LESSON 7: February 24 (the penultimate PDF)

First Required Assignment HW 7.1

First Assignment is go to our WIki and share some reflections in the fourth column of our Who's Who's Table so that others can read about what plans you might have for the future regarding coding at your school moving forward from April 2014 onward.

Second Required Assignment

As we head to our last lesson next week on March 3, I would like to make sure you prioritize HW 6.2 / HW 8.1 in terms of your finite time. I have posted a section on HW 6.2 on our wiki.

HW 6.2 / HW 8.1 is to communicate with your partner by phone, email, skype or google hangout and to jointly pick one of the sixth grade projects by Jeff's students in Houston TX. The two of you will end up teaching the rest of the class about this project the week of March 3-7 when we hit our last week of our course.

The projects are at http://scratch.mit.edu/studios/303333/ but also on our January webpage direct under the list of names and partners.
You and your partner should pick one that
a) you like (I think this will be easy)
b) can decipher the code and make sense of how the program works (not so easy)
You and your partner will be able to choose amongst various options for turning in your work on March 3rd. This assignment is called 8.1 and is due on 3/3.

What are some of these options?
Plan A -- you can just write about it via text on our wiki.
Plan B -- one of you can REMIX it then add some yellow comments and share it. Then the second person can access and REMIX that one and add yellow comments. So the final version has been marked up by both partners and shared so I can add to my studio.
Plan C -- You can create a new Scratch account with a login and password that both partners use. As long as you don't access it at the same time, you can each mark it up with comments. The final version is then shared so I can add to my studio.
Plan D -- You can take screen snapshots and put into a Google Document that is shared between the two of you. The final version is then a Google Document that you mark as ANYONE CAN VIEW.

Plan E -- You can post a private YouTube video of the two of you talking about the app (with one of you on speaker phone or skype) while you record the screen (the way I did with Jeff from Houston). Glad to help you with this option if interested.

So the main assignment is either get moving on this and to put some notes in the 6.2 section of the Wiki where you will see your names with your partners.

That's it folks! The content of the lesson below -- on recursion and fractals -- is one of my favorite topics in computer programming and I believe fascinating. However, the partner activity above is the priority and so I am declaring everything below as OPTIONAL.

Happy end of February and start of March. Spring is here (favorite Tom Lehrer song below)

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

Spring is here, a-suh-puh-ring is here.
Life is skittles and life is beer.
I think the loveliest time of the year is the spring.
I do, don't you? 'course you do.
But there's one thing that makes spring complete for me,
And makes ev'ry sunday a treat for me.

No, this picture below is not a joke but it is part of the "F" for flexibility theme for the final two weeks in our online course. Fractals in Scratch are wonderful but some of you are playing catchup and this way you can partake of whatever you wish from the no guilt Smörgåsbord below.

Recursion and Fractals -- Exciting & Empowering

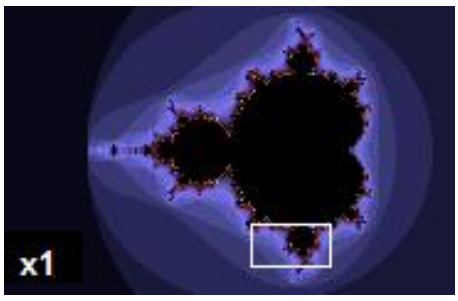
Let us begin with an explanation of recursion -- an important part of computer science and computer programming theory.

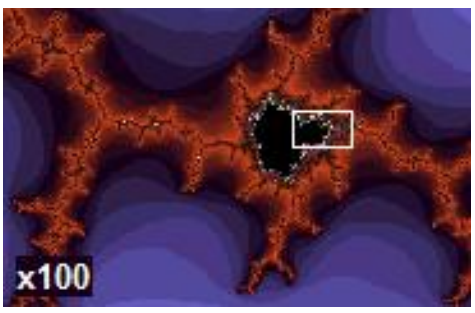
Here is the easiest definition I could find from the website of http://www.techterms.com/definition/recursivefunction
If you look up *Recursion Computer Programming* on Google, you get so much amazing complexity.

"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/r40MHcsWxqI -- 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.

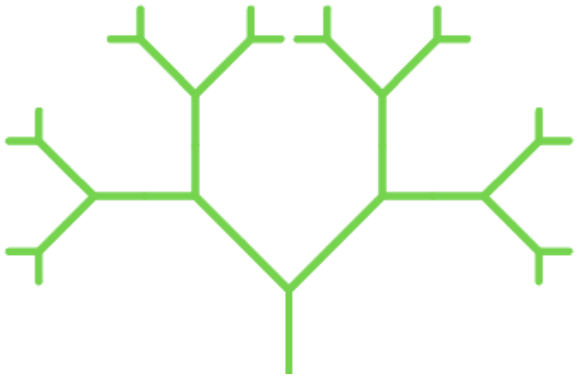




This paragraph from http://en.wikipedia.org/wiki/Fractal may be of help:

"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 journey today is make our own fractal.



Video 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.

