Asian MathSci League, Inc (AMSLI)
Partner: National Olympiad in Informatics Philippines (NOI.PH)
Email address: amsliphil@yahoo.com / ask@noi.ph
Contact Nos: +632-9254526 +63906-1186067

THAT'S WHAT FERNS ARE FOR

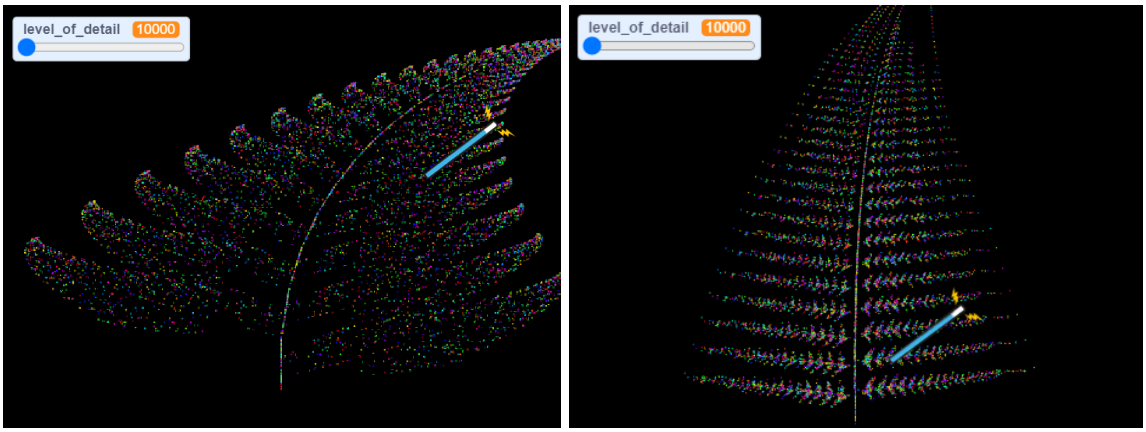
Mark Edward M. Gonzales

In his 1988 book *Fractals Everywhere,* Michael Barnsley explored how fractals can be used to model various things in nature, such as the structures of plants. His investigations led to the discovery of a set of equations that can be used to draw a fern that resembles the black spleenwort.

In this problem, we will be exploring how to draw this fern — now referred to as the Barnsley fern — and a mutant variety that resembles the *Thelypteridaceae* fern using Scratch!

A. VIEW THE FINAL OUTPUT

To hopefully excite you and tease your curiosity, you can view the video of the final output via this link: https://bit.ly/AIEP_Fern_Video.



B. DOWNLOAD THE SKELETON FILE

Download the skeleton Scratch file from this link: https://bit.ly/AIEP_Fern. This file contains the initial code to help you get started with the problem.

> IMPORTANT! You are not allowed to remove any existing block in the given skeleton file. The changes that you can make are limited to adding blocks and supplying missing values.

C. INITIALIZE THE PROJECT COMPONENTS

The variables `x_now` and `y_now` pertain to the current position of the sprite (a wand). When the flag is clicked, the wand should move to the center of the screen and should point in the direction 53 degrees. Make sure that all the drawings on the screen are erased and that all the lists are emptied before proceeding.

> The broadcast messages `initialize` and `clear_all_lists` have already been created for you. Your task is to supply the missing values and add the needed blocks when these broadcast messages are received.

D. ASK FOR USER INPUT

In this problem, we will be generating two types of ferns: Barnsley's original black spleenwort and the *Thelypteridaceae* fern. The user can choose between the two by entering either `1` or `2`, respectively. If the user inputs anything invalid, such as `-1` or `1234`, the program should keep asking until a valid input is received.

> The `repeat until` block for this task has already been provided for you. Your task is to add the needed blocks to accomplish this subtask.

E. BUILD THE LISTS

The equations that we will be using to draw the fern rely on certain constant values called "probability factors" and "coefficients." Following good coding practices, we store these constants inside seven separate lists. The probability factors are stored inside the `probability_factors` list while the coefficients are stored inside `a`, `b`, `c`, `d`, `e`, and `f`.

The tables on the next page show the values that have to be stored inside the lists. Note that the values are different based on the type of fern to be drawn.

> The broadcast messages `build_probability_list`, `build_a_list`, `build_b_list`, all the way up to `build_f_list` have already been created for you. Your task is to supply the missing values and add the needed blocks when these broadcast messages are received.

