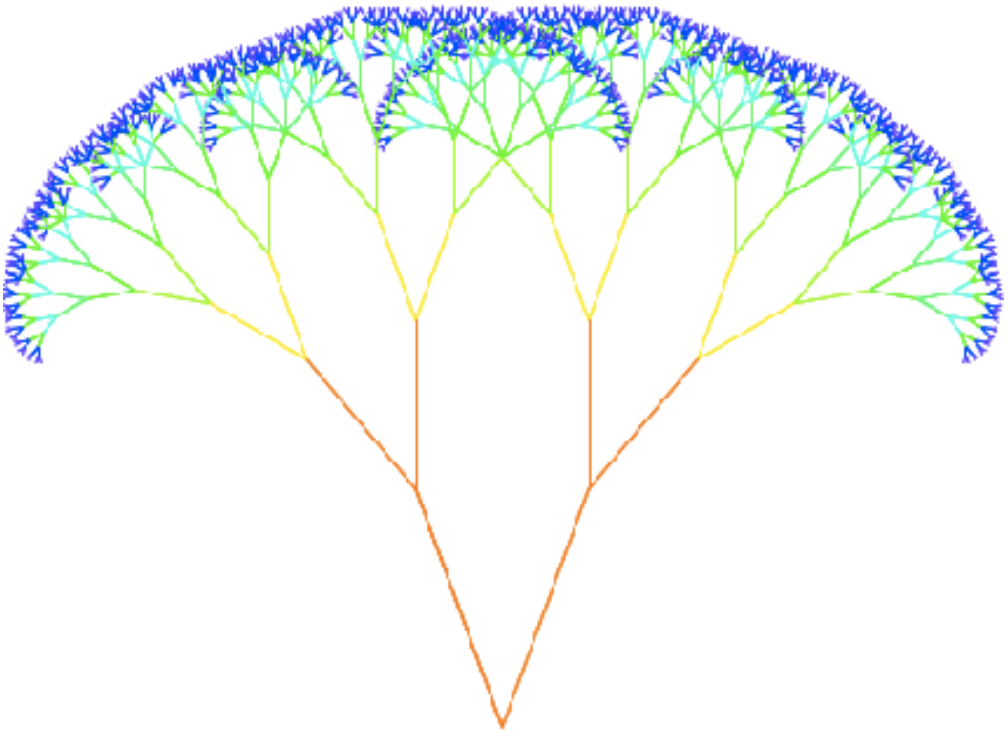


Optional Video Lesson 8.2 Fractals Part 1 (5 min) http://youtu.be/WQfEd5AcnVY
This gives you the orientation to what a fractal is and an overview of the one we are going to code from scratch using Scratch.