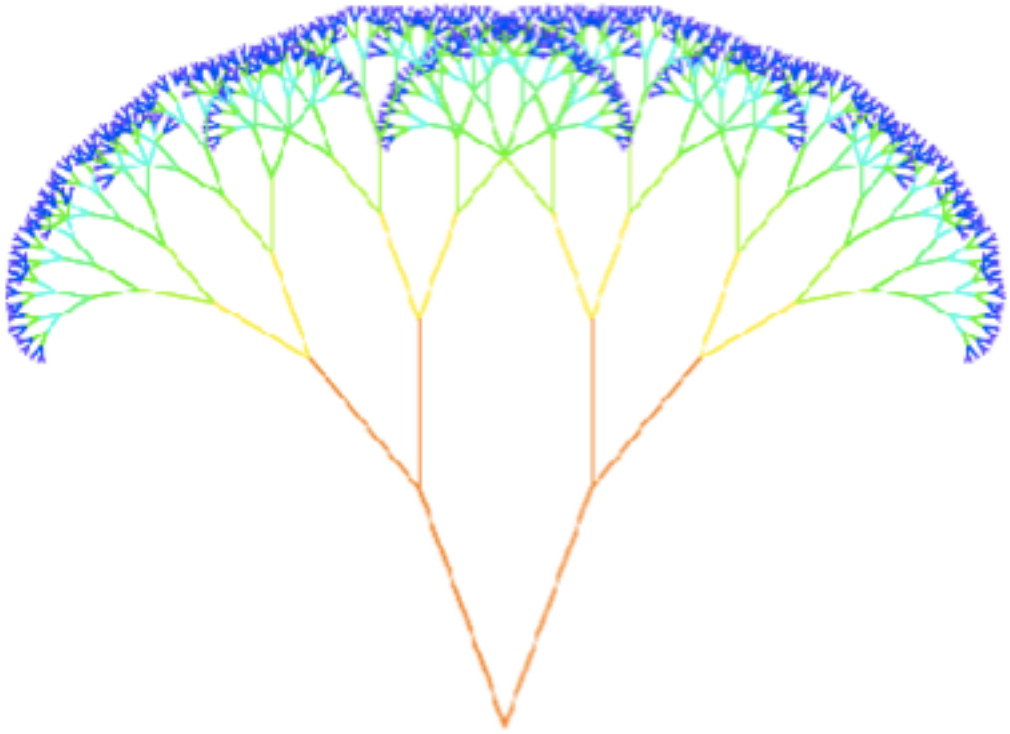
VideoLesson Fractals Part 2 (18 min) http://youtu.be/LKkVzmBB5Zs
This takes you through the nitty gritty programming for creating a tree fractal. This is a tough challenging topic yet needs to be an important part of any coding course. I wish I could teach this to someone in 33 seconds but I can't!

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/studio
http://scratch.mit.edu/projects/13740603/

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 Fractal Tree by NGMR

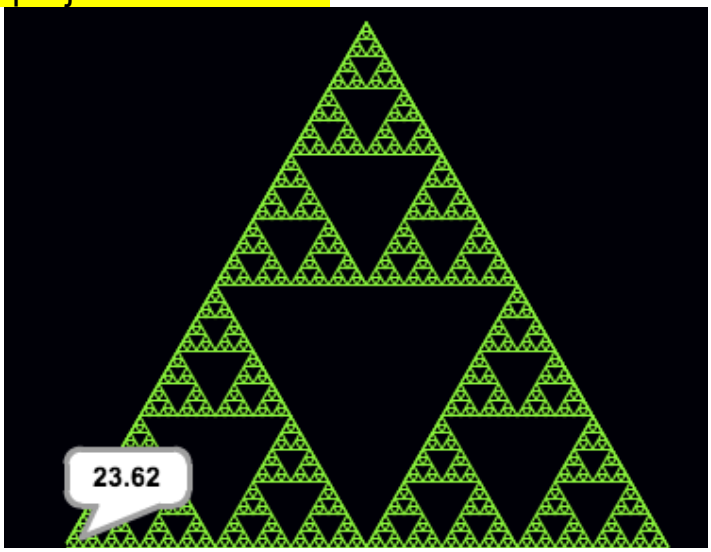
http://scratch.mit.edu/projects/13941742/



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 Sierpinski by S65 and Cyclone103

http://scratch.mit.edu/projects/13941868/



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

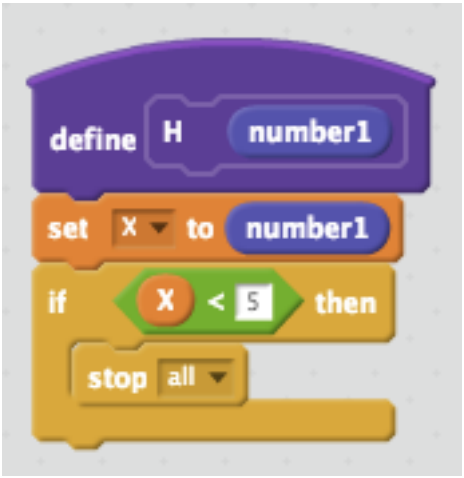
Optional HW 7.2a
Watch my video on how to create the tree fractal, study it, remix it and make it your own. Then 24 hours later, see if you can do it yourself from scratch without looking at notes or code.

Optional HW 7.2b
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 7.2c

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." Let me know and I will add it to my new fractal studio.