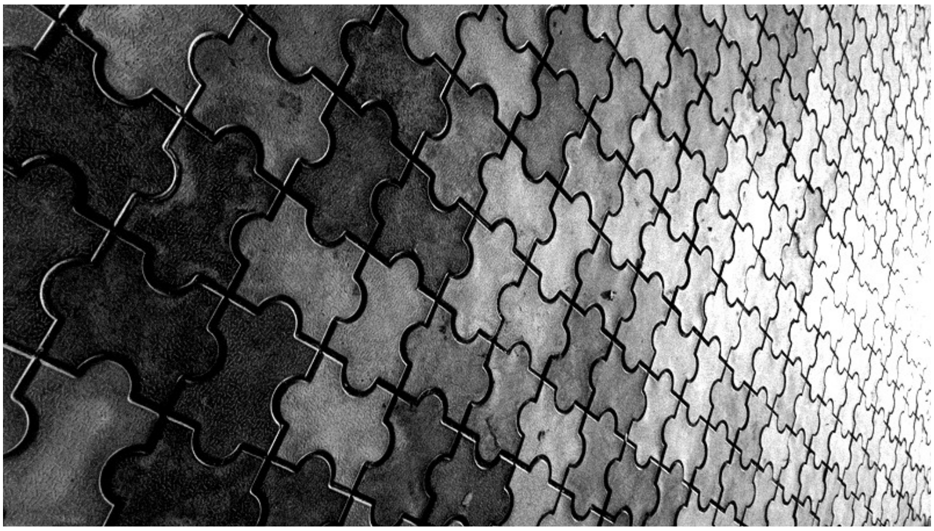
The Structured Puzzle

by tatasz

A bit more on Structured IFS. Make sure you read the intro: Structured IFS Fractals - Basic Elements
This is just the basic ideas behind the stuff, technical aspects will be explored in other tutorial.

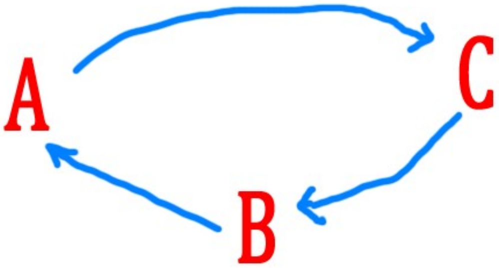
The Puzzle

The main idea of creating structures IFS works is that you have some non overlapping fractal pattern with holes, and you fill the holes. If there aren't enough holes, you add some and then fill them all too. Think of it as a puzzle: you have an empty space and need to find a piece that fits - or maybe several smaller pieces - or maybe replace one piece with another...