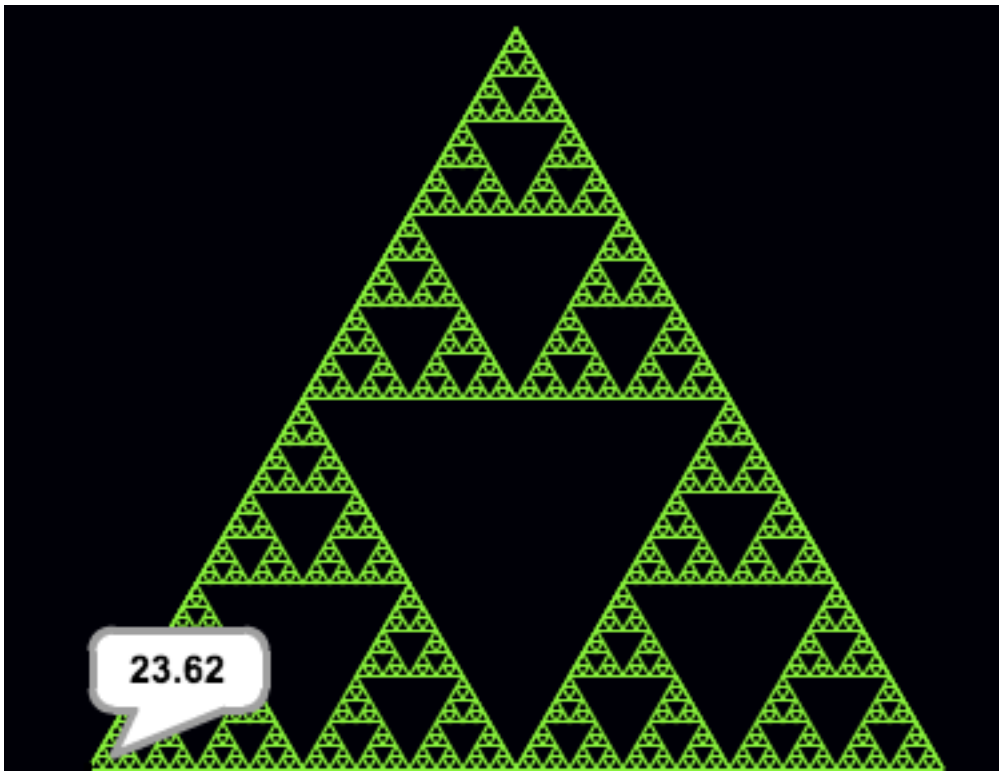
Optional Video Lesson 8.2 Fractals Part 2 (18 min) youtu.be/LKkVzmBB5Zs
This takes you through the nitty gritty programming for creating a tree fractal. This is a tough challenging topic yet needs to be an important part of any coding course. I wish I could teach this to someone in 33 seconds but I can't!

The above two videos will show you how to make this primitive "tree fractal" which of course I have shared with you in my Scratch library as 8.4.

A Better Tree Fractal by someone else on Scratch
This fractal tree from the library of NGMR is much nicer. I have remixed it for you and it is called Remix 8.4 Fractal Tree by NGMR
http://scratch.mit.edu/projects/13941742/



A Different Type of Fractal called The Sierpinksi Triangle
I have also remixed for you a famous fractal called the Sierpinksi Triangle and it is called Remix 8.3 Sierpinski by S65 and Cyclone103



I hope you see the way that each of these graphics is a fractal in that the big picture is proportional and similar to any magnified portion.

Connections to Education and Our Kids
How does fit into education, particularly lower school? Kids can and should learn about these graphics and they can identify and see real world examples such as rivers, trees, leaves and snowflakes. Creating connections between computers, mathematics and nature is part of developing in children an enthusiasm for what is now being packaged as STEM in our schools -- Science, Technology, Engineering and Mathematics.

See http://fractalfoundation.org/2009/02/fractals-on-the-earth/
Fractals on the Earth

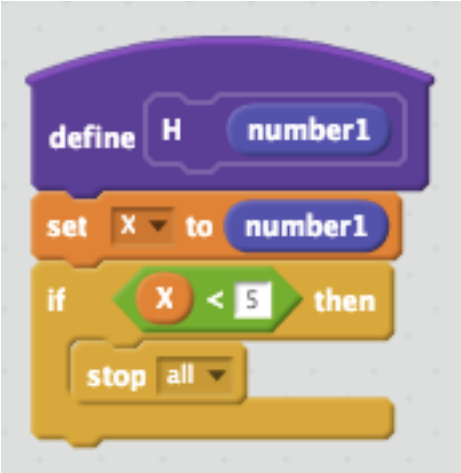
or see http://math.rice.edu/~lanius/frac/
A Fractals Unit for Elementary and Middle School Students

Optional HW 8.2a
Watch my video on how to create the tree fractal, study it, remix it and make it your own. Then 24 hours later, see if you can do it yourself from scratch without looking at notes or code.

Optional HW 8.2b
Modify my program to have colors. Modify my program so that the left branches of the tree are not symmetric with the right branches.

Optional HW 8.2c
Make your own fractal using one of the block letters you created of variable size in a previous lesson.

