Shared linked transforms

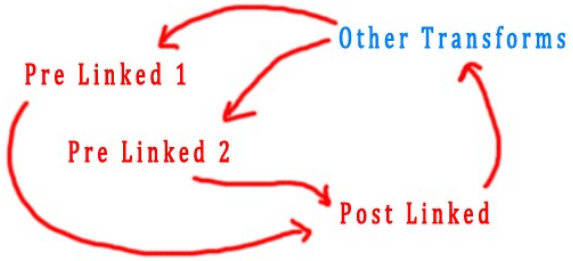
by tatasz

Sometimes, instead of just one pre linked or post linked transforms, you have multiple ones. This usually happens when your pre or post element is not just one piece, but several - for example 2 half planes, or several squares.

The xaos settings are similar to ones for simple pairs of linked transforms:

1. Pre linked transform(s), "from" weight modifiers can assume any values, as long as condition 2 is respected (so "from" pre linked transform themselves are 0)
2. Pre linked transform(s), "to" values are all 0, except the one to the post linked transform(s)
3. Post linked transform(s), "from" values are all 0, except the one from the pre linked transform(s)
4. Post linked transform(s), "to values can be anything, as long as condition 3 is respected (so "to" post linked transform themselves are 0)

For example, on the fluxogram below, a shared post linked transform with 2 pre linked transforms:



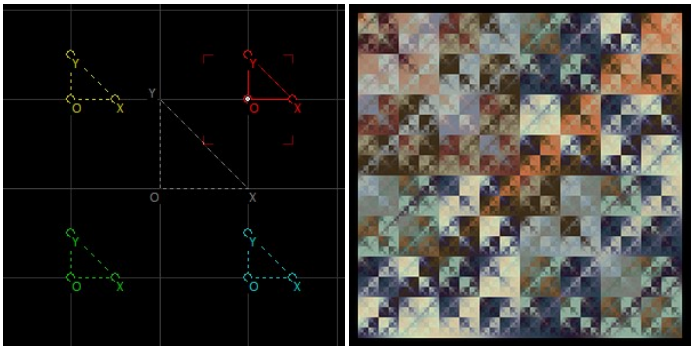
Also, a shared linked transform can be replaced with several identical linked transforms. For example, on fluxogram above, we can add a post linked transform to each of the pre linked ones, and have identical setup for both post linked ones.

Linear tile with linked loonie

Lets start making a simple linear tile (you may learn a bit more here: Linear Tile Tutorial).
You need 4 transforms, with linear = 1. Scale all 4 down 200% (so they all are half of the original size). Now, position them as follows:

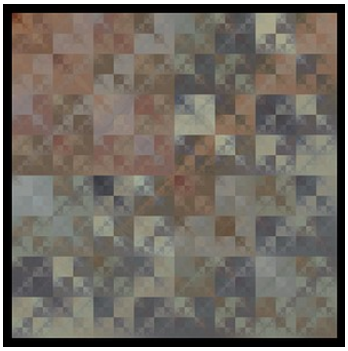
- transform 1: 1, 1
- transform 2: -1, 1
- transform 3: -1, -1
- transform 4: 1, -1

Keep in mind the order doesn't matters, so as long as the transforms take all those position, it doesn´t matter where is each one. You should have something like this (change or randomize colors to obtain the preview image):

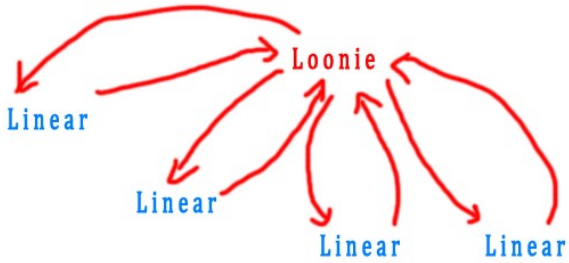


Parameters here: Square Linear Tile

Now, we want the pattern to be less square, by adding a loonie to it.
If we just add a loonie transform, it will overlap and we will lose the clean pattern (without getting any cool loonie shapes):



So what we actually need is to add a shared post linked transform with loonie to it. The fluxogram looks like this:



On all linears, set all "to" weight modifiers to 0, and leave only one to loonie (transform 5) 1. On loonie, set the weight modifier to itself to 0, and leave others as 1.
On loonie transform, set color speed to 1.
Then, set the opacity of the loonie to 0 - or do the same with all 4 linears and leave the loonie visible.

