
JULIA UNCOVERED v.2.1



An Intermediate Guide
for Apophysis 2.05b2

Henceforth, Claire Jones is referred to as the “author” and the reader as the “user”.

All information and images contained within this guide are protected work and **may not** be altered by any means, electronic or analog, unless the author has given explicit permission to do otherwise. The guide **may** be placed on a user’s personal site and offered for public download at no cost, provided the original ZIP file and its contents remain unaltered.

Any flames/images produced with the help of this guide are considered the user’s property and may be used in any manner, including commercially. No permission is needed to use such flames.

Any images produced from the flame pack included in the ZIP file may be used in any manner, including commercially, as long as the new flame is somewhat modified from the original (a simple gradient change is not significant modification whereas transform tweaks are). Etiquette suggests the user credit the author as the flame’s source and not claim as the user’s own work.

Introduction

You've probably seen them around. You know, those spectacular Julia formations that pop up on deviantART every once in awhile. You want to know how it's done, but you can't just walk up and ask someone to spill their secret. This is where I come in. I spent a lot of time working on these "Grand Julians" and now I'm going to pass on what I have learned to the Apophysis community.

Before I describe the process, I will give a bit of history behind my discovery. After seeing one in particular by fellow deviant Zueuk and spurred on by Carl Skepper's Julia tutorials, I decided that I wanted to know the secrets behind the technique. So, one day I opened up Apophysis, messed around a bit and created a Grand Julian. I wish.

The process took months to discover and there were many, many missteps before I finally had the "Matrix moment", as I call it. I'm going to describe the journey for several reasons. It's a sort of proof that I really did figure out this completely on my own. Also, everyone who has spent time in Apophysis has their own way of developing techniques and I will give some insight on how I do it. Very few fractal artists are able to open the application and make something amazing, although apparently there are some. In general, we spend many hours – or sometimes weeks and months – working with triangles to develop our own techniques.

I have a confession to make. I hate triangles. Sure, it's fun to mess around with a couple of triangles, but the fewer out there on the *Transform Editor* the better. Every time a triangle is added, it builds off of the others making it difficult to keep track of each element. I've seen starter flames that have had over eight triangles and I have no idea how people can keep track of what everything does. This is why I spent so much time developing the *Two Transform* method described in my other tutorial. Fortunately, while Grand Julians use more than two triangles (generally 4-6), the flame is created in an altogether different method than normal flames.

Surprisingly, this technique is incredibly simple. Mind-bogglingly simple. The good news is that this technique doesn't rely on finding the "perfect flame". Just pick any random flame, clear it out and get started. I regard this an intermediate guide simply because I have already written a tutorial that described the basics of Apophysis and didn't want to include that information again in this manual. Basically, you already need to have working knowledge of the *Transform Editor* and a few of the other features before beginning. There are several tutorials that describe the process listed in the *Links* section.

Second Introduction

Times change and so does Apophysis. I thought that I might need to begin a tutorial rewrite when I noticed that Grand Julians created in Apophysis version 2.04b were rotated oddly in both the later Joel Faber and Gygrazok (Apophysis Z+) versions. After the release of the 2.05b/2.05b2 versions and the situation remained, I *knew* I had to rewrite *Julia Uncovered* to address the changes. But that's not the only thing I've included in this new (*much* larger) guide.

Oh no.

In this version I've included many extras, including options to create visually appealing centers with different variations other than Blur, *two* additional Grand Julian techniques, "fixing" Julians created with Apophysis 2.04b and even a mini-glossary. There are also far more images – roughly 90 – to help the user understand the Julian process better. I was initially hesitant to include so many images, considering that the end result would be a tutorial large in file size. I generally do not like to offer bulky downloads out of respect to dial-up users. However, I felt that I needed the images to fully outline the Julian creation process.

Hopefully, this improved *Julia Uncovered* will further users on their Apophysis journey. Good luck!

Claire
January 2007

Table of Contents

INTRODUCTION	<i>ii</i>	JULIAN TECHNIQUE 2	27
SECOND INTRODUCTION	<i>iv</i>	Step 1	27
THE JOURNEY	<i>1</i>	Step 2	29
THE JOURNEY CONTINUED	3	Step 3	31
THE TRANSFORM EDITOR AND YOU	4	JULIAN TECHNIQUE 3	32
THE PROCESS	8	Step 1	32
Step 1	8	Step 2	33
Step 2	9	Step 3	34
Step 3	12	Step 4	35
Step 4	13	Step 5	36
Step 5	15	CENTER OPTIONS	38
Step 6	17	Spherical	39
Step 7	19	Swirl	40
Step 8	21	Handkerchief	41
Step 9	22	Julia	42
FINISHING TOUCHES	24	Bubble	43
EXAMPLES FOR JULIAN TECHNIQUE 1	25	“FIXING” JULIANS	44
Julian 2	25	GLOSSARY	47
Julian 3	26	LINKS	50

The Journey

I never had much interest in creating Julias until I read Carl Skepper's tutorial. I tried the various techniques described, but I had difficulty making anything spectacular. When I first saw "Majestic Julia" by Zueuk, though, I knew that I wanted to make that style of Julia. I saved the render and would sit at the computer just staring at it, trying to figure out how it was created. Not only was the render bright and clearly detailed, but also I didn't understand how to form the "side bars". More frustration came when I discovered that other Apophysis users had managed to figure out the technique.

I worked on Julia basics following Carl's tutorial to the letter, though few of those renders met with any success on deviantART. For whatever reason, I kept working on Julias in addition to working on the *Apophysis Guide v.2.1* rewrite. One break came when I read Zueuk's tutorial on *Blob Spirals* described at FracFan Forums (information in the *Links* section). At one point, he mentions setting a variable to a negative value. In a moment of clarity, I built a Julia by Carl's method and set the last transform to the largest Julian variation variable of the group (at that point set randomly to Julian 2). Then, I set the `julian_power` to 2 and the `julian_dist` value to -1. Voila. I now had a Julia with sidebars.

However, it wasn't a Grand Julian. There were "ghost" images all over the place and the main area still lacked detail. I commonly used up to 9 transforms to compensate. I would build new Julias over and over again without success. Along the way, I did discover the goodness of rotating transforms, which will be discussed later. Carl had mentioned this in his tutorial, but I hadn't really understood the purpose at the time. The Julia would generally look good, but as soon as I added the last transform to form the sidebars, the flame would become distorted. I couldn't figure out why this happened. Everything was fine, then a total disaster with the addition of the last transform.

As time went by, I began to wonder that I was going about the process in a way that was fundamentally incorrect. I had relied so heavily on Carl's tutorial that it had hindered my ability to understand what I had set out to accomplish. I realized finally that I needed to set what had been the last transform *first*. It isn't that one *really* needs to place this transform first, but rather it is far simpler to design the Julian by starting with the basic framework first then building the inside details second as opposed to forcing the framework on the Julian last.

I opened up a new flame and set Transform 1 to Blur and Transform 2 to Julian variation 1, variables `julian_power` 2 and `julian_dist` -1. Adding a couple more transforms produced a Julian much more similar to what I had sought. Unfortunately, while this had fixed the distortion of my previous attempts, the

problem of ghost images still remained. I worked about two weeks straight trying to figure out what I was missing. It was a frustrating time, needless to say, but I kept trying. I had already devoted too many hours on this technique and I knew I almost had it.

I came up with all sorts of ideas, including altering transform weights. Still, nothing worked. A couple of weeks later, I finally remembered that I had wanted to try increasing the transform weight, but never gotten around to actually *applying* it. I had read a discussion on the Internet where someone had commented that “plastic” flames needed to have transforms with weights of about 10.

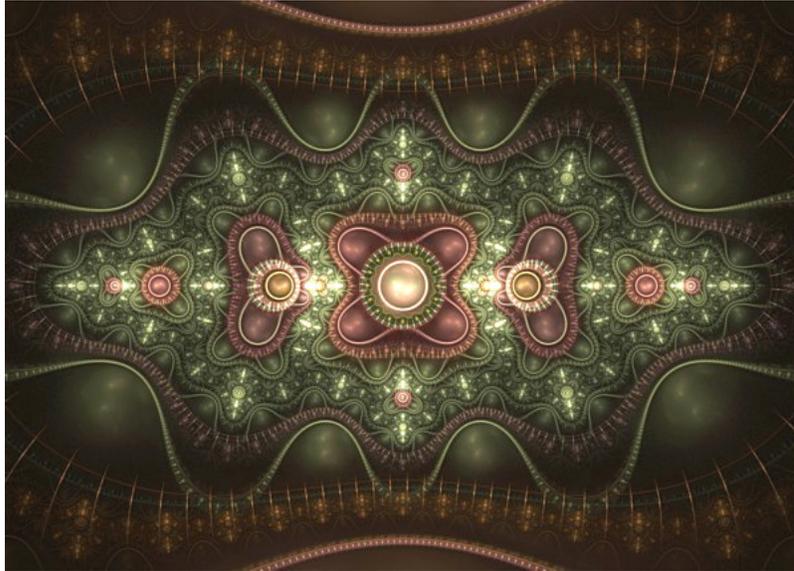
I realized that could be the missing element to the Julia conundrum. I set Transforms 2-4 at a modest weight of 1 (Transform 5 was left at 0.5 for some reason) and got the image below. I stared a bit, amazed that I had finally figured it out.



It was by no means perfect and I still needed to figure how to properly rotate the transforms, among other things, but it was a start.

The Journey Continues

If I spent a couple of months struggling through the basics of a basic Grand Julian, I probably spent about six months trying to figure out the alternate technique, shown here below, to no avail.



Michael Faber was kind enough to explain it to me once, but I couldn't put his advice into practice. I was already attempting something similar to what he had suggested, without any luck. There was also a thought in the back of my mind that figured I would write a tutorial about the technique once I finally discovered it, so I thought it best to struggle through on my own rather than rely on someone else's suggestions. I developed another technique to compensate, which is also outlined in this guide, but it was not what I wanted.

Oddly, it was not until I sat down to rewrite *Julia Uncovered* that I inadvertently moved my triangles into the proper position and achieved the look I had worked to create for so many months. It was a happy moment (and one of great relief to finally get "it"). Ironically, most people were rather blasé about my discovery, thinking I already knew how to do it. Oh well. I have it now.

The Transform Editor and You

I know you want to jump right into flame making, but there are a few basics for Grand Julians that you need to know and fully comprehend before beginning. Carl had mentioned the following items in his tutorial, but I didn't really understand what the two Julian variables did until I worked with them a bit more. I generally prefer the Julian variation, but you can try using Julia and Juliascope as well. It will create a very different flame, however, than the technique outlined.

Terminology:

For clarity, I need to explain my terminology. In the past, I referred to transforms as “triangles”. While I still personally refer to them as triangles, I have decided to call them by the official designation.

Julian comes from Julia (n), but because it is listed as Julian in the *Transform Editor*, I refer to them as such. Sidebars are the two areas flanking the center area, either vertically or horizontally, depending on how the flame is rotated. Julians tend to render as little “circles” around a center point and this is how I refer to them.



Examples of Julian “circles”

Transform Movements:

A note must be made about transform movements. Moving a transform upward from the (0,0) coordinate (i.e. 0, 0.2 or 0, 0.5) on the graph results in the circles traveling into a cross or “+” shape. On the other hand, moving a transform below the 0,0 coordinate (i.e. 0, -0.1) results in the circles moving into an “X” shape pattern. Traveling right or left on the graph moves the circles in the opposite direction.

Note that the transform movements now differ from the original 2.04b version of Apophysis. The new movements hold true for Apophysis Z+, versions of Apophysis above 2.04b and the Joel Faber Apophysis. In a later chapter, I will show how to restore an existing 2.04b Julian with the new form.

To better understand transform movements, look at the examples below. The first shows Transform 2 with the 0 apex at the (0, 0.3) coordinate. In the second example, the 0 apex is moved down to (0, -0.3). Notice also that the entire flame has been rotated.



Transform 2 at (0, 0.3)

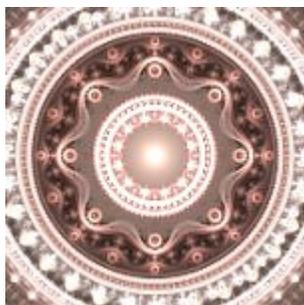


Transform 2 at (0, -0.3)

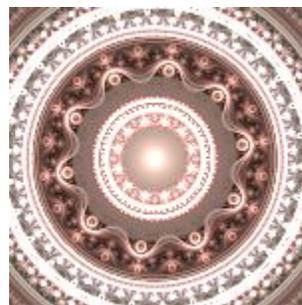
julian_power Values:

The *julian_power* value under the *Variables* tab determines how many circles will appear for a particular transform. To understand this better, take a close look at the examples below. In general, double the value of the *julian_power* and that is how many circles are visible.

In the first example, Transform 3 has a *julian_power* of 4. This is apparent by eight prominent circles around the center. In the second example, the value of *julian_power* is raised to 5. As a result, the number of circles around the center has increased to 10.



Transform 3 with a julian_power of 4



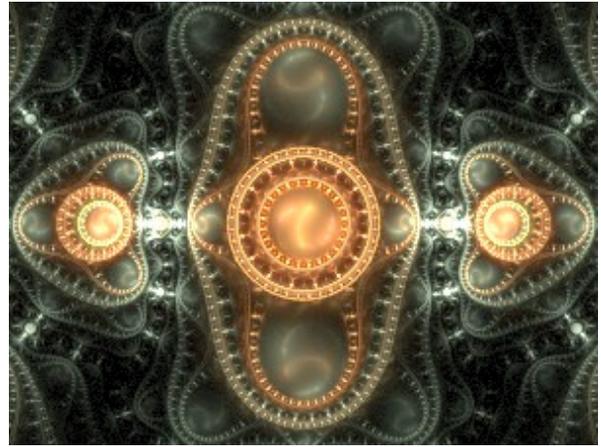
Transform 3 with a julian_power raised to 5

For beginners, I recommend using variables above four. Values four and below produce odd results which are interesting, but need to be tackled after the basics are understood.

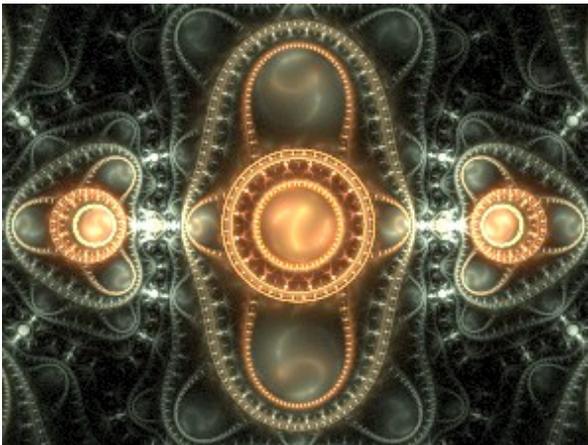
Additionally, the lower the value of the `julian_power`, the thicker and denser the resulting layers. Setting the `julian_power` to values above 25 will create layers that are nearly a line. In the first example below, Transform 3 has a `julian_power` of 6. Over the next few images, the value is steadily increased to `julian_power` 50. All other transforms remained unchanged.