Let's say you made a H of variable size.
Then you can make a fractal by adding the H command at each corner of the H.
Here is the big picture:



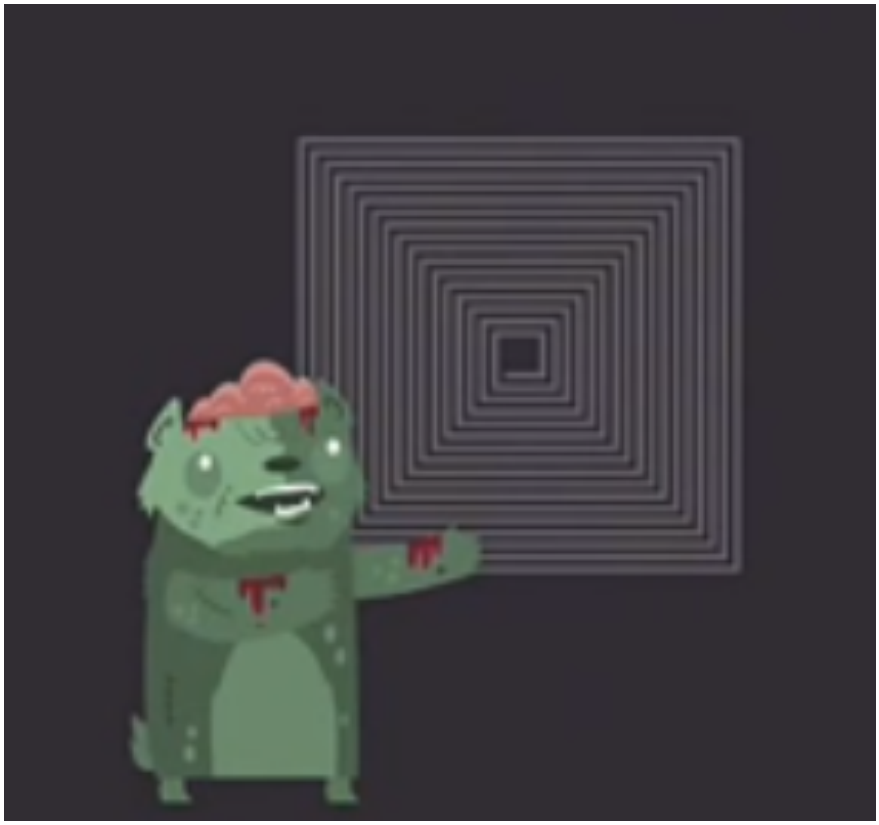
then do the steps for an H of variable size
and at each corner of the H you include

Get the logic? It will draw an H of specified size NUMBER1 which then goes into X. As long as X stays above 5 then all is good. At each "corner" of the H, it will do a smaller H of one third size. If you succeed with your H fractal, you can then sing the song from West Side Story "Recursion, recursion, I just wrote a program with recursion."

Here is another dish from the Smorgasbord for the two Hopscotchers (or anyone else since both problems could be done in Scratch if they catch your fancy)

Hopscotch Challenge One of Lesson 8 -- let us call it 8H1
Write a program to draw a graphic that is "square like" with each side being somewhat bigger than the previous one so that it looks like a maze.

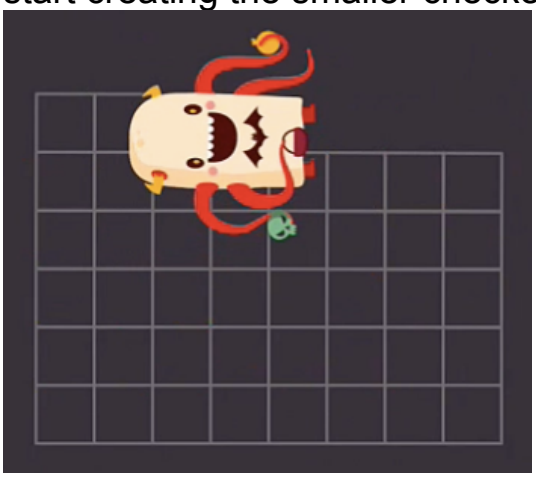
This 3 minute video at http://youtu.be/6UzwhsBQrtc provides you with one solution. Do not watch it unless you need hints or a solution which is also on the next page sideways. Yes, this challnege needs a variable!