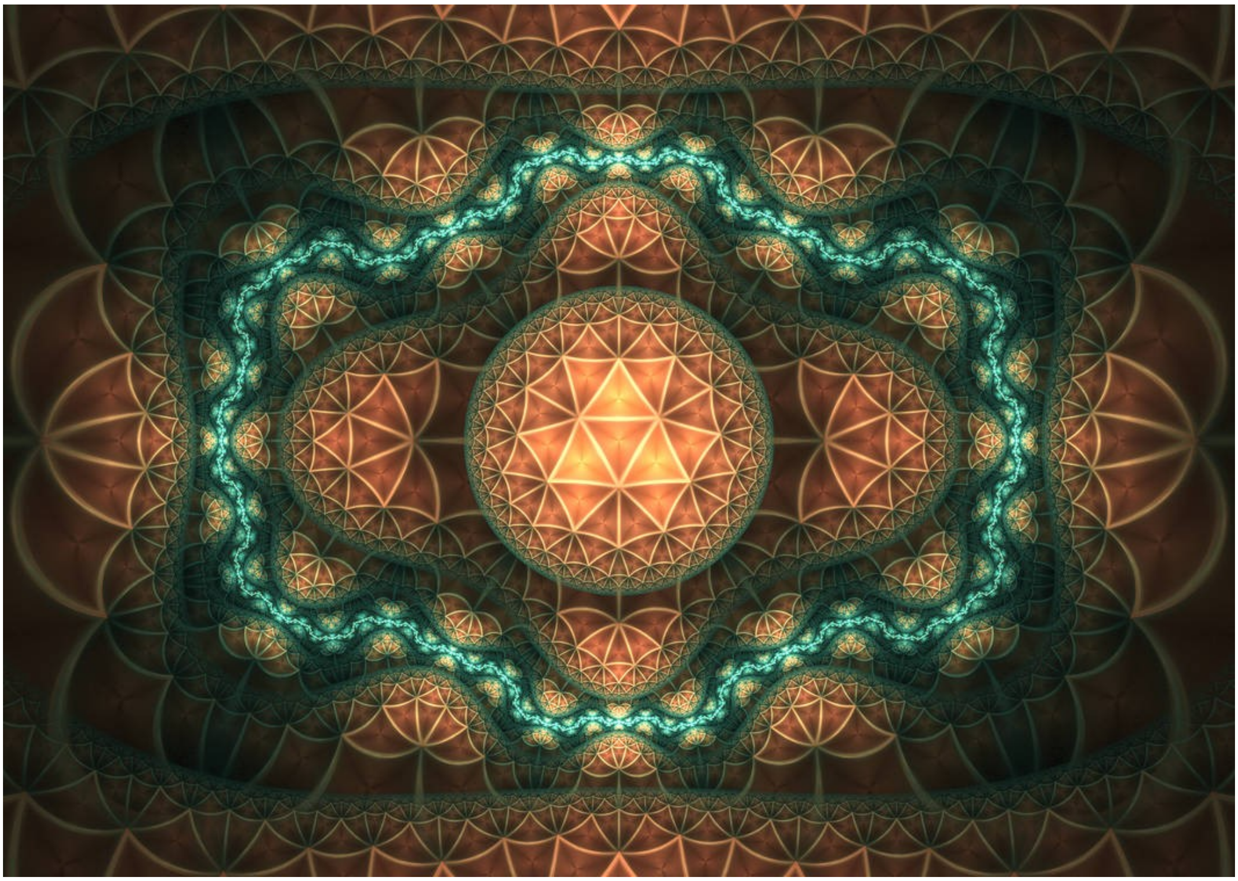
In this tutorial series, I will try to focus on the puzzle aspects of the Structured IFS: some common puzzle "pieces" and how to connect and mix them. Of course, will explain some of the technical aspects too 😁

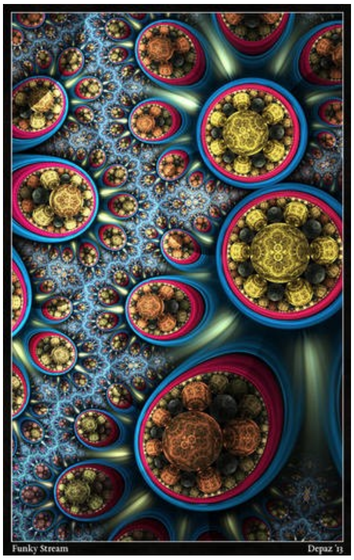
This is just an overview, with examples and blablabla. The basic idea, as it was said, is to fill a hole in fractal structure (lets call it A), with some element B. The element B, in its turn, can have holes to be filled too, so then you can put some C in there. Like this:



Lets take a look at some examples. They all feature hypertile as element B.

hypertile -> julian (left) and hypertile -> bipolar (right)





Now, there is an issue: you see that we have a flow from C to B to A - but in Apophysis, you always must have a closed loop, so points from A must go back to C somehow. Could be just a straight A -> C, or something more complicated, like A -> B and then B -> A.

Sometimes, this is easy, as you can use A to fill some holes in C. For example, the 3 different elliptic splits below roughly follow the pattern C -> B -> A -> C.
Look at the extreme left and right ones - you can clearly see them inside the 2nd and 4th stripes.

It is not always possible though: sometimes A just doesn´t fit in C at all. Below, two examples of hypertile -> other transforms -> glynnsim. In this case, it is very not trivial to just link glynnsim back to hypertile without producing an ugly mess (because of the hypertile symmetry requirements). So an extra blurry element is added

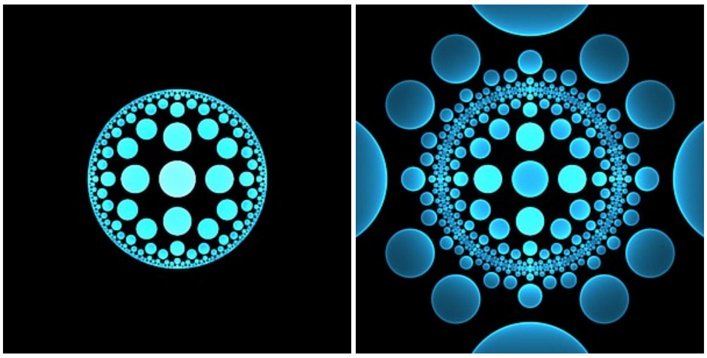
hypertile -> bipolar -> linear tile -> unpolar -> glynnsim -> blury element

The blur "hides" the glynnsim, so it is easy to send something symmetrical into the hypertile.