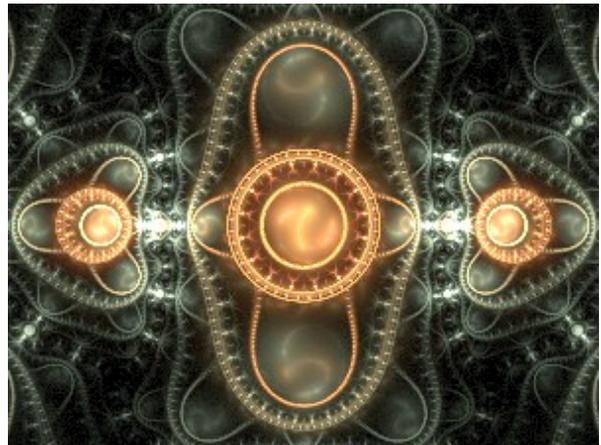
Transform 3 with a `julian_power` of 6



Transform 3 with a `julian_power` of 12



Transform 3 with a `julian_power` of 25



Transform 3 with a `julian_power` of 50

`julian_dist` Values:

The second important variable is `julian_dist`. This figure has a multiplying effect for the length of the individual Julian circles. The value 0.5 distorts the circles by half, while a value of three multiplies the

length (but not the width) of the circle producing an elongated circle. A value of 1 is usually the default and 0 typically produces a line instead of a pattern.



A high julian_dist was used for one of this flame's transforms and results in elongated circles

Negative Variable Values:

One of the most powerful elements is the use of negative values. Not only does a negative value for the julian_dist. create sidebars, but they also have an overall effect of the flame depending on which transform has a negative value. It's often best to alternate a positive and negative value for Transform 3 and above.

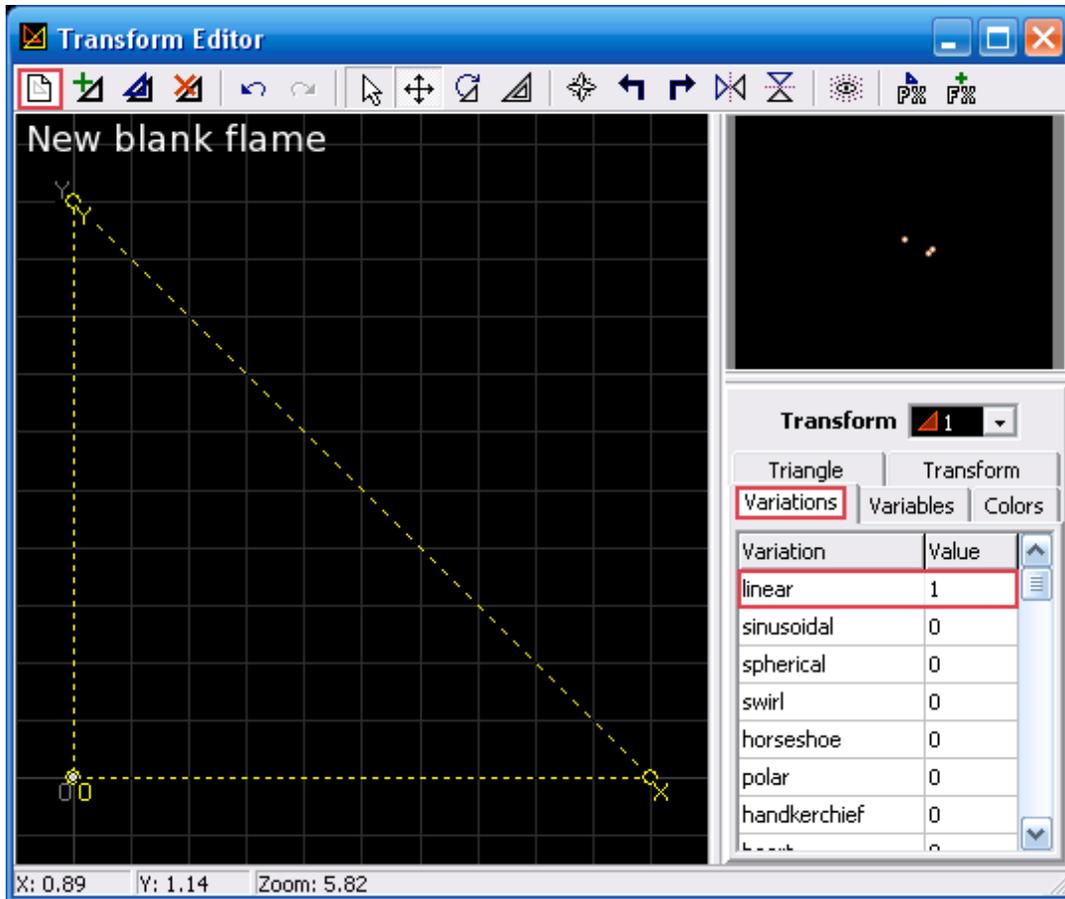


Yellow circles indicate positive values while purple circles show negative values

The Process

Step 1:

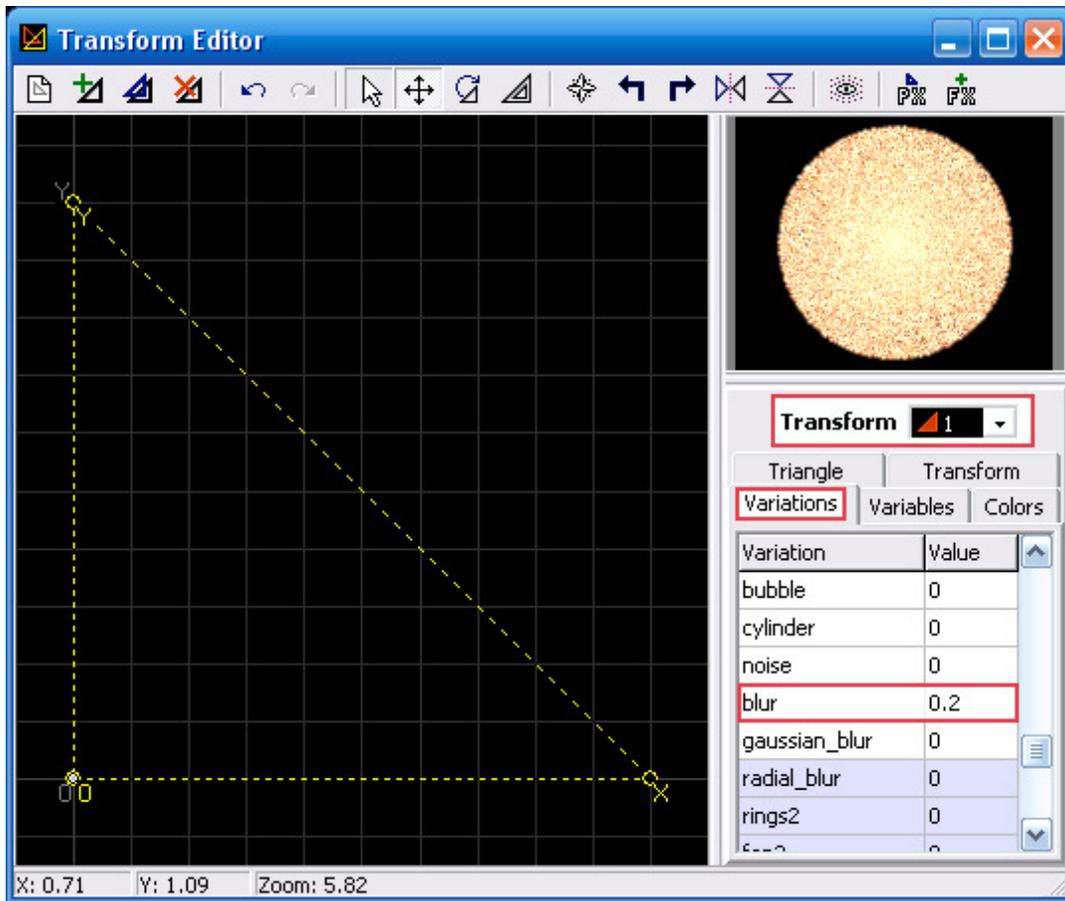
Take a flame, any flame, and clear it out. Now, there is a Transform 1 and a Transform 2, both with a variation of Linear 1.



It's very exciting.

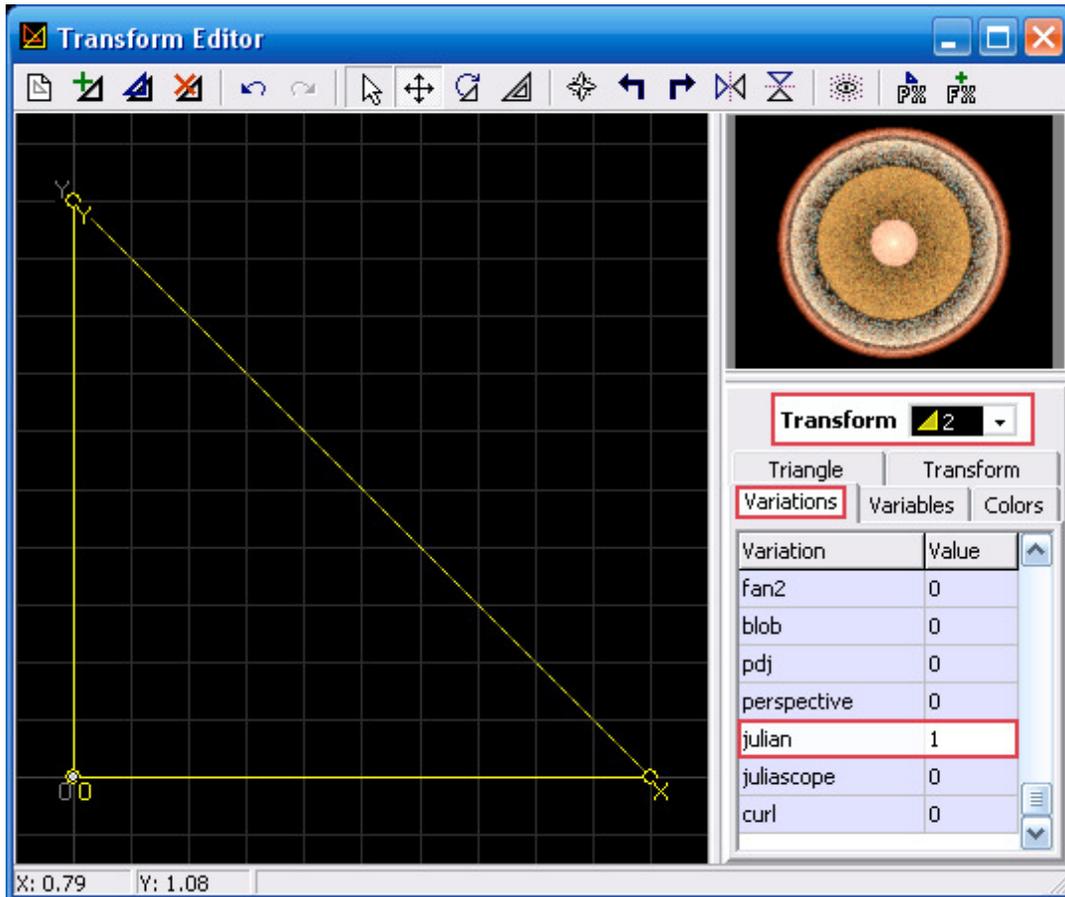
Step 2:

Change Transform 1 to Linear 0 and add Blur 0.2 in the *Variations* tab.



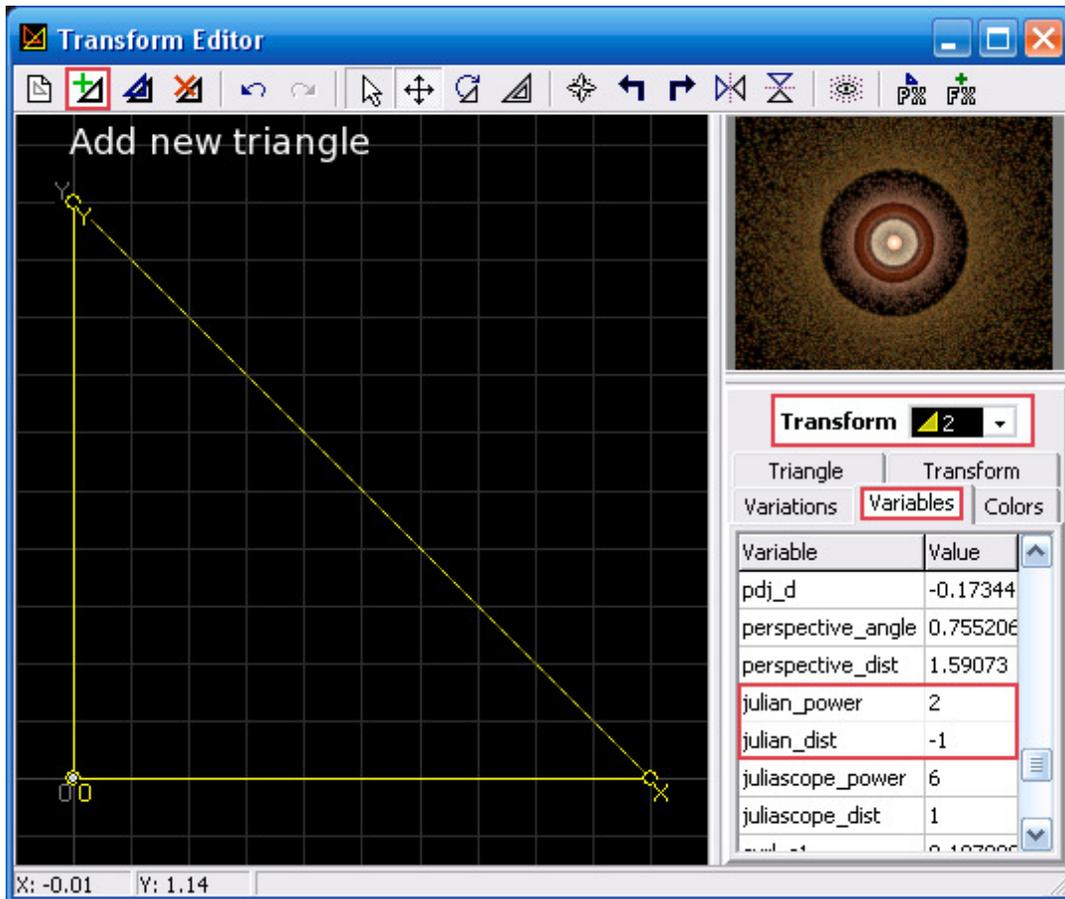
Step 2 cont'd:

Set Transform 2 to Linear 0 and Julian 1. The Julian 1 is mostly an arbitrary number, but one of the easiest values for Julians. Just remember not to go above the value of Transform 2 when adding additional transforms later.