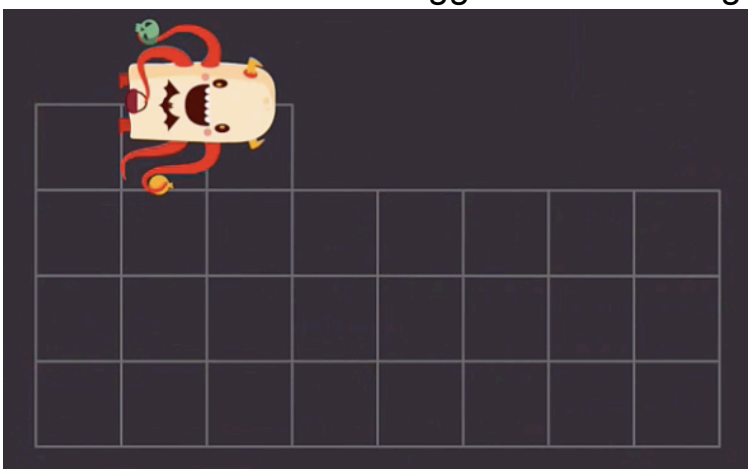


Notice if you squint that this is not "nested squared" but rather line segments that increase a little bit at each corner so in effect this is a maze that a very thing ant could walk thorugh.

Hopscotch Challenge Two of Lesson 8 -- let us call it 8H2

Write a program that builds a VARIABLE SIZED CHECKERBOARD of size 8x8
In other words, it is built with a variable X so depending on the value of X, it might start creating the smaller checkerboard on the left or the bigger one on the right.





The 10 minute video at http://youtu.be/SHtRDZ4Kn7s describes the problem and shows some samples. In the last few minutes, there is a clearly noted SPOILER ALERT with the solution but you are given plenty of notice in the video to stop.

One possible solution to 8H2 is on the next page sideways.

