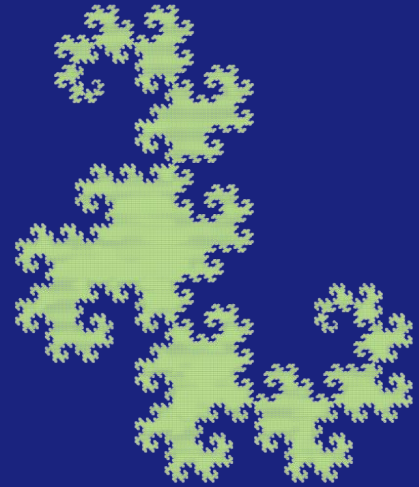
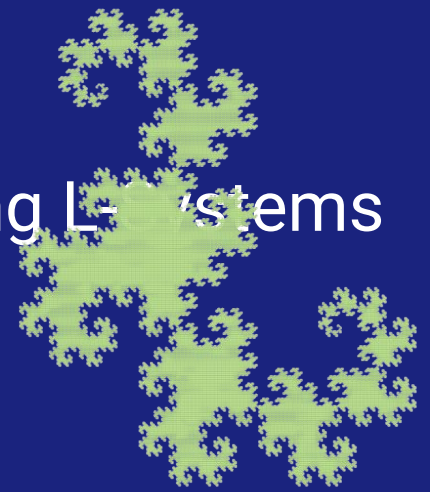


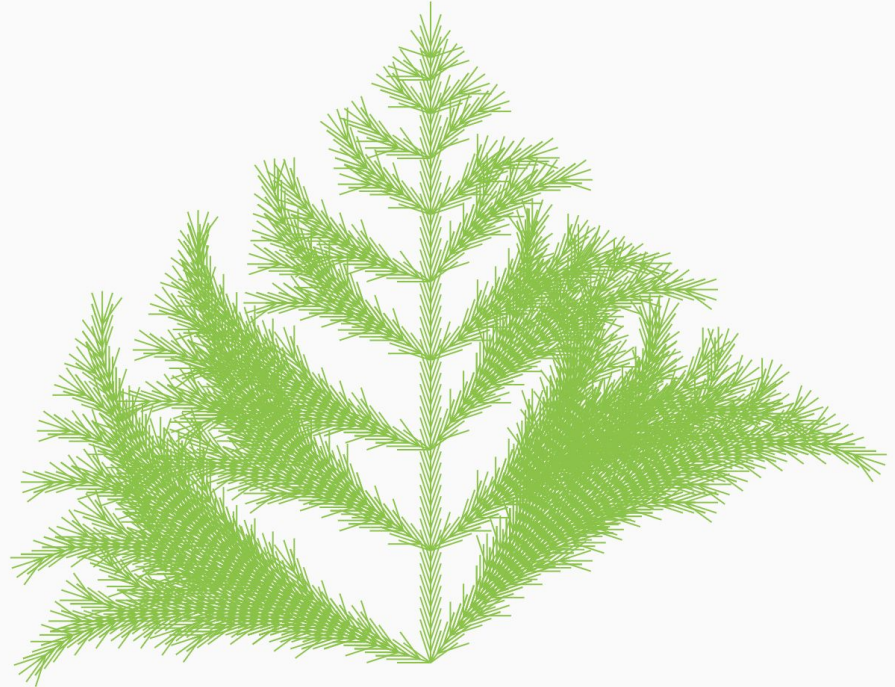
# Creating images and sound with strings using L-systems



# Contents



- Introduction
- What are Lindenmayer Systems?
- L-Systems in Flutter
- How to generate images with L-Systems
- How to generate sound with L-Systems
- Q&A





# (Ricardo) Gil



Flutter Team Lead @ Pvotal Technologies 

Kermit and Elmo Memes 

Music (play guitar and learning the violin)  



$$G = \{V, \omega, P\}$$

**Alphabet**

**Axiom**

**Production Rules**



*What does it all mean?*

$$G = \{V, \omega, P\}$$

- The alphabet is just like any grammar's alphabet, i.e., it is a list of symbols.
- The axiom is the starting point of the Lindenmayer System.
- The set of production rules dictates how we rewrite our axiom into an ever growing string by establishing a replacement relationship of the alphabet's symbols.



