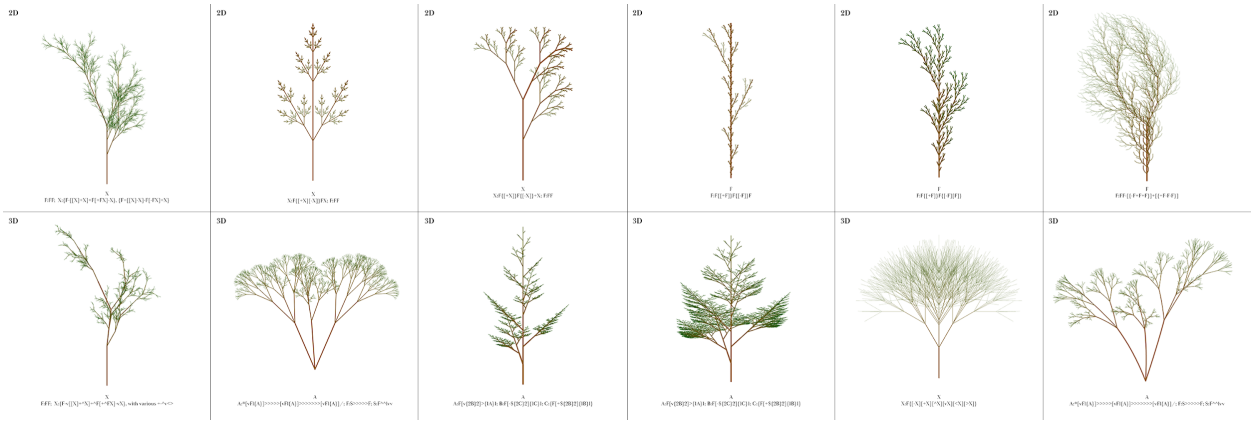
Geometree

By Cole Sohn

Man-made objects can be recreated in 3D with hard-surface modeling, but organic objects can be extremely detailed and pose a challenge for modeling by hand. Many organic objects feature structures that can be represented through multiple repeating patterns.

L-Systems is an early procedural modeling technique developed by biologist Aristid Lindenmayer in 1968 [2]. L-Systems excel at generating geometry with self-repeating patterns like trees, bushes, and other fractal-like objects.

For this project, I designed a web-based l-systems geometry generator that can generate 2D and 3D output geometry exportable as SVG and OBJ formats. My project extends the traditional L-Systems approach by giving the designer more control over additional parameters like widths, colors, and angle variations.



L-Systems:

A string of characters and a rule dictionary is passed to the program as input. A rule dictionary is a dictionary mapping strings to arrays of possible new child strings. The algorithm iteratively expands the input string by replacing substrings with randomly chosen corresponding child strings for a specified number of iterations.

Once a string has been generated representing the final geometry, it must be interpreted. This is done using a “turtle” approach. A turtle is an object that can move forward and rotate in 3D space. It can draw lines as it moves, and it can save and return to locations.

L-Systems are explained in more detail in the first chapter of Lindenmeyer’s *The Algorithmic Beauty of Plants* (TABOP) [1].

Approach:

For this project, I implemented a 2D L-Systems generator as an add-on for the 3D modeling software Blender. Then, I implemented a more robust L-Systems generator in javascript using the Three.js library for geometry processing and visualizing. I hosted this final app using Google's Firebase services.

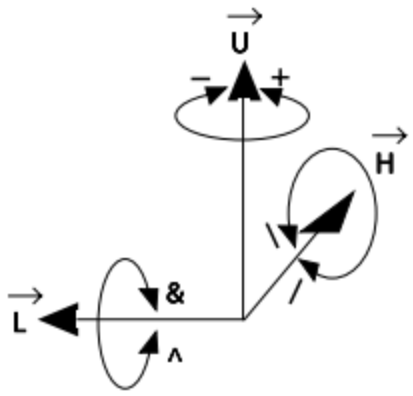


Figure 1.18: Controlling the turtle in three dimensions

The L-System starts with an Axiom (input string) and a rule dictionary of characters to strings as inputs. The strings can contain the characters seen in the table below. Each character can be interpreted as an instruction for a turtle to draw lines of the L-System.

F	Draw Forward
l	Place Leaf
+	Rotate Right
-	Rotate Left
^	Pitch Up
v	Pitch Down
>	Roll Right
<	Roll Left
[	Store Position and Orientation
]	Recover last saved Position and Orientation
{	Create a new 'level', ie. material index, line width, and line length

}	Recover and use info from old level

This implementation extends that defined in *TABOP* by allowing the designer more control over the appearance of their output. Each string has multiple "levels" differentiated by curly brackets. The designer can specify parameters that change the appearance of these levels such as variations in line lengths and widths, colors, and rotation amounts.

In the Geometree app, rule dictionaries are input as strings. Key and rule sets are separated using ';'. Different rules for one input can be separated by a ','. In this case, rules will be chosen randomly. Lindenmayer[1] calls this approach *Stochastic Similar Species*, as it can generate several unique outputs with similar unique appearances. In the Geometree web app, templates *Fuzzy 2D* and *Fuzzy* are examples of stochastically similar species. Every time they are generated they will have a slightly different appearance.

Representation:

Lines are drawn following the instructions provided by the output string. Cylinders are transformed to match each line and give the trees width. In some cases, cylinders can have different end-widths for gradual width tapering.

An 'l' in the ruleset represents the location of a leaf, which is a 2D plane that can be used for drawing a leaf texture.

Tropism:

A tropism vector can be specified to simulate forces (ie wind or gravity) that affect branch angles. The turtle's *h* value is rotated a certain amount towards the tropism vector according to the figure's equation found in *TABOP*[1]. This adds realism and complexity to the L-Systems. *e* is an intensity coefficient that the user may define.

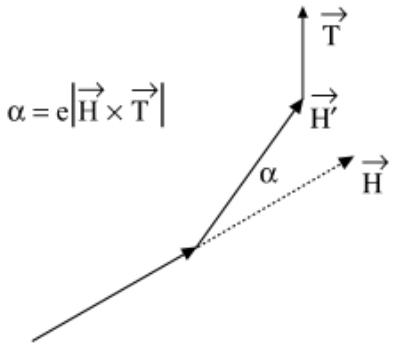


Figure 2.9: Correction α of segment orientation $\vec{H}$ due to tropism $\vec{T}$

Resources:

1. Prusinkiewicz, P., & Lindenmayer, A. (1996). The algorithmic beauty of plants. Springer. http://algorithmicbotany.org/papers/abop/abop-ch1.pdf

2. Wikimedia Foundation. (2021, July 13). *Aristid Lindenmayer*. Wikipedia. Retrieved March 30, 2022, from https://en.wikipedia.org/wiki/Aristid_Lindenmayer

3. Sher Minn Chong's *Plants are Recursive*: https://www.youtube.com/watch?v=0eXg4B1feOY

4. Rawin Viruchpintu and Noppadon Khiripet https://www.bioquest.org/products/files/13157_Real-time%203D%20Plant%20Structure%20Modeling%20by%20L-System.pdf