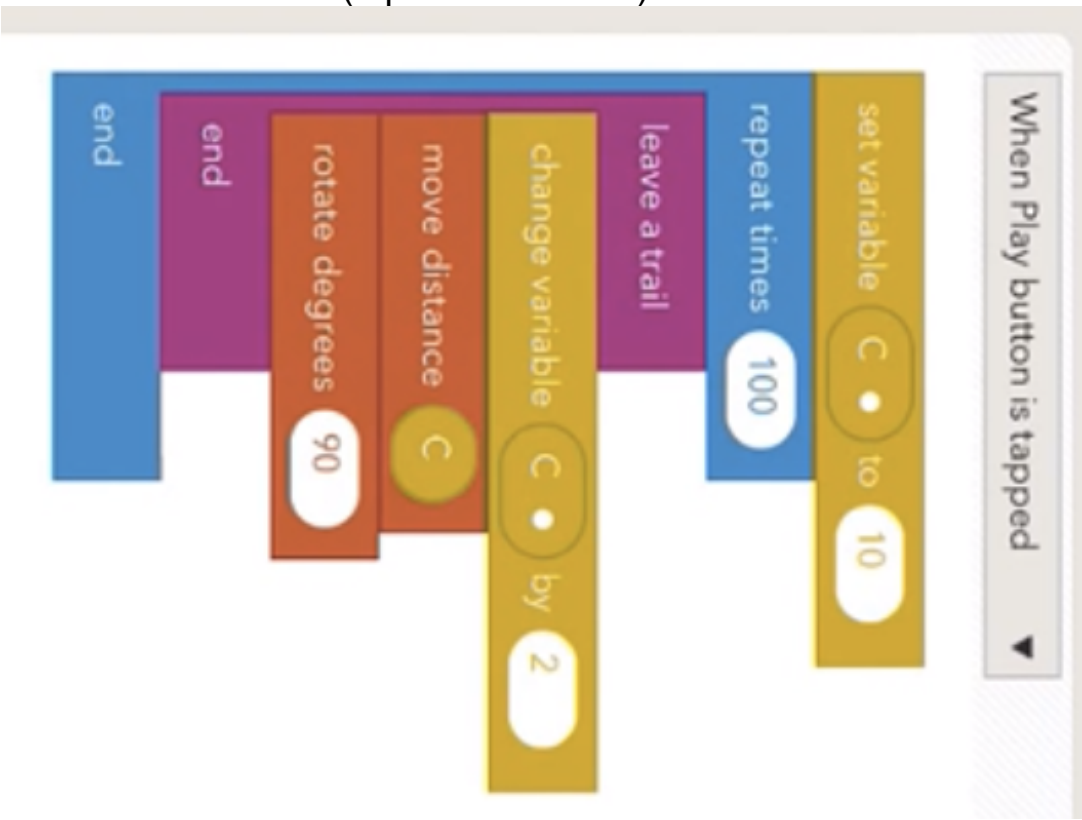
What you have to think about is drawing one square and then positioning yourself so that you are ready to draw the next square. What are the steps after square one is drawn so that you are located at the spot and with the orientation ready to draw the next square. This program does not involve recursion but it absolutely involves repetition with the REPEAT command.

REPETITION
using the REPEAT command
the same shape is drawn repeatedly
classic example: a checkerboard
just imagine: a checkerboard where each square contained a smaller checkerboard

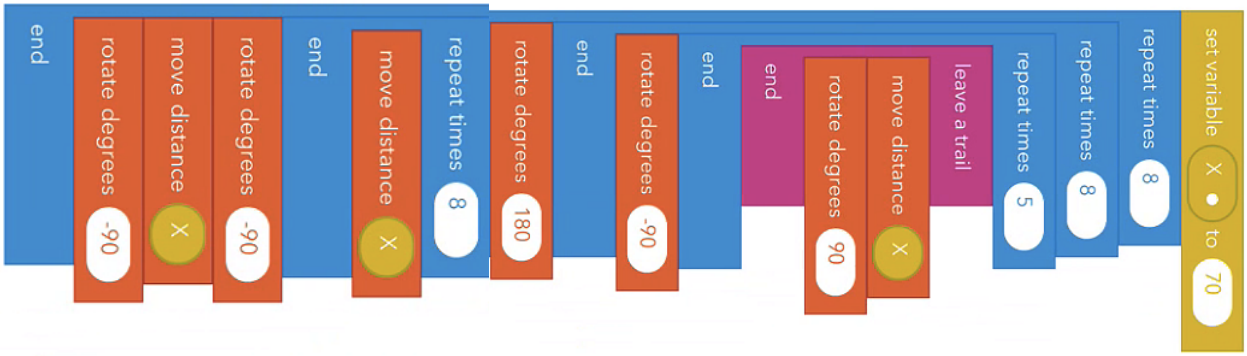
RECURSION
using the REPEAT command
each subsequent shape is similar or proportional to the original one by some fraction, e.g. 1/3 or 1/4
that is why we get fractals
classic example: a tree
just imagine: a tree where each branch was the same size and thickness of the original one. It would topple!

Hopscotch Spoiler Alerts

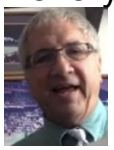
One solution to 8H1 (square like maze)



One solution to 8H2 (variable sized checkerboard)



Have a good week everyone. Bon Appétit with the Smorgasbord!

Steve 

781-953-9699 • skype name = stevebergen (no spaces) • sbergen33@gmail.com