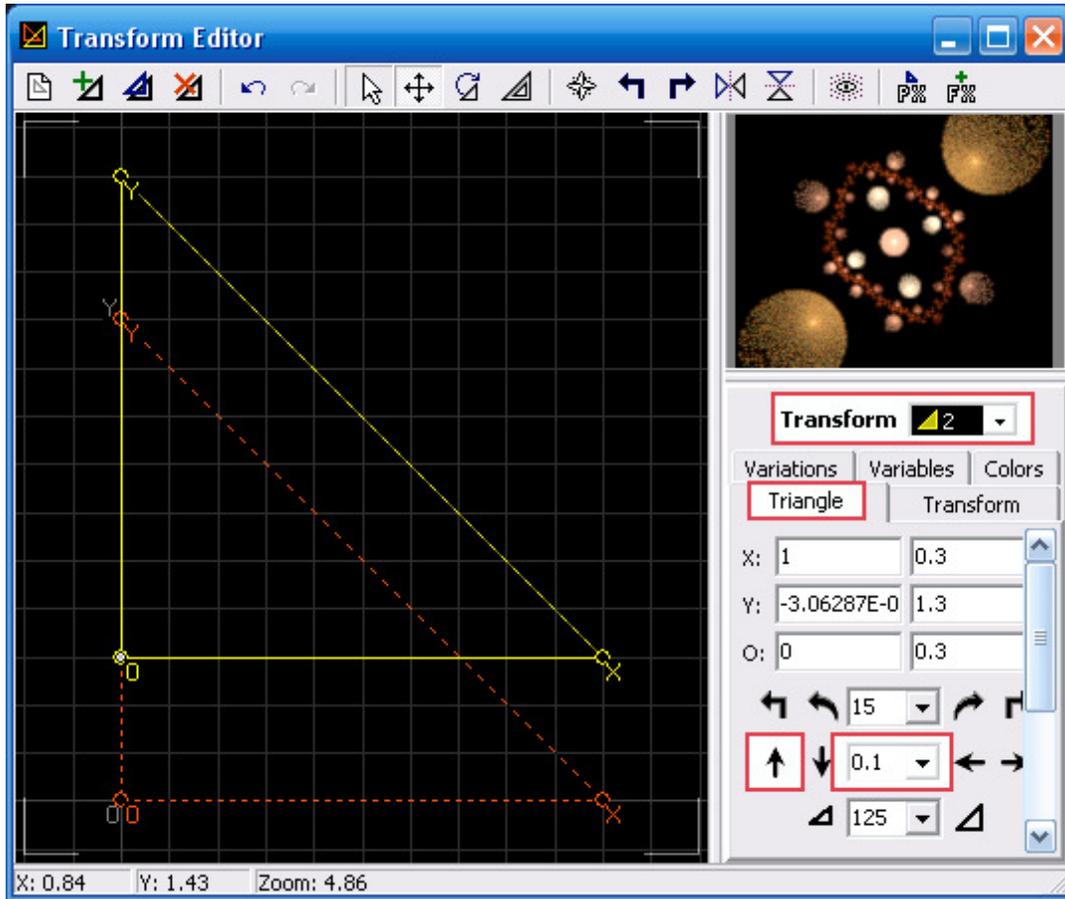
Step 2 cont'd:

Change julian_power to 2 and julian_dist to -1 under the *Variables* tab.



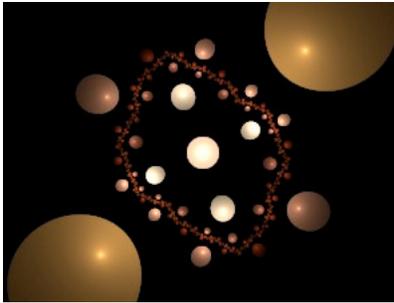
Step 3:

Set the *Move transform* up/down function to 0.1 (if leaving the value of Transform 2 at Julian 1, otherwise, you will need to adjust this accordingly). Move Transform 2 upward by three 0.1 units.

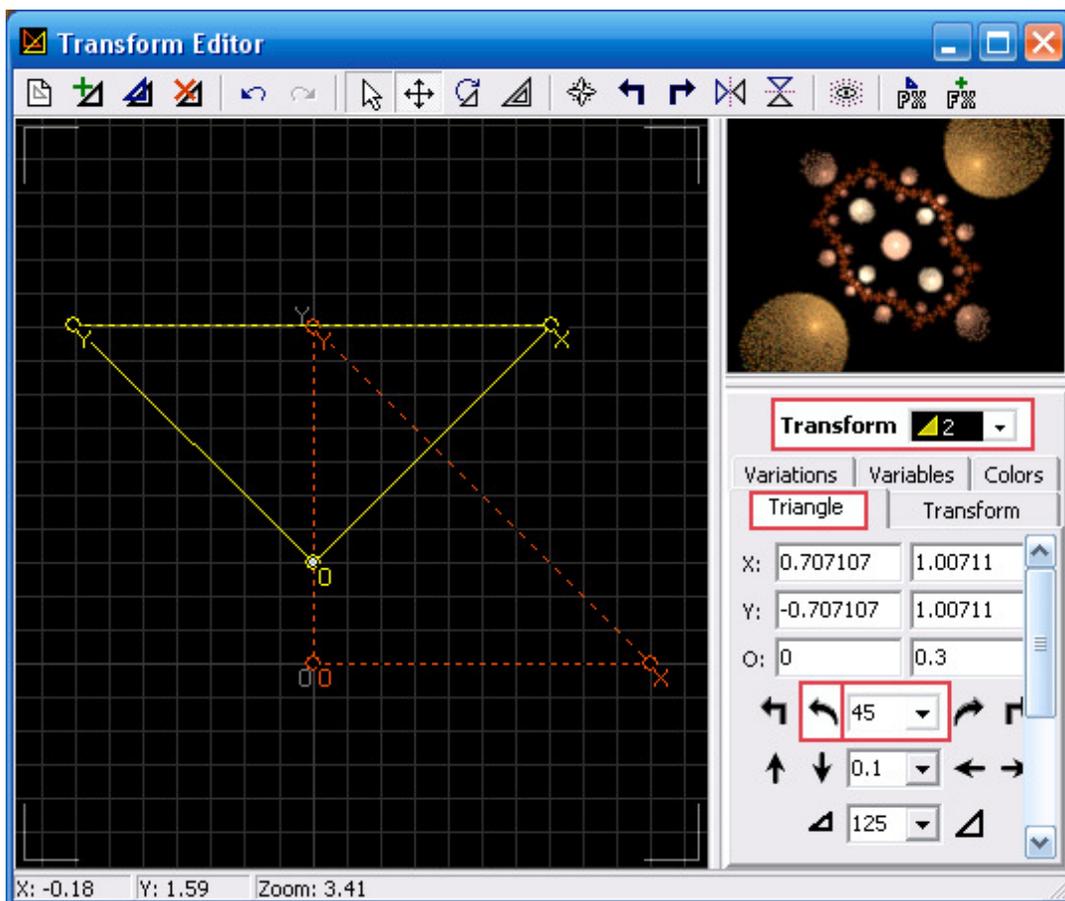


Step 4:

The basic shape of the Julian has begun to form, but it's at a 45-degree tilt and the inside is undesirable.

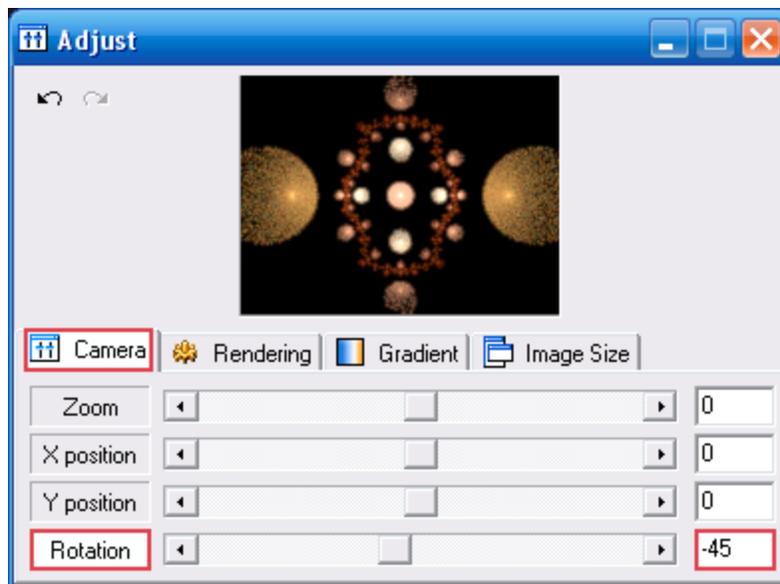
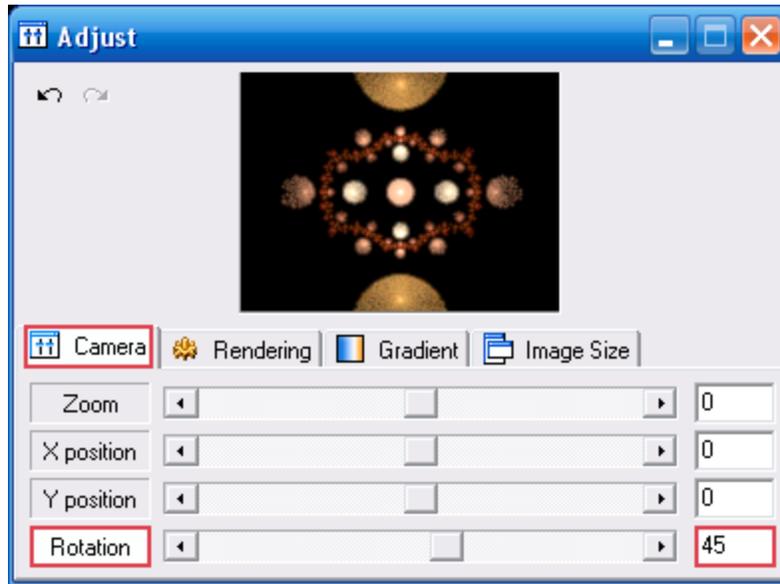


To rectify this, we need to do two things. To fix the inside of the flame, set the flame rotation under the *Triangle* tab to 45. Click on the "rotate counterclockwise" button once so that Transform 2 is positioned as shown below.



Step 4 cont'd:

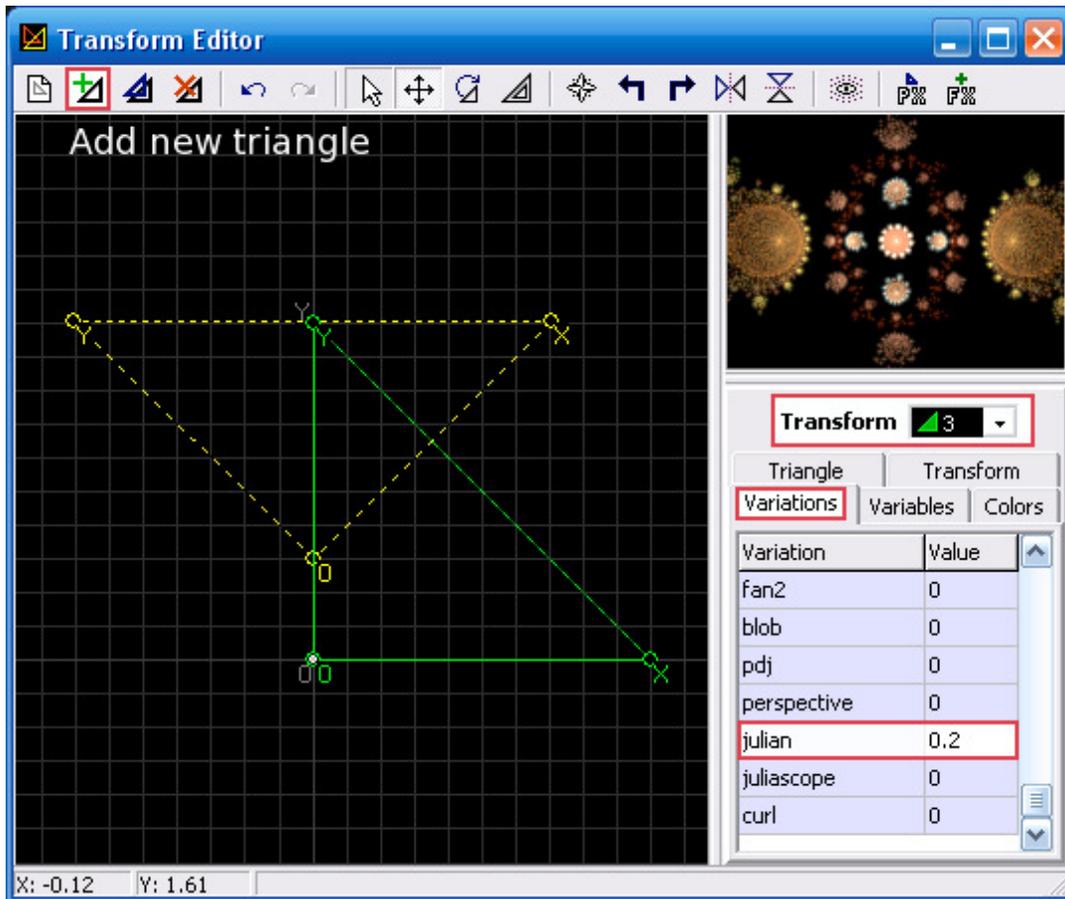
Second, change the *Rotation* under the *Camera* tab in the *Adjust Window*. Depending on which style of Julian is desired, rotate to either 45 or -45. The two different styles are shown below.



In the past, I suggested moving Transform 2 up/down to adjust the style of Julian. However, this method alters the Julian itself and it is much better to simply rotate it as desired with the *Rotation* function.

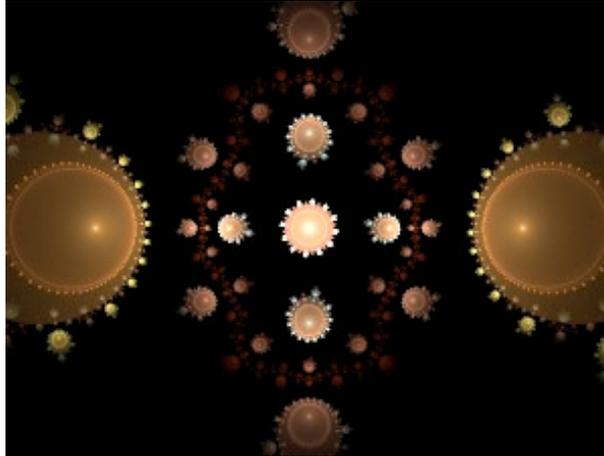
Step 5:

Add a new transform and set the variation to Linear 0 and Julian 0.2. Change the julian_power to 15 and the julian_dist to -1.