$$V = \{A, B\}$$

$$\omega = A$$

$$P = \{A \rightarrow AB, B \rightarrow BA\}$$



Iteration 0 -> A

(we see our string has an 'A' symbol, so we replace it with AB in the next iteration)

Iteration 1 -> AB

Iteration 2 -> ABBA

Iteration 3 -> ABBABAAB

Iteration 4 -> ABBAABBAABABBA

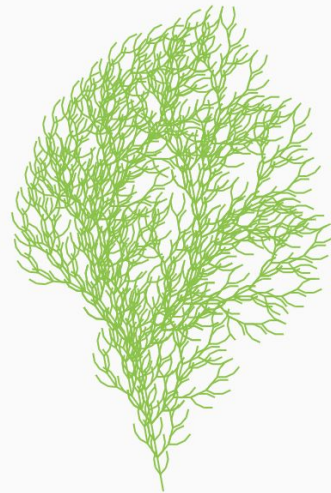
*And so on and so forth...*





Lindenmayer Systems can be used to:

- Model and draw bacteria, algae, plants and tree growth (and even entire forests! In 2D and 3D!)
- Draw fractals
- Procedurally generate maps or character models for video games
- Generate music



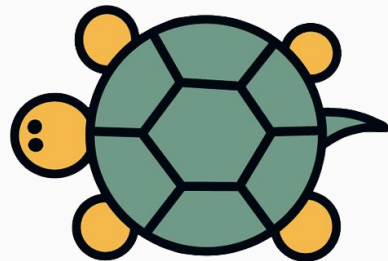
# L-Systems - Graphical Representation

When paired with turtle graphics, L-Systems can output beautiful renderings of fractals, plants and more!

A turtle has:

- An orientation (going left, right, up or down in 360 degrees)
- A location
- A “pen” to draw

*(for 3D, one has to consider an “extra orientation” the turtle has to move alongside a 3D plane)*



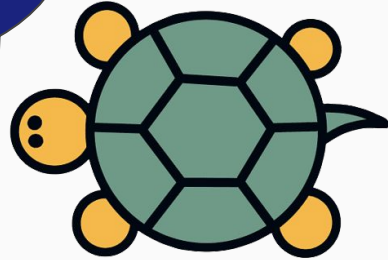
# L-Systems - Graphical Representation

We define symbols in our L-Systems that have meaning for the turtle to interpret:

- + -> Turn left by 120 degrees
- -> Turn right by 120 degrees
- F -> Draw forward by X units
- G -> Draw forward by X units

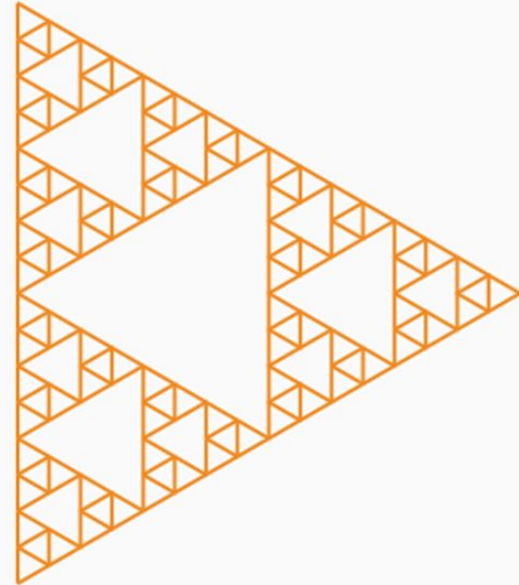


I did this!



# L-Systems - Sierpinski Triangle

- $V = \{F, G, +, -\}$
- $\omega = F-G-G$
- $F \rightarrow F-G+F+G; G \rightarrow GG$
- $n = 0 \rightarrow F-G-G$
- $n = 1 \rightarrow F-G+F+G-F-GG-GG$
- $n = 2 \dots$