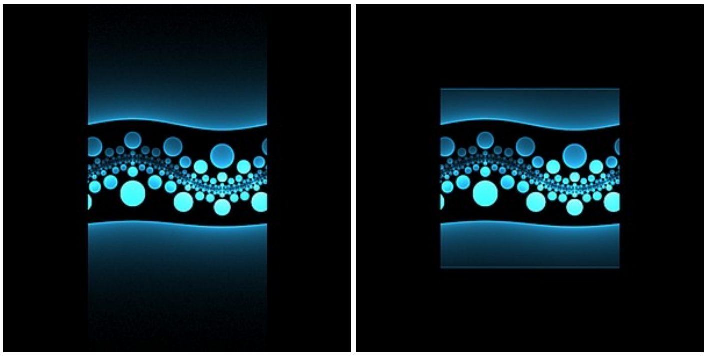
So, we have something like C -> B -> A -> *** -> C, where *** is something blurry or heavily distorted.

The next example is based on Circular Flame Tutorial or Unpolar+GlynnSim. Mind that this is not an explanation on how to implement the set up in practice, but just an illustration of how the puzzle works.

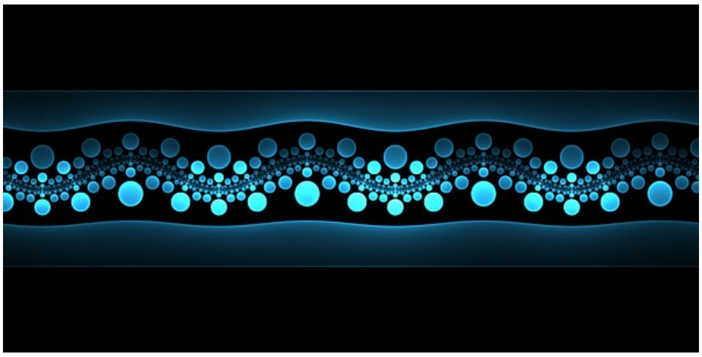
Structure 1: We start with a hypertile (left) and then link 2 sphericals to it, similar to Linked Transforms example (right).



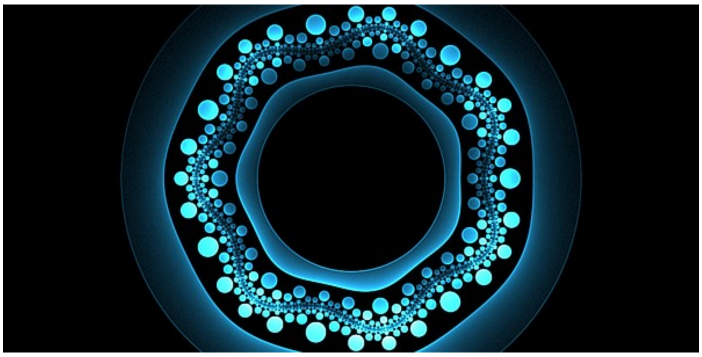
Structure 2: Now, we have a hypertile -> polar2 (left), which is cropped for tiling purposes (right).



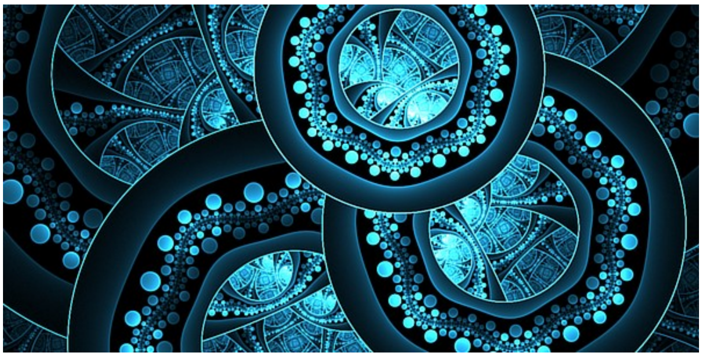
Structure 3: Now, the previous element is tiled horizontally.



Structure 4: The stripe obtained in the previous step goes now into unpolar.



Structure 5: Finally, we use unpolar to fill a glynnsim. See that as the bubbles in the hypertile are blurry, we can just link glynnsim back to the bubbles.



So we have basically a hypertile -> polar2 -> bipolar -> linear tile -> unpolar -> glynnsim