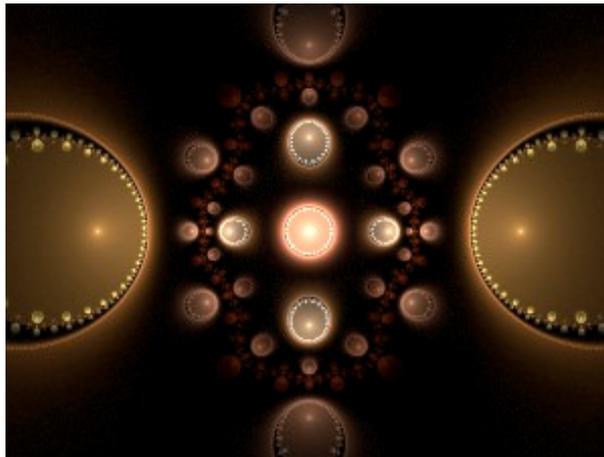


Step 5 cont'd:

Notice the small changes that occur when the variables are changed. Later you can experiment with the variables by increasing and decreasing the `julian_power`.



Flame after Transform 3 addition

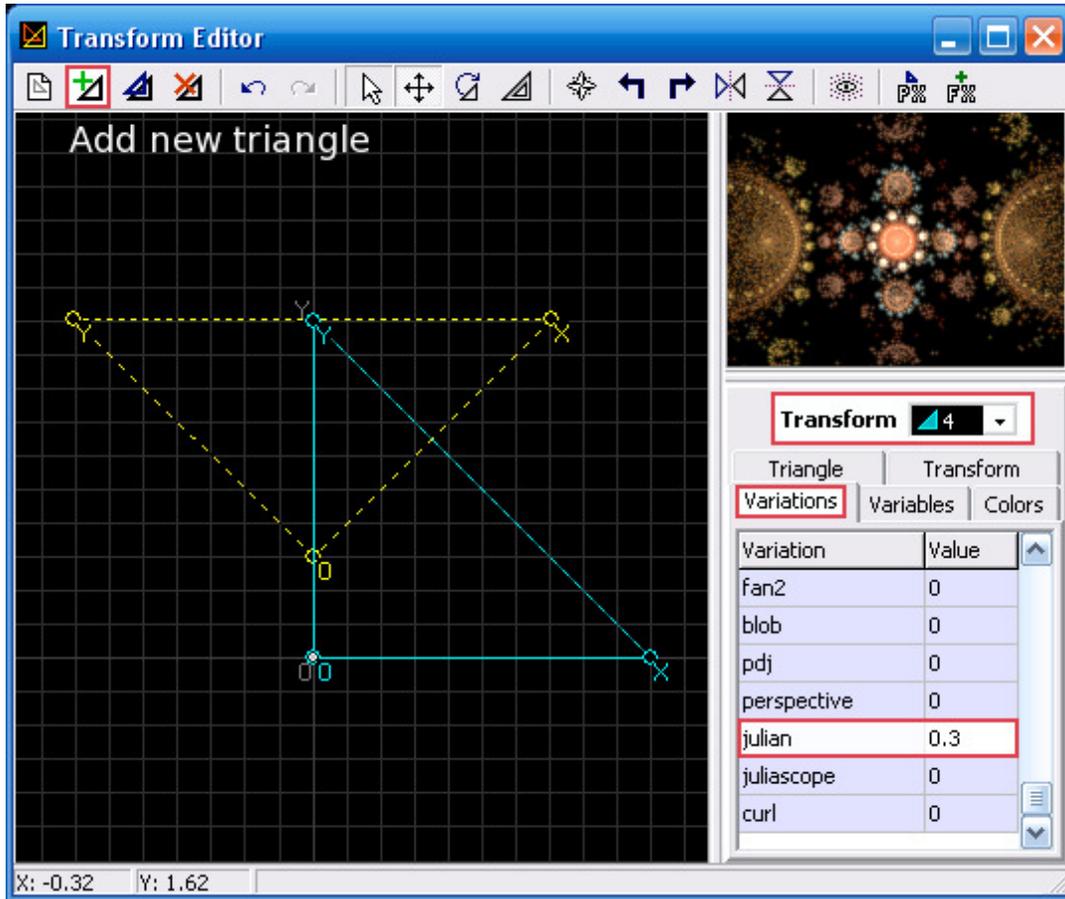


Flame after the variable alterations

By this point, your flame may look different than mine, such as lacking detail. Try shifting to a more colorful gradient or increasing the frequency in the *Gradient Editor*. In this basic Julian example, I relied on gradient `#667_Tryst`.

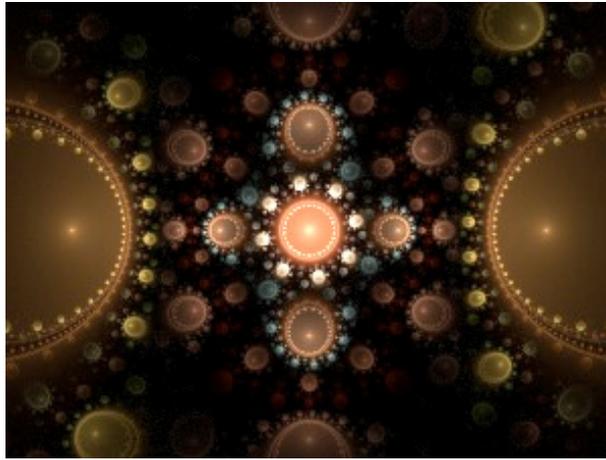
Step 6:

Add Transform 4 and set the variation to Linear 0 and Julian 0.3. This time change the julian_power to 8 and julian_dist to -1.



Step 6 cont'd:

As mentioned before, there are radical changes as the julian_power variable is altered.



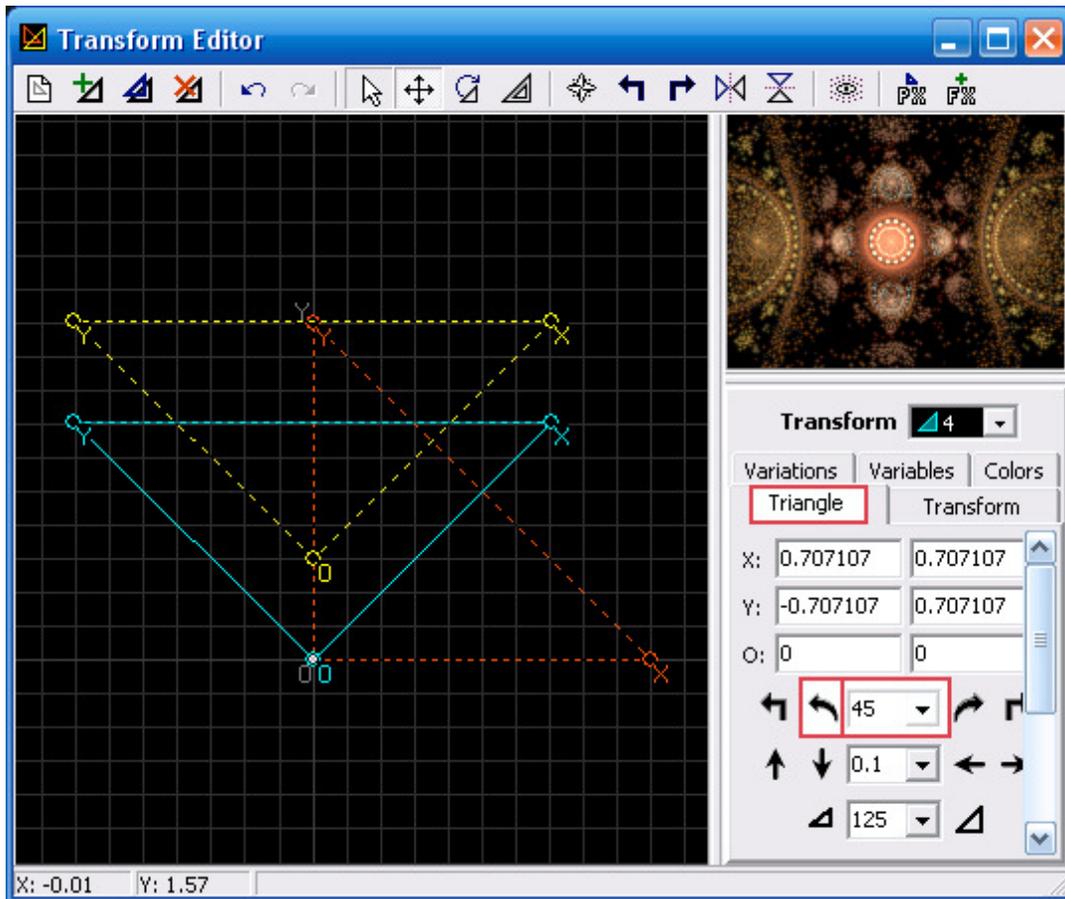
Flame after Transform 4 addition



Flame after the variable alterations

Step 7:

The Julian is beginning to look good, but it could be better. We help this by rotating Transforms 3 and 4 as in *Step 4*. I usually rotate counterclockwise once at 45° , which is what I have done here.



Step 7 cont'd:

Look carefully at the three examples. The circles within the main disk have shifted in response to the transform rotation. It's slightly difficult to see in these examples due to the high julian_power values.



Julian with no transform rotation



Julian with Transform 3 rotation



Julian with Transform 4 rotation

Step 8:

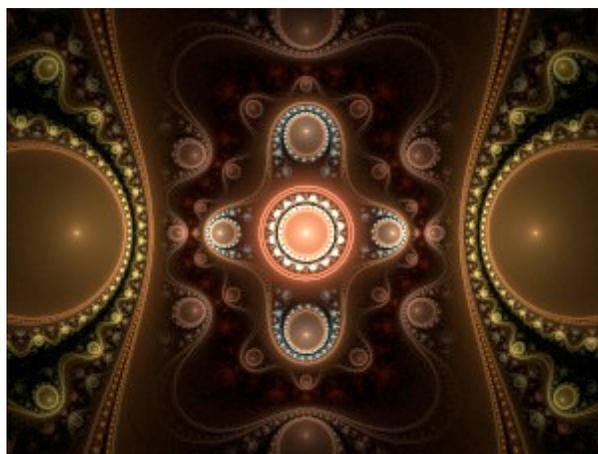
It's close, but still not right. The background is a bit difficult to make out. Here's the important step that took me the longest to discover. Set Transforms 2-4 to *Weight 1* (found under the *Transform* tab).



Flame after the Transform 2 weight increased to 1



Flame after the Transform 3 weight increased to 1



Flame after the Transform 4 weight increased to 1

Step 9:

At this point, I would generally increase the Transform 2 weight. If the background area is difficult to see, that generally indicates that the main Transform 2 weight needs to be strengthened. However, I am happy with the weights in this example as they are and decide to leave them for now.



Step 9 cont'd:

Notice that there is a halo effect caused by the Blur of Transform 1. If this is bothersome, change the `julian_dist` of Julian 3 to +1 to reduce it somewhat. Keep in mind that this does change the form of the Julian. Also, to combat the halo effect on Julians, try reducing the Transform 1 weight and/or *Gamma*, though changing the lowest transform to a positive value generally works well enough.



So, there you have it. This is the basic framework of a Grand Julian with only four transforms. It's actually quite amazing that the flame can look so put together just by altering the *Weight* value.

Finishing Touches

Additional Transforms:

This basic style of Julian generally benefits from additional transforms. Here, I have added a Transform 5 with a *Variation* of Linear 0 and Julian 0.35. I set the *julian_power* to 20, leaving *julian_dist* at 1 and *Weight* at 1. To balance the flame, I increased the Transform 2 weight to 2.



Gradients:

Grand Julians are incredibly sensitive to colors, so be sure to spend some time playing around with gradients. Try *Randomize Color Values* under the *Flame* menu to find additional options. This function randomizes the *Transform color* values found under the *Colors* tab in the Transform Editor.

When using additional variations to Blur in Transform 1, adjust the *Transform color* by setting the *Symmetry* to 0.2 – 0.5 and manually altering the color to maximize the extra colors present

Centering the Flame:

Before rendering, manually set the *X Position* and *Y Position* under the *Camera* tab of the *Adjust* window to a value of 0. This centers the flame in the *Preview* window. Be sure to complete this step *after* setting the *Master Scale/Zoom* to the desired values.

Example Flames for Julian Technique One

Julian 2



This is the result of the test run I did for the first *Julia Uncovered*. Originally, I used a value of -1 for the `julian_dist` on Transform 4 when I started working on transform variables for the guide. After I saw the result of a positive value, I changed my mind. The rest of the variables are identical to the values illustrated in the original *Julia Uncovered*.

	<i>Variation:</i>	<i>julia_power:</i>	<i>julia_dist:</i>	<i>Weight:</i>
Transform 1:	Blur 0.25	N/A	N/A	0.25
Transform 2:	Julian 1	2	-1	2
Transform 3	Julian 0.25	4	-1	1
Transform 4	Julian 0.5	15	-1	1

Julian 3



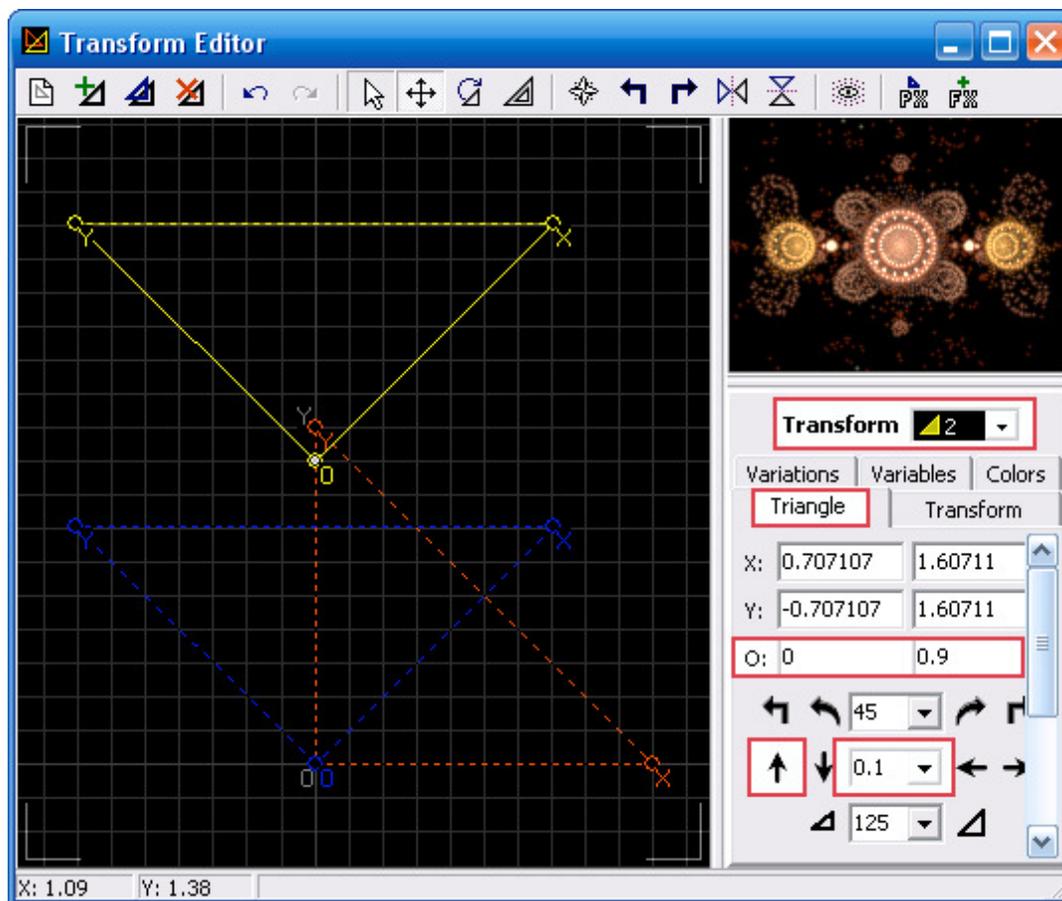
I like this example. Normally, I prefer strong backgrounds, but in this case I went with a gradient that darkened the area. The flame is a slight modification of Julian 2.