Constant Values for Black Spleenwort

	probability_factors	a	b	c	d	e	f
Item 1	0.01	0	0	0	0.16	0	0
Item 2	0.85	0.85	0.04	-0.04	0.85	0	1.60
Item 3	0.07	0.20	-0.26	0.23	0.22	0	1.60
Item 4	0.07	-0.15	0.28	0.26	0.24	0	0.44

Constant Values for *Thelypteridaceae* Fern

	probability_factors	a	b	c	d	e	f
Item 1	0.02	0	0	0	0.25	0	-0.4
Item 2	0.84	0.95	0.005	-0.005	0.93	-0.002	0.5
Item 3	0.07	0.035	-0.2	0.16	0.04	-0.09	0.02
Item 4	0.07	-0.04	0.2	0.16	0.04	0.083	0.12

F. SET THE LEVEL OF DETAIL

The number of points to be used in drawing the fern is controlled by a variable named `level_of_detail`, which is also used as the counter for the `repeat` block. To allow the user to control its value, display it on the screen as a slider, and set its minimum and maximum values to 10 000 and 100 000 000, respectively.

> Once the level of detail has been set, steps G to L will have to be repeated for every iteration of the `repeat` block.

G. GENERATE RANDOM PROBABILITIES

Now we have come to the interesting part! Using a set of equations, we will be generating the ferns. These equations are quite special, as they describe a mathematical concept known as "affine transformation."

How do we know which equation to use? It depends on the probability factor. Your task is to set the variable `probability_factor` to a random (decimal) number between 0 and 1, inclusive.

In generating the random number, be mindful of the distinction between integers and floating-point numbers. You can read more about it here.

Follow these instructions to determine which equation to use:

- If the random probability factor is less than the first item in the list `probability_factors`, use the equation for the stem, so go to Step H.
- If the random probability factor is less than the sum of the first two items in `probability_factors`, use the equation for the small leaflets, so go to Step I.
- If the random probability factor is less than the sum of the first three items in `probability_factors`, use the equation for the large left leaflet, so go to Step J.
- Otherwise, use the equation for the large right leaflet, so go to Step K.

The `if-else` structure has already been created for you. Your task is to supply the missing conditions.

IMPORTANT! You should get the values from the list `probability_factors` and not manually type them in your conditions.

H. COMPUTE THE EQUATION FOR THE STEM

When the `draw_stem` message is received, the My Block `affine_transform` is called; this My Block accepts one numerical value named `item number`. In drawing the stem, the first item is considered.

Your tasks are to supply this missing value and complete the `affine_transform` My Block. Inside this My Block, perform the following operations:

- Get the product of the item `<item number>` in `a` and `x_now`.
- Get the product of the item `<item number>` in `b` and `y_now`.
- Add the sum of these two products to item `<item number>` in `e`.
- Assign the resulting value to the variable `x_next`.

- Get the product of the item `<item number>` in `c` and `x_now`.
- Get the product of the item `<item number>` in `d` and `y_now`.
- Add the sum of these two products to item `<item number>` in `f`.
- Assign the resulting value to the variable `y_next`.

I. COMPUTE THE EQUATION FOR THE SMALL LEAFLETS

Assuming that you have completed `affine_transform`, observe that, when the message `draw_small_leaflets` is received, this My Block is called. Your only task is to supply the missing value, which pertains to the item number. For the small leaflets, the second item is considered.

J. COMPUTE THE EQUATION FOR THE LARGE LEFT LEAFLET

Same as in Step I, your only task is to supply the missing value, which pertains to the item number. For the large left leaflet, the third item is considered.

K. COMPUTE THE EQUATION FOR THE LARGE RIGHT LEAFLET

Same as in Step J, your only task is to supply the missing value, which pertains to the item number. For the large right leaflet, the fourth item is considered.

L. DRAW THE FERN

Time to draw the fern! But before the `draw` message is broadcast, make sure to set `x_now` and `y_now` to `x_next` and `y_next`, respectively.

> The broadcast message `draw` has already been created for you. Your task is to supply the following missing values:
> - **Pen color.** Set this to a random number between 0 and 100, inclusive.
> - **Pen size.** Set this to 1.

M. TURN ON TURBO MODE

Since the animation may take some time to finish, turn on Turbo Mode when testing your project. Hope you see your fern come to life!

For the curious trainees: The Barnsley fern is an example of an **iterated function system**, a mathematical method developed in the 1980s to generate fractals. You can read more about it here.