Yay, tiled loonies!

You can take a look at the params here: Tile + Loonie

Note that this is equivalent to adding a linked loonie to each of the linears: try doing it and comparing the results (make sure you have same transforms visible though).
Or just look at this: 4 linked loonies.

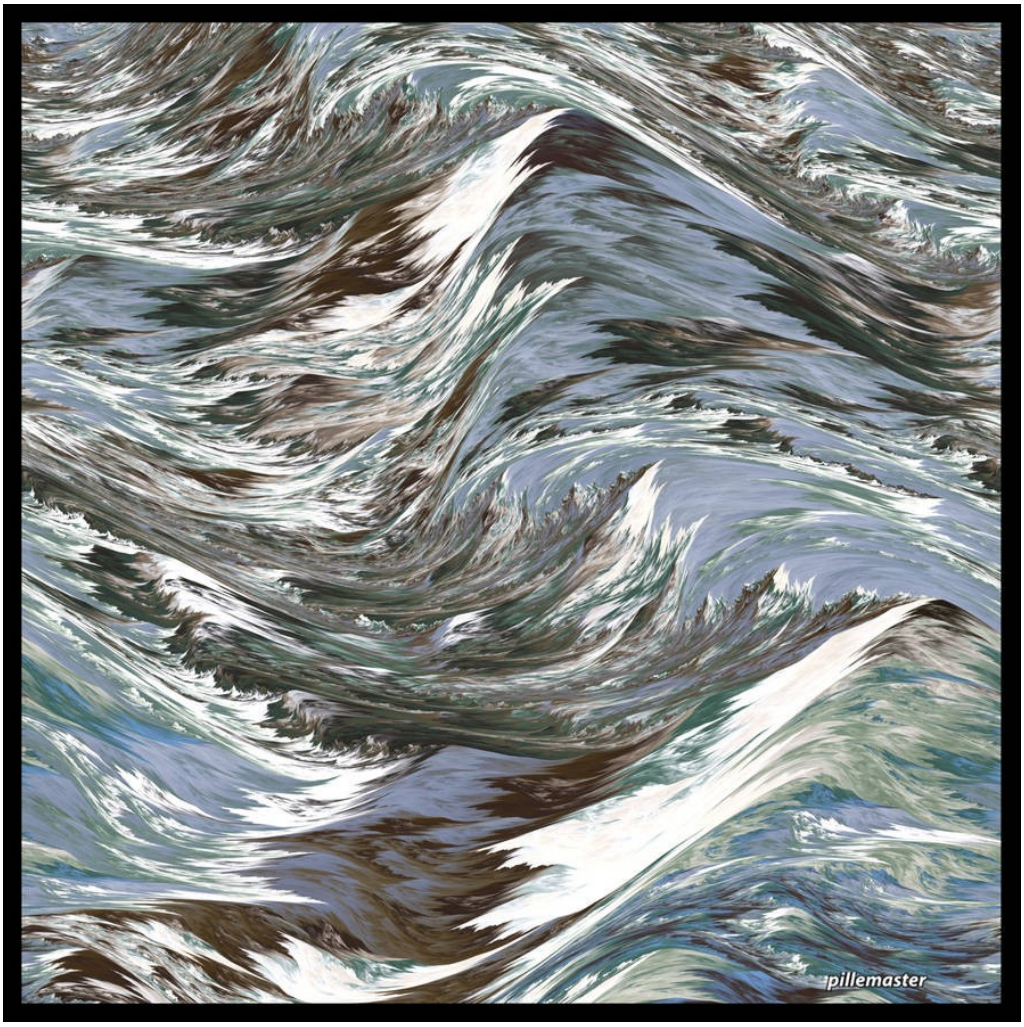
Last but not least, a lazy shortcut. Sometimes, setting the weights by hand can be a bit of a trouble, specially when dealing with complicated fractals with tons of transforms - apophysis slows down, taking up to several seconds on each change, and there are many changes to be made. So you can use duplicate transform to avoid it. Follow those steps:

- Starting from blank, scale down the transform 1 by 200% (this way, you wont have to scale the others ^^)
- Add a linked transform to transform 1, and replace the linear with loonie on the linked.
- Duplicate transform 1 three times, so you have 4 linear transforms overall and one loonie.
- Position the linears as described in the beginning of this example.

And you get exactly the same result as in previous cases, but with way less setting up.

Inspiration

Just a few inspiring examples:



This one is linear tile -> circlize -> target -> squarize