	<i>Variation:</i>	<i>julia_power:</i>	<i>julia_dist:</i>	<i>Weight:</i>
Transform 1:	Blur 0.2	N/A	N/A	1
Transform 2:	Julian 1	2	-1	10
Transform 3	Julian 0.25	4	-1	10
Transform 4	Julian 0.4	15	1	3
Transform 5	Julian 0.1	8	-5	10

Julian Technique Two

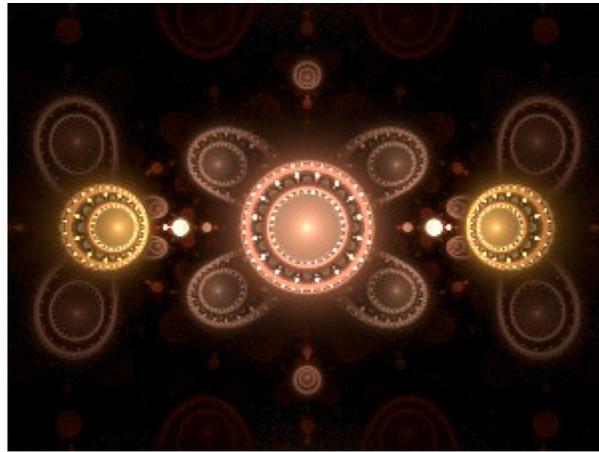
Step 1:

Begin with the final Julian containing five transforms of Julian Technique 1. Move Transform 2 up by six 0.1 units until the 0 apex is at the (0, 0.9) coordinate.



Step 1 cont'd:

The flame is very dark and the Julian pattern is difficult to see.

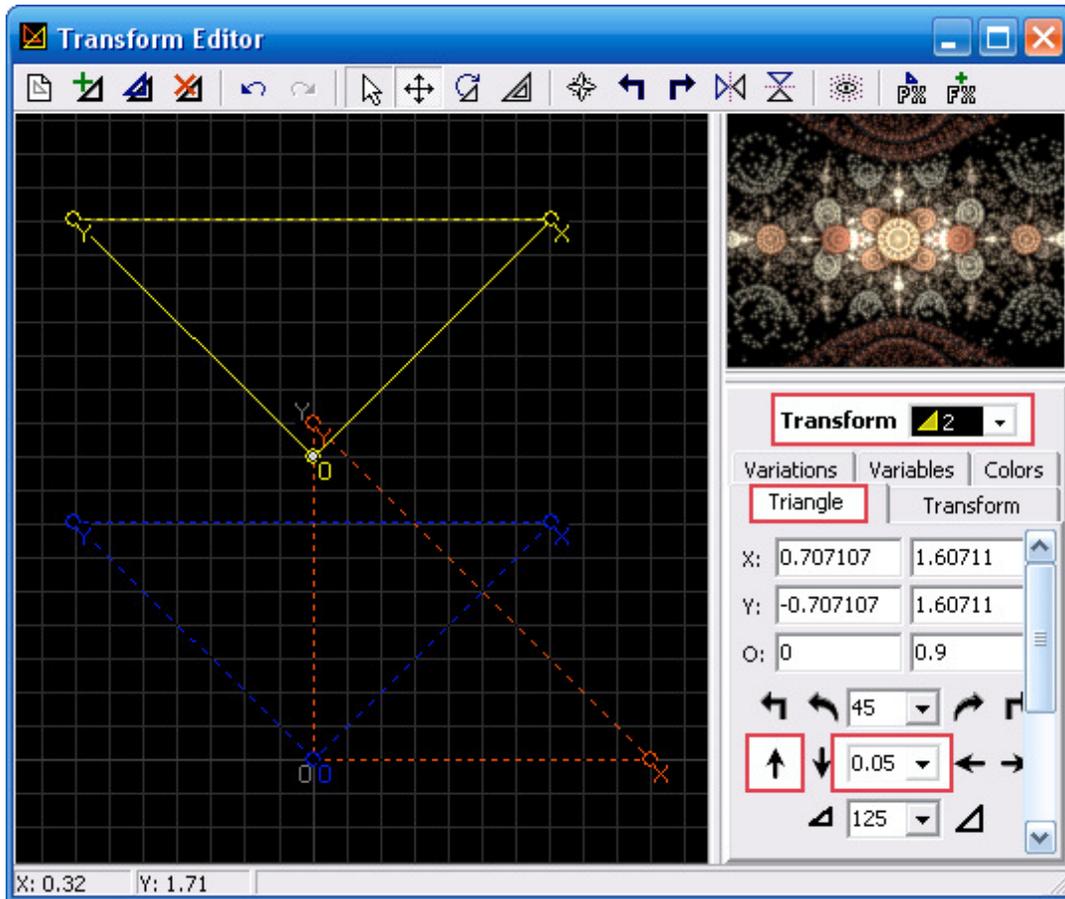


Increase the weight of Transform 2 to 10. For viewing ease, I have altered the gradient values.



Step 2:

Decrease the *Move transform* function to 0.05. Move up until the desired Julian is found. To fine-tune the Julian, decrease the *Move transform* value to 0.01.



Coordinate Examples:



Transform 2 at (0, 0.95)



Transform 2 at (0, 1)



Transform 2 at (0, 1.05)



Transform 2 at (0, 1.1)



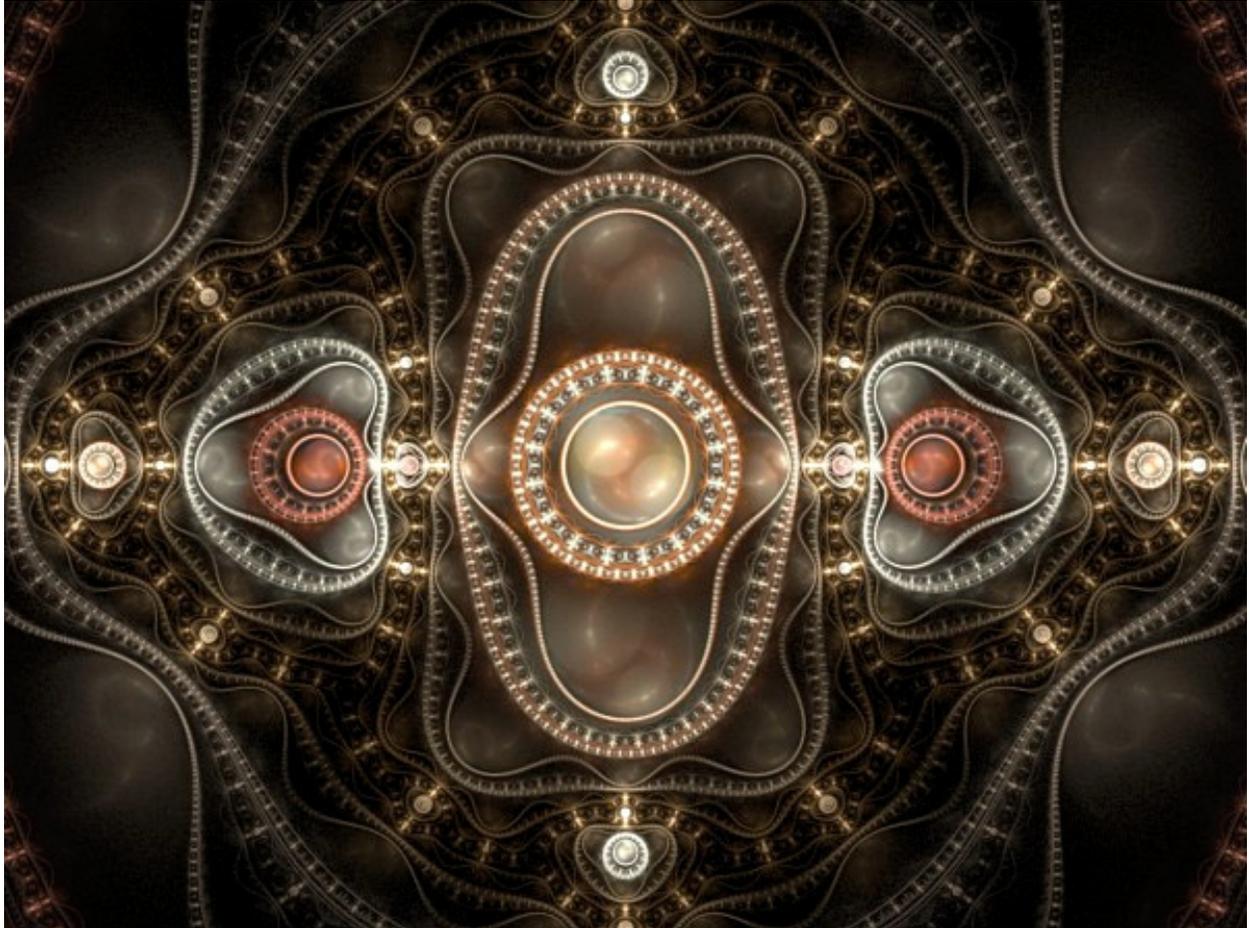
Transform 2 at (0, 1.15)



Transform 2 at (0, 1.2), Weight increased to 20

Step 3:

For the final render, I moved Transform 2 to (0, 1.05). I reduced the Blur of Transform 1 to 0.1 and add Swirl 0.1 and Disc 0.1. Finally, I increased the julian_power of Transform 3 to 50. After another gradient shift, I end up with this:



This style of Julian benefits from a reduced number of transforms. In this example, I have used five transforms, but generally I use only four. Experiment to see what works best for you.

Additionally, at higher coordinate values it may be necessary to substantially increase the weight of Transform 2. I have a few Julians with weights above 50. As with all Julian styles, weights are the key to a success.

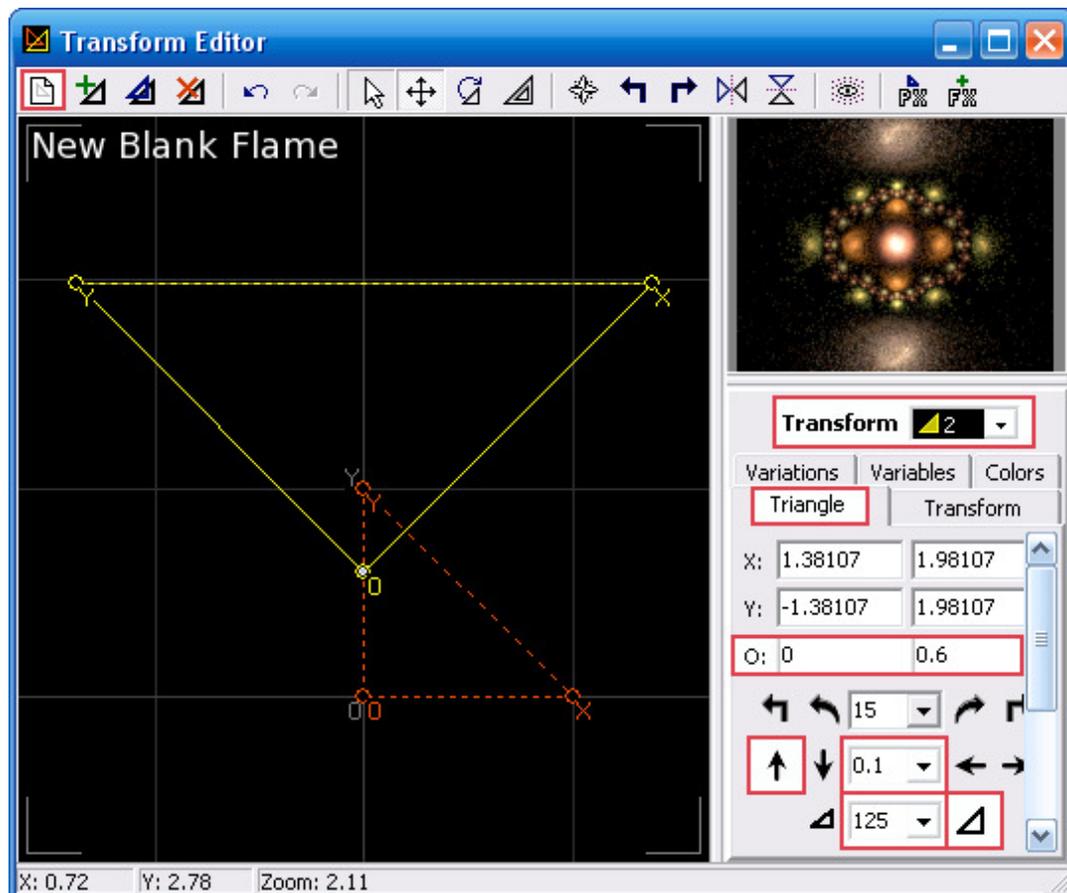
Julian Technique Three

This method involves creating a very large Transform 2 and later duplicating it to create mirrors inside the main Julian formation.

