Apophysis Back to Basics: Zoom And Scale Demystified

Having two such similar parameters with wildly differing ranges of values is apt to cause confusion. I hope here to present both a context for their mutual existence and a relatively simple numerical relationship with the quality parameter.

Quality: the parameter and the image

The quality of the final image is dependent on the hit density of points produced in the Chaos Game over the actual 'canvas' area. This density is in turn dependent on the parameter called 'Quality' (henceforth referenced 'Q'). The relationship is one of direct proportionality:

image quality $\propto$ Q

Zoom

This parameter allows for true zooming, that is zooming whilst maintaining image quality. This is achieved by a built-in relationship between image quality and zoom factor. And here's where things get slightly more baffling for the mathematically-averse: the zoom parameter works on a logarithmic scale, base 2, that is for a unit increase in numerical zoom, the visual zoom increases by a factor of 2. Of course, we think in terms of the single linear dimension, that is a line measuring 40px measures 80px after a 2x zoom. The area, however, increases as the square of the linear dimension, hence a 2x zoom magnifies a given area to 2^2 = 4x its original size. Given what we already know of the image quality/Q relationship, and that Q remains constant, we see that there must be a Z-related factor within the image quality. To cut straight to the chase, due to the combination of logarithmic scale and the square, we can write:

image quality $\propto 2^{(2 \times Z)}$

We'll return later to this after looking at the much simpler Scale

Scale

Formerly known as Master Scale, this is effectively a quick-and-dirty zoom, taking no account of sample density changes, hence giving a reduction in image quality at any given Q value. It works on a linear scale (sic!), meaning that doubling the Scale value produces a 2x linear zoom.

History and usage

A personal take: the Scale was effectively determined by the random batch generation, as the 'canvas' size containing most of the image points. Once a suitable Q setting was established, changes in magnification were to be effected using Zoom in order to preserve image quality.

From a usability angle, Scale is preferable for use as a zoom since it doesn't affect preview time. Quality can then be adjusted at render time.

For scripting animations, Zoom (as Flame.Zoom := n;) is the parameter of choice in order to preserve frame quality.

The numbers

First, the table, then some examples to clarify how it works:

	Scale		Quality
Zoom	Value	Factor	Multiplier
-3	3.125	8	64
-2	6.25	4	16
-1	12.5	2	4
0	25	1	1
1	50	1/2	1/4
2	100	1/4	1/16
3	200	1/8	1/64

Default values are Zoom = 0, Scale = 25.

Let's say you have a flame at default settings that looks fine at Q=1000. Applying Zoom = -3 would require a compensating multiplication of Scale of 8, that is 25 x 8 = 200. However, as would be very noticeable on the preview, the image quality is reduced by a factor of 64, hence Q=64000 would be required to produce an identical image. If by some miracle it still looked fine at Q=1000, this would mean that it could be rendered at default with Q=1000/64 ≈ 16!

Or let's imagine that you scale to 100 from 25. In order to preserve image quality, you'll need to adjust the Quality setting in the Render dialogue by a factor of x4 (the factor of ¼ in the table is the *reduction* in image quality).

Because render time ∝ image quality, there can never be any time reduction without sacrificing quality.