## L-Systems - Sierpinski Triangle



using  
math to create  
fractals



using  
L-systems and  
a turtle to  
create fractals

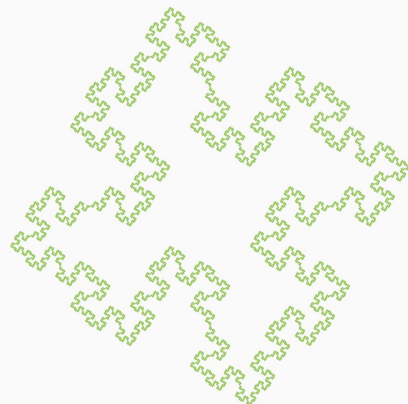
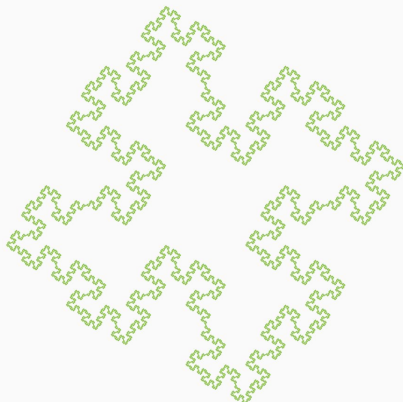
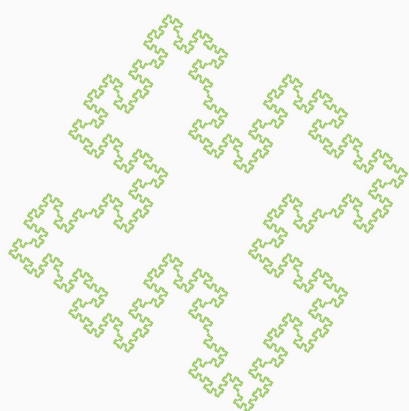
# L-Systems in Flutter

Step 1 - Add *lindenmayer-systems* to your project

Step 2 - Add *flutter\_turtle* to your project

Step 3 - Follow the example at <https://github.com/gildurao/flutter-lsystems-example>

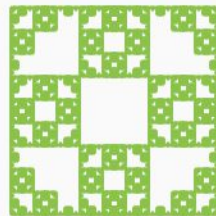
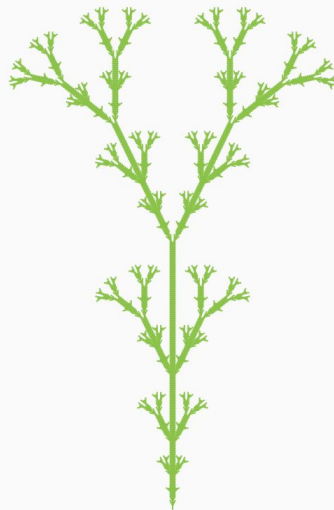
Step 4 - Create your own L-Systems and have fun!



The same way we attached a graphical meaning to an L-System's alphabet, we can attach a musical meaning:

- + Increase note pitch by X semitones
- Decrease note pitch by X semitones
- F Add note to be played to first voice
- G Add note to be played to second voice

*Note duration in this case is not handled by the L-System*



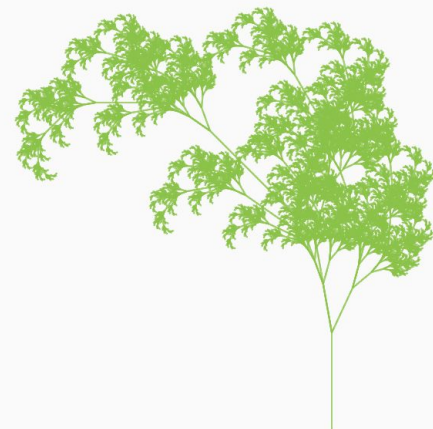
# L-Systems - Musical Representation

Step 1 - Add *lindenmayer-systems* to your project

Step 2 - Add *flutter\_sequencer* to your project

Step 3 - Follow the example at <https://github.com/gildurao/flutter-lsystems-example/tree/feature/audio>

Step 4 - Create your own L-Systems and musical parsers and have fun!





# L-Systems - What else is there?

L-Systems can be split into categories:

- Deterministic and context free
- Deterministic and context aware
- Table L-Systems
- Stochastic L-Systems
- Parametric L-Systems
- Etc..

L-Systems are a great example of how something simple can create complex results



The end



# The end

If you'd like to know more:

- The Algorithmic Beauty of Plants <http://algorithmicbotany.org/papers/#abop>
- Algorithmic Botany Papers <http://algorithmicbotany.org/papers/>
- L-Music: an approach to Assisted Music Composition using L-Systems (by me!)

<https://shorturl.at/jBNPV>