Step 1:

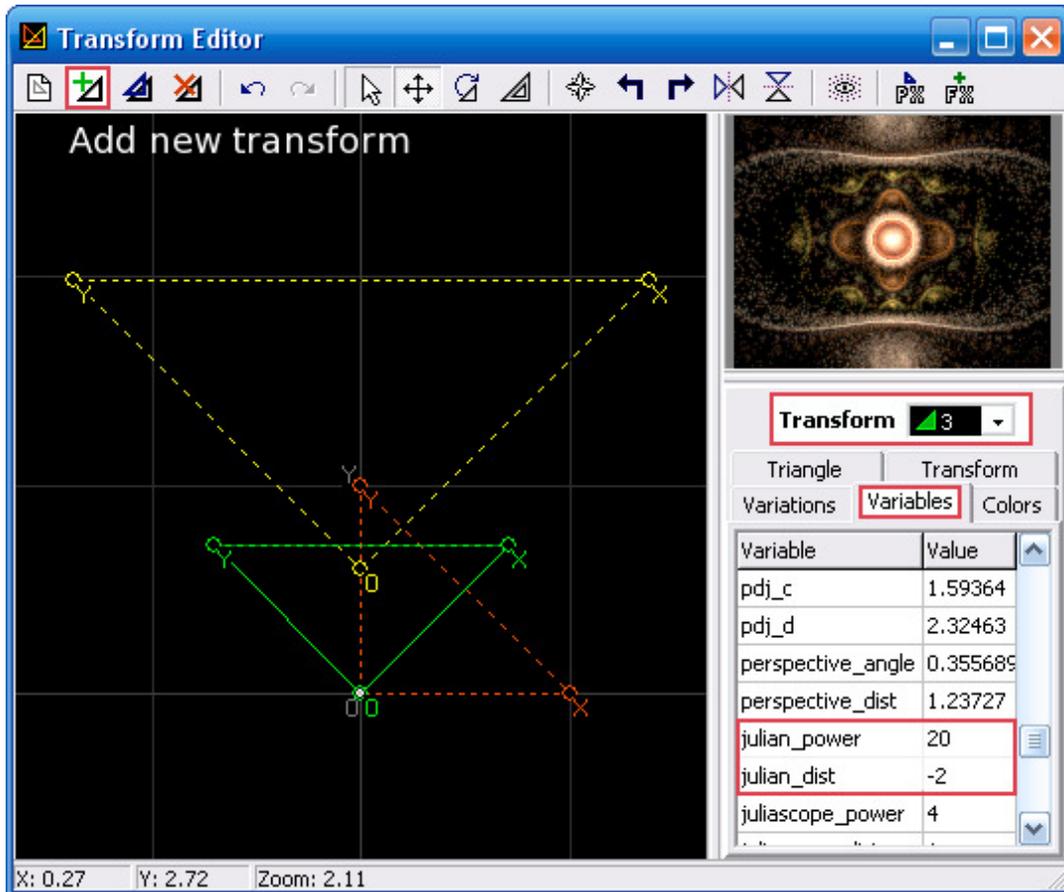
Clear an existing flame to have two new Linear flames. Set Transform 1 to variations Linear 0, Swirl 0.1, Disc 0.1 and Gaussian Blur 0.2.

Set Transform 2 to Linear 0 and Julian 2. Move upward by six 0.1 units so that the 0 apex rests at the (0, 0.6) coordinate. Rotate counterclockwise 45°. Check that the *Scale* function is set to 125 and increase Transform 2 three times (e.g. hit the “scale larger” button three times). Remember to set the *Rotation* under the *Camera* tab in the *Adjust* window to +45.



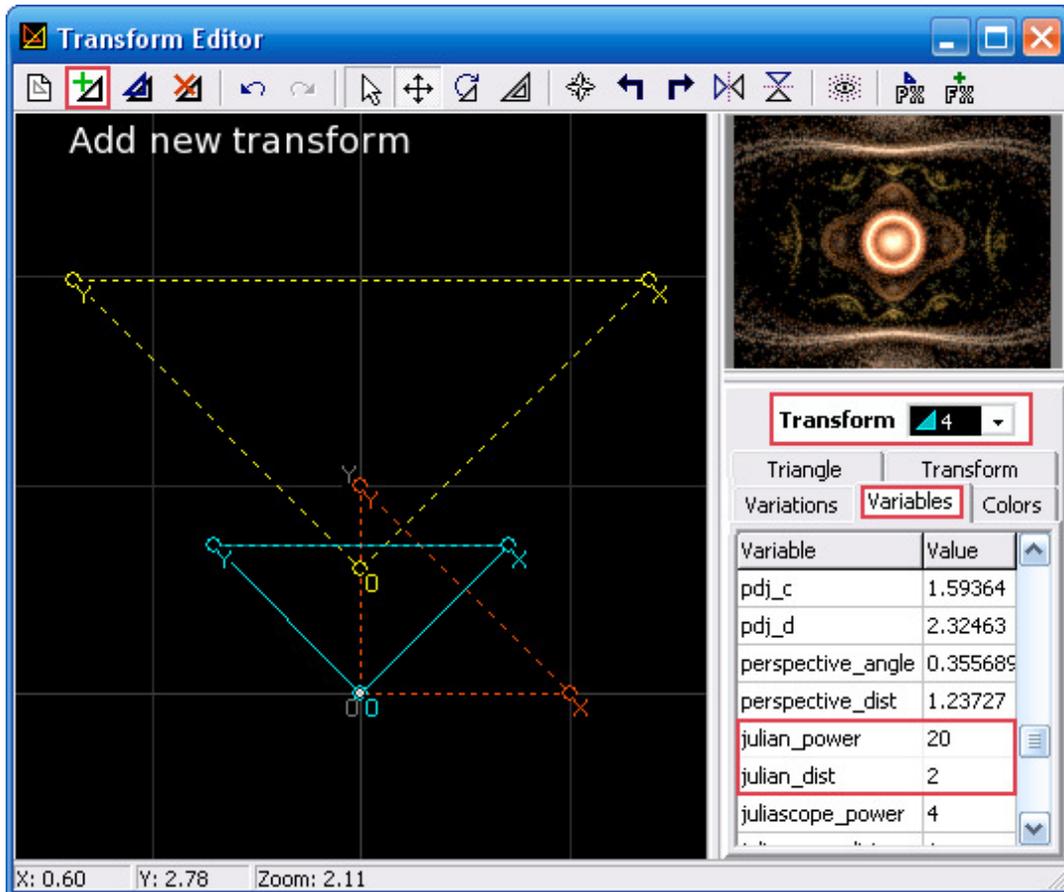
Step 2:

Add a new transform and set the variation to Linear 0 and Julian 0.4, variables at julian_power 20 and julian_dist -2. Rotate Transform 3 counterclockwise 45°.



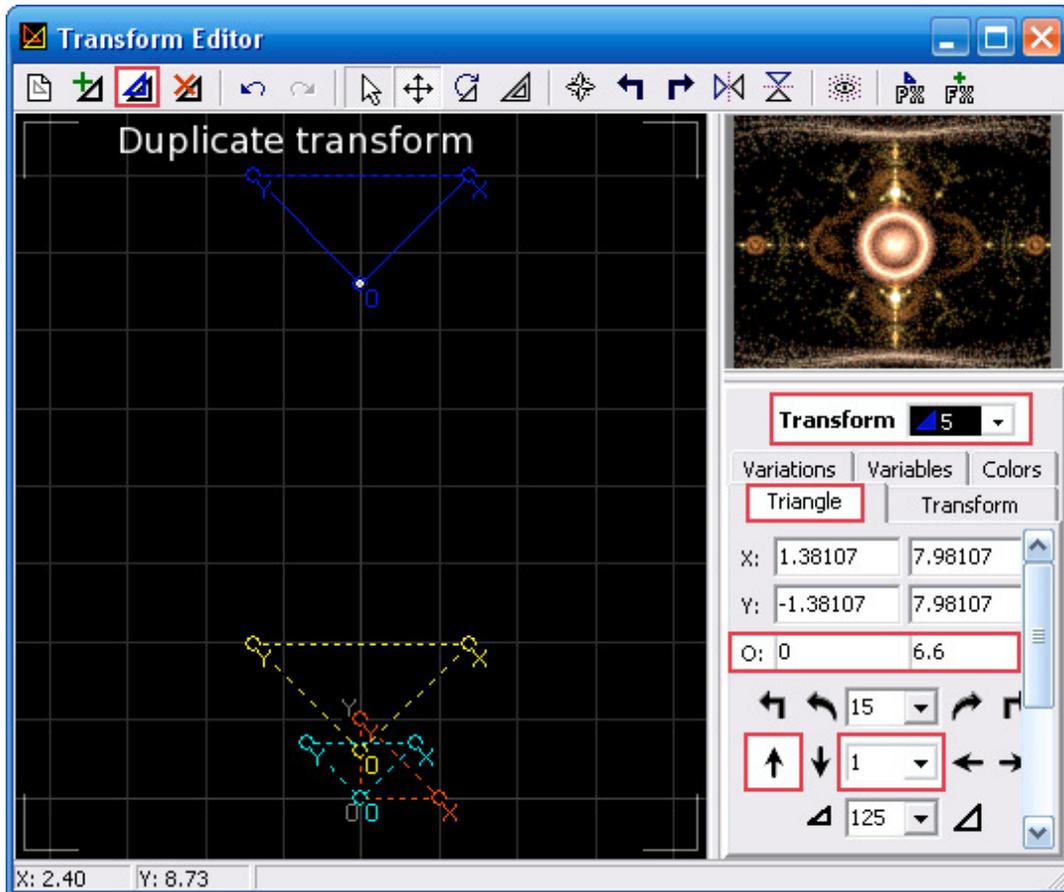
Step 3:

Add a new transform and set the variation to Linear 0 and Julian 0.5, variables at julian_power 20 and julian_dist 2. Rotate Transform 4 counterclockwise 45°.



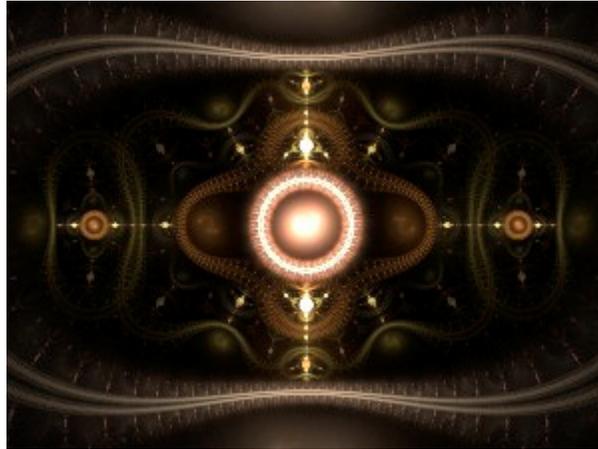
Step 4:

Duplicate Transform 2. Increase the *Move* value to 1 and move up about six units until the 0 apex hits the (0, 6.6) coordinate.



Step 5:

The Julian should look roughly like the image below. Just like this example, your flame will probably be dark, difficult to see and a bit fuzzy.



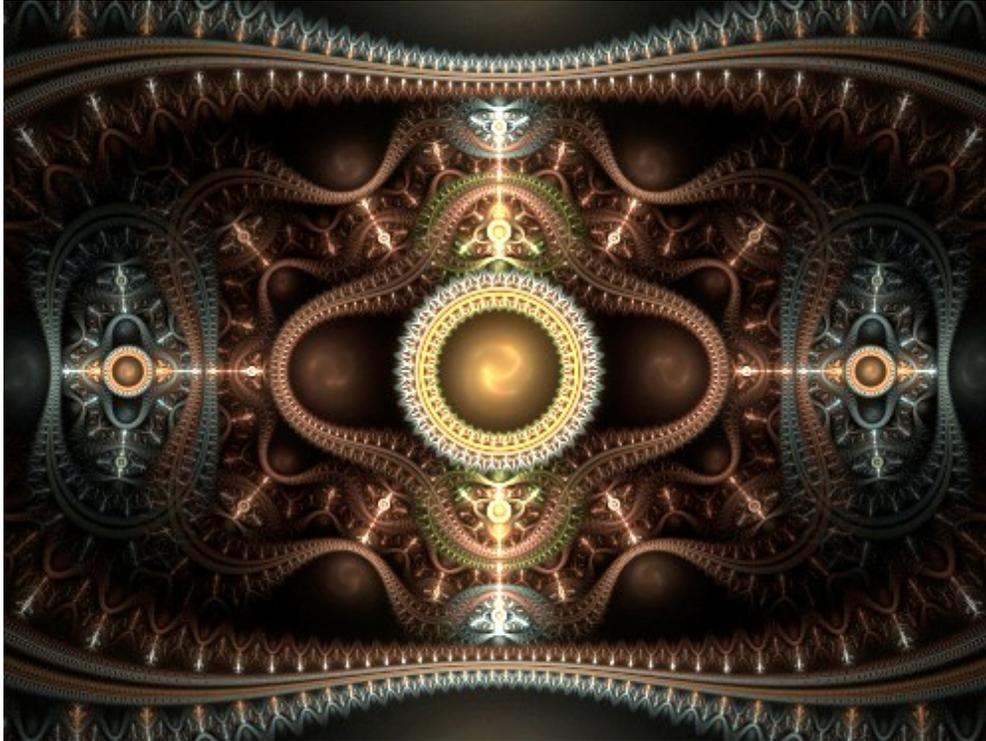
As per the usual, the weights need alteration. Change the transform weights to those noted below.

	<i>Transform 1:</i>	<i>Transform 2:</i>	<i>Transform 3:</i>	<i>Transform 4:</i>	<i>Transform 5:</i>
<i>Weight:</i>	1.5	15	6	4	8



Step 5 cont'd:

Using the *Randomize Color Values* under the *Flame* menu in the main Apophysis window, I alter #667_Tryst to better suit the Julian.



Definitely spend more time tweaking this flame. Given some work, the Julian could easily look much better.

Center Options

If a Julian were comprised solely of the Julia (n) variation, the resulting flame would be much more boring. By simply adding a transform with the Blur variation, the look of flame changes. However, while Julians traditionally use only Blur, additional variations may be added to dramatically change the overall appearance of a flame. Blur need not even be a variation used.



Julian without a Blur transform



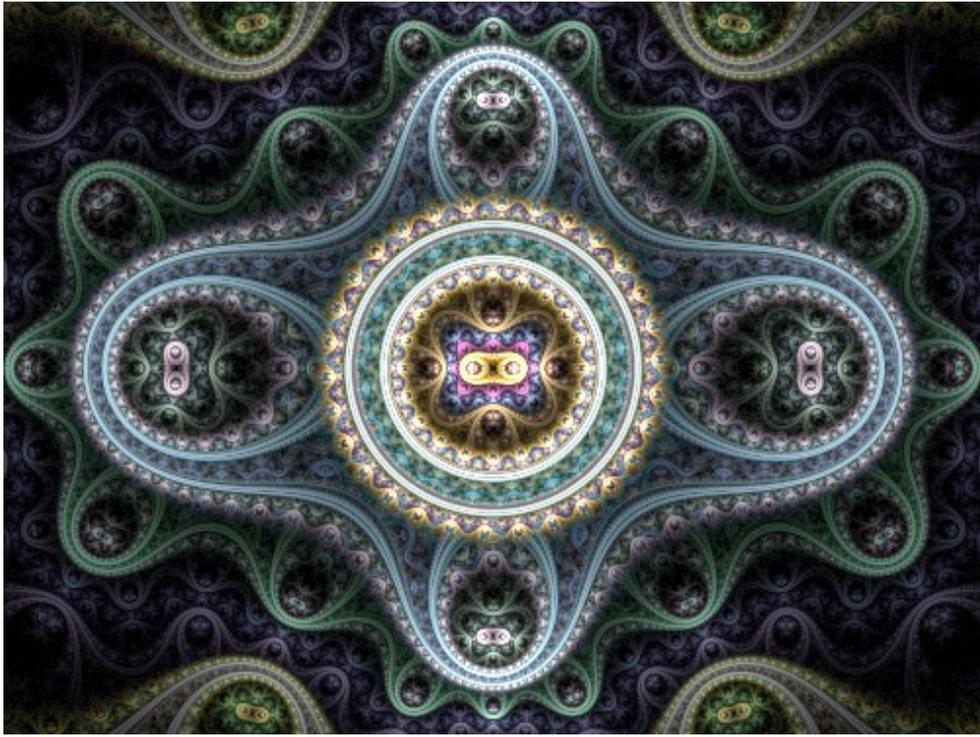
Julian with a Swirl/Disc/Gaussian Blur combo

Over the next several pages are a collection of examples showing some of the different styles available and their effect on the Julian as a whole. All of the following examples show the main variation alone and coupled with Gaussian Blur. Keep in mind that it is also possible to combine multiple variations with one of the Blur options as shown in the example above.

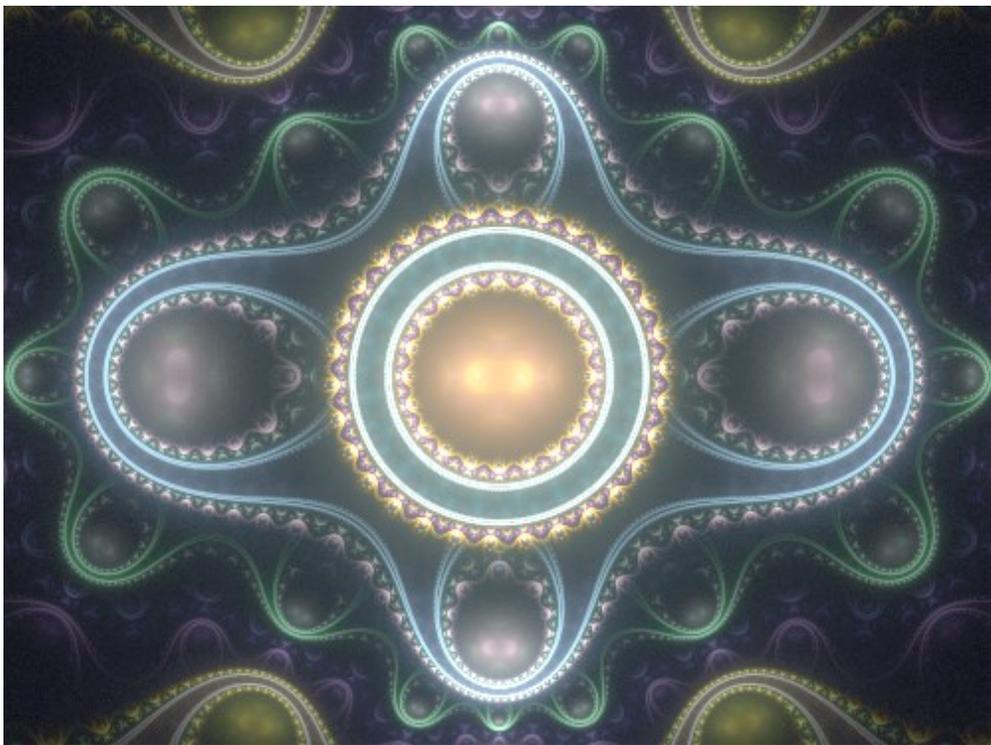
Be warned that adding a Blur variation to Transform 1 can result in an excessive halo depending on the construction of the Julian. If the halo is particularly obvious, change the smallest Julian transform to a positive value (Transform 3, if following the *Julia Uncovered* steps).

Also, not all variations produce good results or even work well with other variations. Be sure to try different variations to find a combination that you like best. In the past, I relied on the Heart and Blur variations for Transform 1 for traditional Julians, whereas now I rarely use Heart for newer work.

The *Colors* tab under the *Transform Editor* is an incredibly powerful tool for maximizing color values in Transform 1. I prefer using a value between 0.2 and 0.5 for *Symmetry*, then altering the *Transform Color* settings. Be aware that changing the Transform 1 values in this method affects the other transforms and additional alterations may be required.



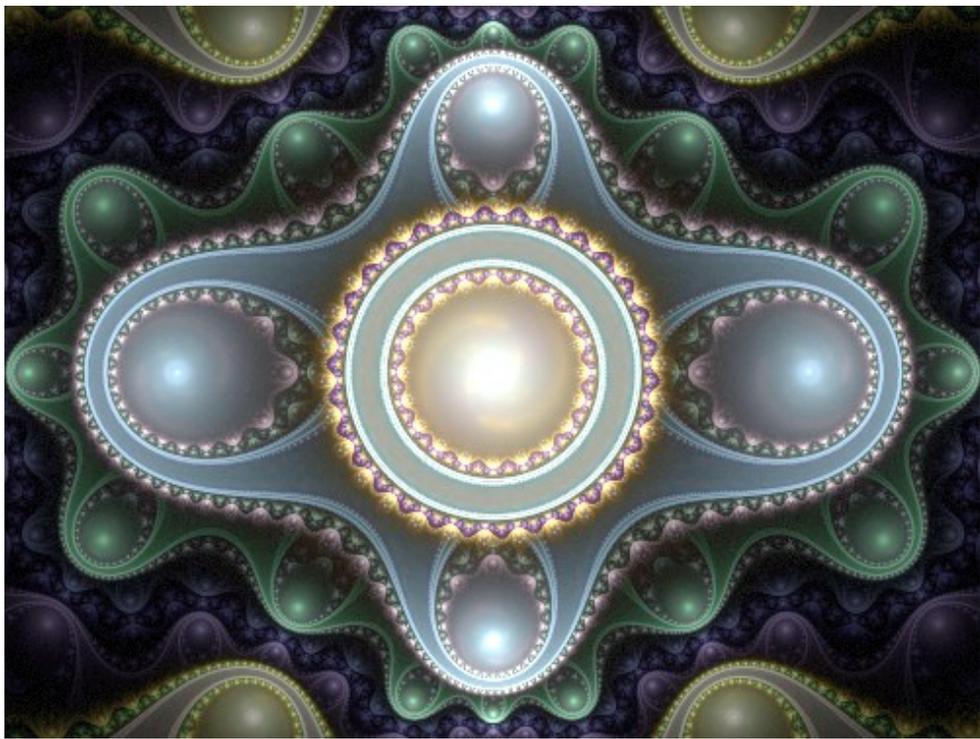
Spherical 0.1



Spherical 0.1, Gaussian Blur 0.1



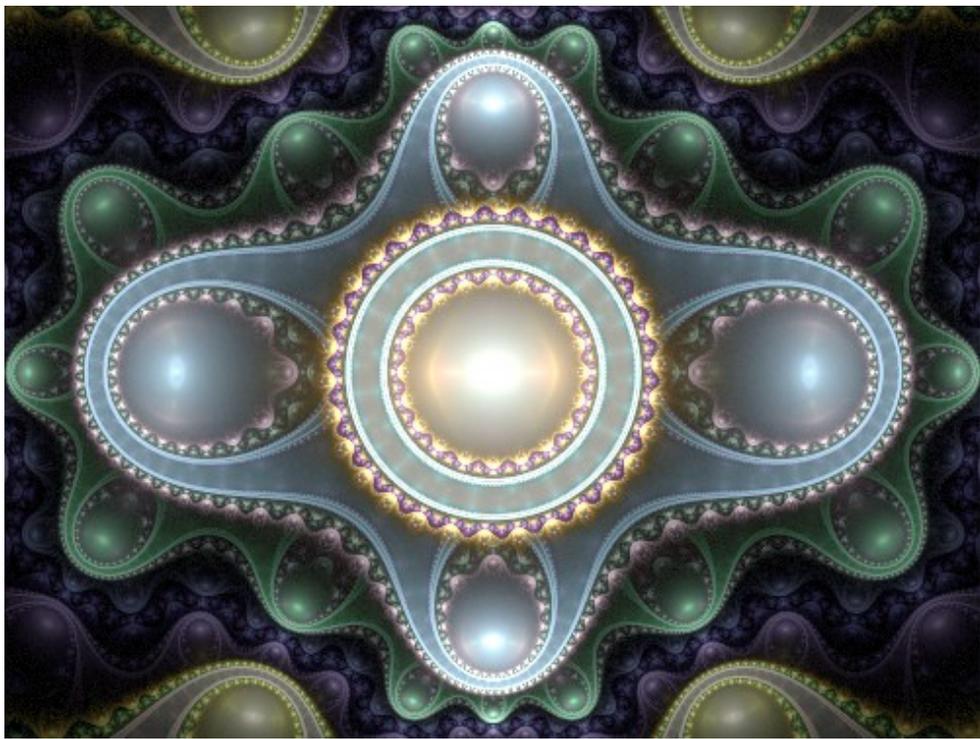
Swirl 0.1



Swirl 0.1, Gaussian Blur 0.2



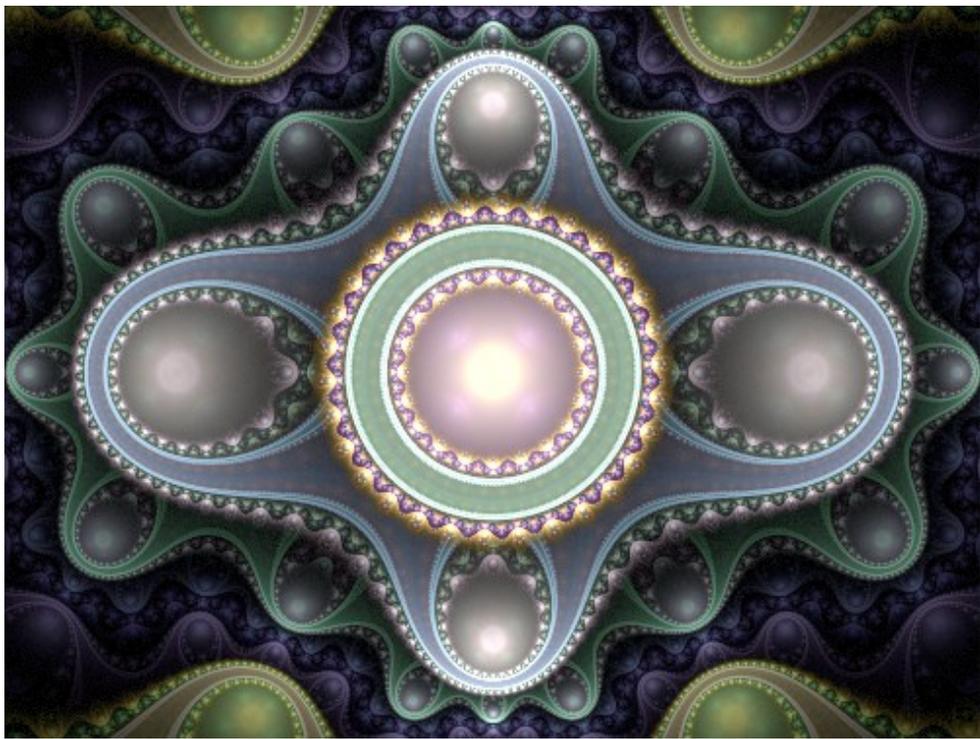
Handkerchief 0.1



Handkerchief 0.1, Gaussian Blur 0.2



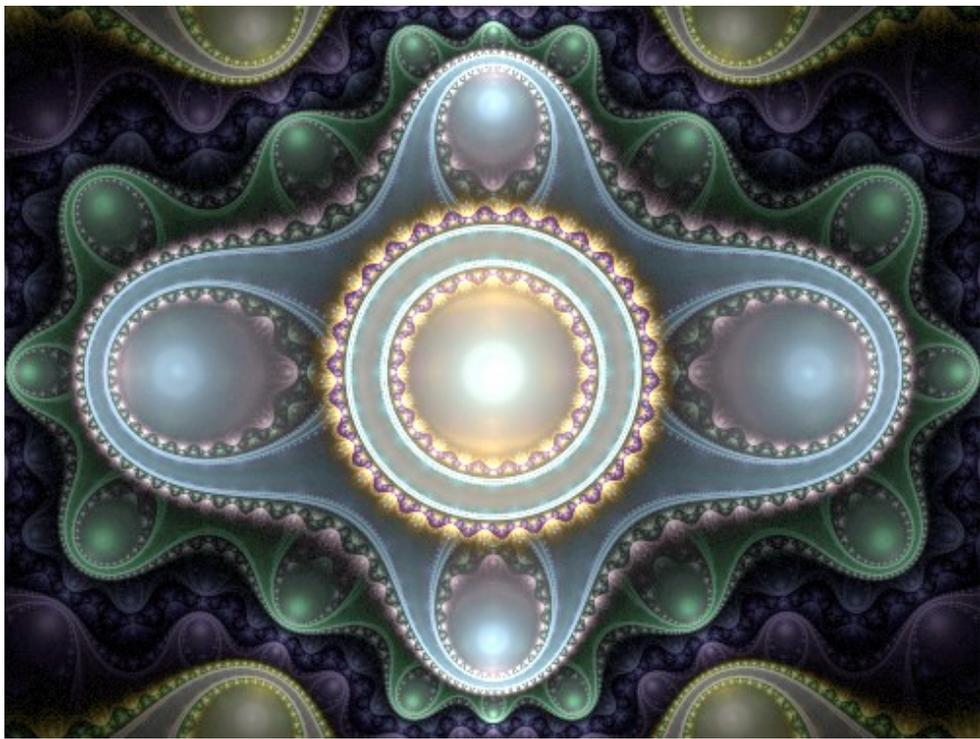
Julia 0.1



Julia 0.1, Gaussian Blur 0.2



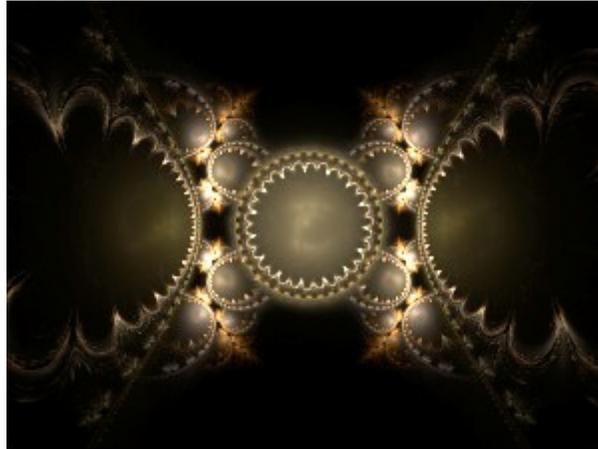
Bubble 0.2



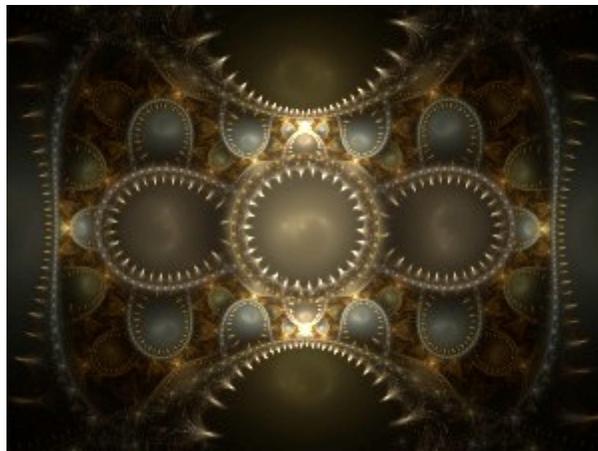
Bubble 0.2, Gaussian Blur 0.2

“Fixing” Julians

I open a Julian created in Apophysis 2.04b and discover that it looks like this:



When it should look like *this*:

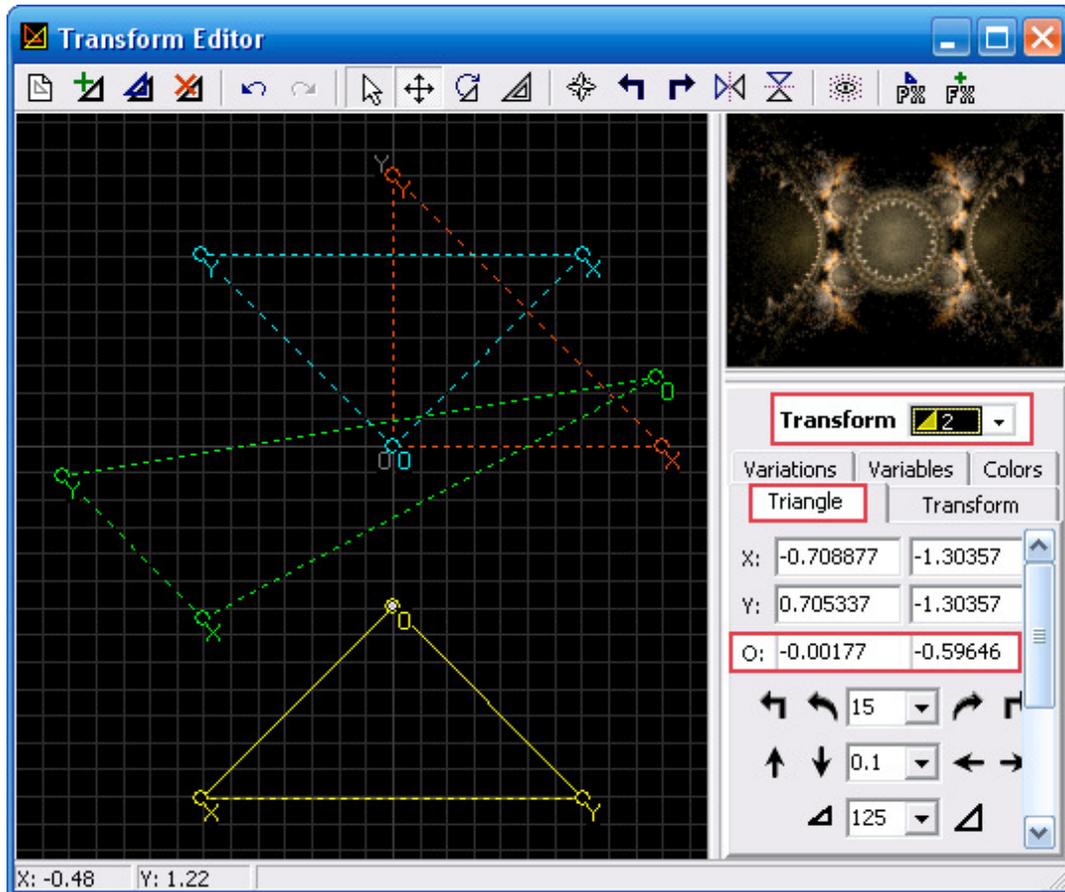


What to do?

The key is to move Transform 2 (or whichever transform sets the main framework, usually set to `julian_power 2` and `julian_dist -1`) to its mirrored position on the graph.

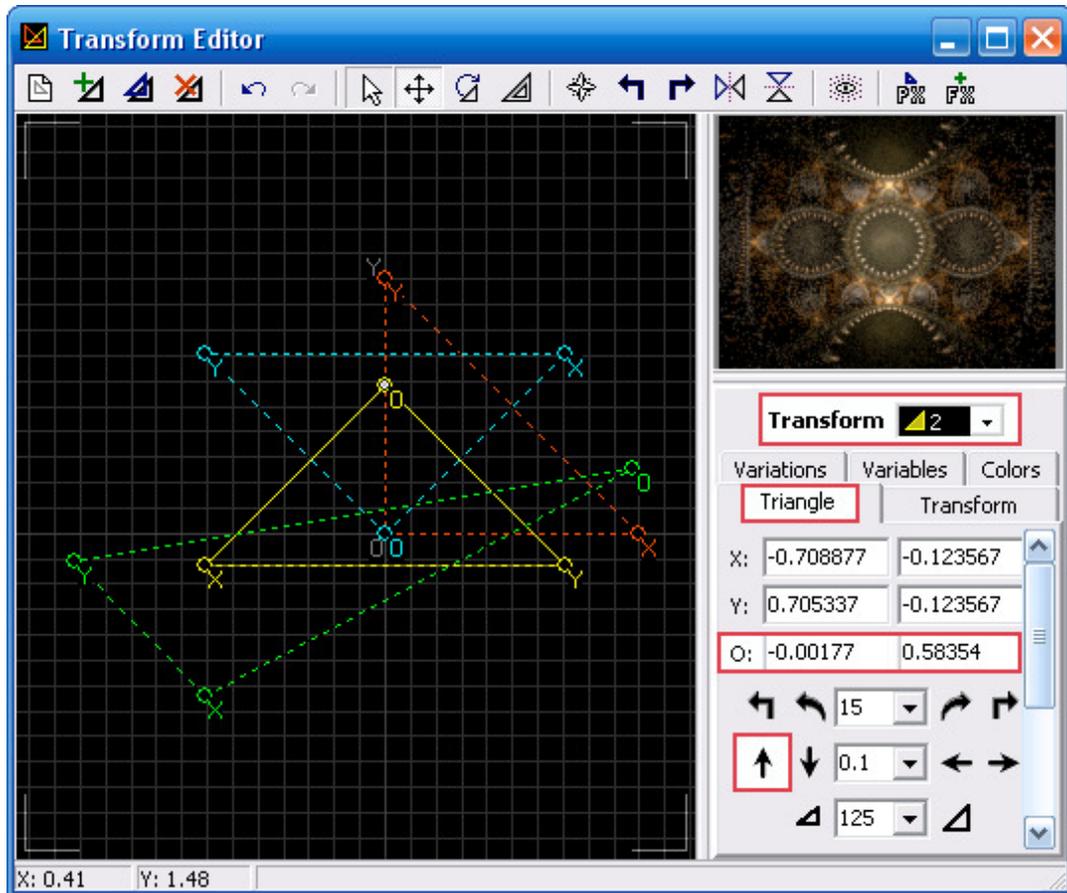
Step 1:

Look under the *Triangle* tab to see where the O apex lies. In this example, it rests at about the coordinate (0, -0.6).



Step 2:

Move Transform to (0, +0.6) and the Julian is now "fixed".



Alternate Method:

The other method to "fix" a 2.04b Julian is to rotate Transform 2 clockwise or counterclockwise by 90°. The flame will need an additional overall rotation, which can be fixed with the *Rotation* setting in the *Camera* tab under the *Adjust* window.

Glossary

Adjust Window:

Currently contains four tabs (*Camera*, *Rendering*, *Gradient* and *Image Size*) to control the overall view of the flame.

Apophysis:

A Windows open-source application for creating IFS “flame”-styled rendered fractals. Created and developed by Mark Townsend until the 2.02 version when the project was continued by a new group of developers. Currently there are three different versions available.

Camera Tab:

Determines the basic placement of the flame by way of *Zoom*, *X* and *Y* position, and *Rotation*.

Colors Tab:

Found in the *Transform Editor*. The options in this tab affect the color values of the selected gradient.

Export Flame:

Exports the current flame to an outside rendering application such as flam3.

Filter Radius:

Determines the rendered image’s particle size. Typically set between 0.4 – 0.8. The smaller the value, the more likely the image will contain jagged lines whereas the higher the number, the more likely it will appear blurry.

Final Transform:

Also known as the *FX Transform*. Applies a variation to the entire flame as opposed to a single transform.

Flame:

Refers to a style of fractal as opposed to the traditional Mandelbrot type. Often associated with Sierpinski triangles.

Gamma:

A very important visual tool for rendering. A low gamma value can result in “hot spots” whereas too high a gamma value creates “fuzzy” images. A value around 3 tends to work best, but it depends greatly on the flame.

Gradient Tab:

A very important tool as it is responsible for coloring the flame. The gradient may also be changed by increasing/decreasing the *Saturation*, *Brightness*, *Rotation* and *Contrast*, among other options.

Hot Spots:

The term refers to bright, often white areas, in a render caused by too much color saturation. Colors like teal, red and other bright colors are very difficult to render properly. Hot spots are sometimes unavoidable, but can often be minimized by lowering the *Brightness* or with post-processing in an image manipulation program such as Photoshop or GIMP.

Image Size Tab:

Sets the image size in the *Preview* window.

Julian:

Generic and now standard term for flames almost entirely comprised of the variation Julia (n). Often contains a Transform with a high Blur variation ratio. Initially added to the 2.04b version of Apophysis.

Julian “background”:

Author’s term for the gutter-like ring around the center of a Julian and controlled by Transform 2.

Julian “circles”:

Author’s term for describing the number of Julian points to appear for a selected transform. Amount per transform is determined by the `julian_power` value.

julian_dist:

This variable sets the length of the Julian circle for the selected transform. The length of the circles is multiplied by the user’s input. The default value is set to 1.

julian_power:

This variable determines how many Julian circles will appear for that transform. The amount of circles visible is double the `julian_power` value.

Master Scale:

Found under the *Rendering* tab in the *Adjust* window. Works visibly the same as the *Zoom* function, though it only enlarges the flame, whereas zoom mathematically recalculates the fractal. When rendering, there may be a need to increase quality to compensate for a traditionally zoomed flame.

Options Window:

Controls the technical aspects of Apophysis, including setting *PNG transparency*, *Display*, *Multithreading*, *Variations*, *Random Batch size*, and selecting the *Export renderer*.

Oversample:

The rendering application multiplies the oversample by the desired image size. For example, if rendering a 1280x1024 image with an oversample of 1, the image will be rendered only at 1280x1024. However, if rendering a 1280x1024 image with an oversample of 2, the image will be rendered at 2560x2048 then automatically scaled to 1280x1024. Note that some prefer to render larger then later manually scale the image by hand to the desired size.

Parameters:

Term for flames saved in Apophysis. Certain aspects of a parameter can be altered in a text editor.

Preview Window:

The main window of Apophysis.

Render:

The rendering application mathematically computes the fractal and outputs the resulting file to a predetermined destination. Note that the image does not appear on screen as it renders. Users may use applications such as the Apophysis or flam3 renderer. Also used to describe the finished image.

Rendering Tab:

Controls various visual aspects of the flame to include *Gamma*, *Brightness*, *Vibrancy*, *Background Color* and the *Master Scale*.

Rendering Window:

In Apophysis, the window that controls the final output image. A variety of options, such as *Size*, *Filter Radius* and *Quality* determine the resulting render. System resource controls under the *Memory Usage* section help render images too large for the current system specifications.

Rotation:

In the *Transform Editor*, the *Rotation* function rotates a transform either on a local or world pivot. This is a very useful tool for the Julian technique. Under the *Camera* tab, the rotation value visually rotates the flame, though the transforms remain unmoved.

Transform:

Also known as a “Triangle”. In the *Transform Editor*, transforms derived from the reference transform may be added and altered by various means to change the shape of the flame.

Transform Editor:

The window in which most of the flame creation/alteration takes place.

Transparency:

To combat “hot spots”, flames may be rendered either in transparency with either flam3 or Apophysis. In Apophysis, the transparency option is found under the *Options* window.

Triangle:

More correctly known as a transform. See “Transform”.

Variations:

Any of the current variations that alter the basic transform shape. They can be used alone or in conjunction with others. The variations available vary depending on which version of Apophysis is used.

Variables:

Some variations also contain variables to customize that particular variation. The *julian_power* and *julian_dist* are very important in Julian creation.

Weight:

Found under the *Transform* tab in the *Transform Editor*. Shifts the importance of a transform in relation to the other transforms present in the current flame. Default value is 0.5.

Links

General Apophysis Information:

Apophysis website

:: <http://www.apophysis.org/>

Latest Apophysis Release Center

:: <http://sourceforge.net/projects/apophysis/>

Apophysis 2.05beta2Z+C+r5 by ~cyberxaos

:: <http://cyberxaos.deviantart.com/journal/11602281/>

Tutorials:

Apophysis: An Intermediate User Guide by Carl Skepper (a.k.a. 2B2H)

:: <http://www.deviantart.com/deviation/33185407>

The Apophysis Guide v.2.1: An Introduction to Basic Flame Creation by Claire Jones

:: <http://www.deviantart.com/deviation/34031367>

Julian Tutorial by Nemo Paice (a.k.a. nemopaice)

:: <http://www.deviantart.com/deviation/36434472>

Acknowledgments:

Thanks to Gill (a.k.a. WelshWench) for taking the original tutorial on a test run and giving it a name. Be sure to check her out at:

:: <http://welshwench.deviantart.com>

Additional thanks to everyone who has downloaded and read the original *Julia Uncovered*. If it had not been for all of the support and encouragement I have received, I would not have felt so compelled to rewrite and expand the guide. Thanks again. ☺

Author Information:

If you see an error or need to contact me, I can be reached at:

:: <http://clairejones.deviantart.com>

:: clairejone@gmail.com