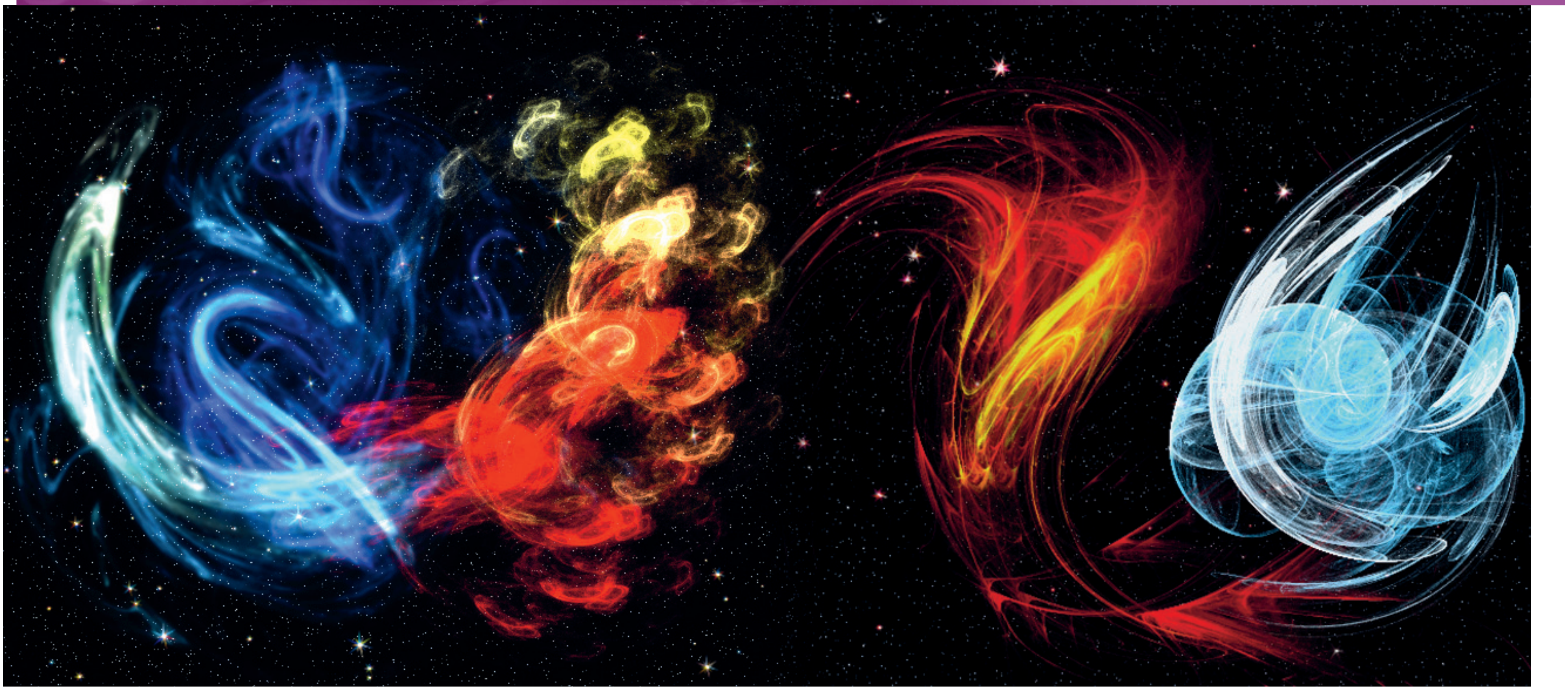


ILLUSTRATION & IMAGE EDITING

Gimp Animated fractals

Michael J Hammel just wants to have fun, so he takes a trip with one of *Gimp's* creative filters.

In last month's tutorial we learned how to make a metallic bordered emblem with a recessed logo. If you missed the issue, call 0870 8374773 or +44 1858 438795.

When I started work on this month's column I realised that I've been talking about using *Gimp* for real-world projects for quite a few issues: logos, posters, retouching... But sometimes you just need to have a little fun at work, and have to be creative just for the sake of being creative. After all, what's the point of being a graphic artist if you can't be artistic?

So, I got to playing with a few of the artistic plugins in *Gimp* and stumbled once again upon the Flame filter. This is a plugin based on open source code from the computer artist and programmer Scott Draves. It uses long strings of numbers – similar to a genetic code, according to Scott – to generate random shapes, or flame fractals. The shapes can be coloured using standard and custom *Gimp* gradients and the results from the Flame filter, and other tools based on the same open source code, can be extremely impressive. Several websites exhibit galleries of these flames *(see Links box on page 81)*, many of which have space or fantasy themes.

The Flames filter

Still, on the surface this filter does nothing more than generate random fractal designs. You can set some initial parameters and tweak the size and position of the image, but in the end you really don't have as much control with this filter as you do with the IFS Fractal filter (Filters > Render > Nature > IFS Fractal). But if you do a little digging you'll find the Flame filter holds a little secret: animation. Scott has expanded on the original source from which Flame originated to produce a few more tools that work with the same type of designs, and the *Gimp* Flame filter combines with these other tools to produce a real creative outlet. As we'll see, the shapes generated with the Flames filter can be used as input to another Draves program called *Flam3*. This program produces a series of images based on an initial version created in the *Gimp*, and this series can then be turned into an animation using *FFmpeg* or *mpeg2encode*. And that's exactly what I'm going to do in this month's tutorial.

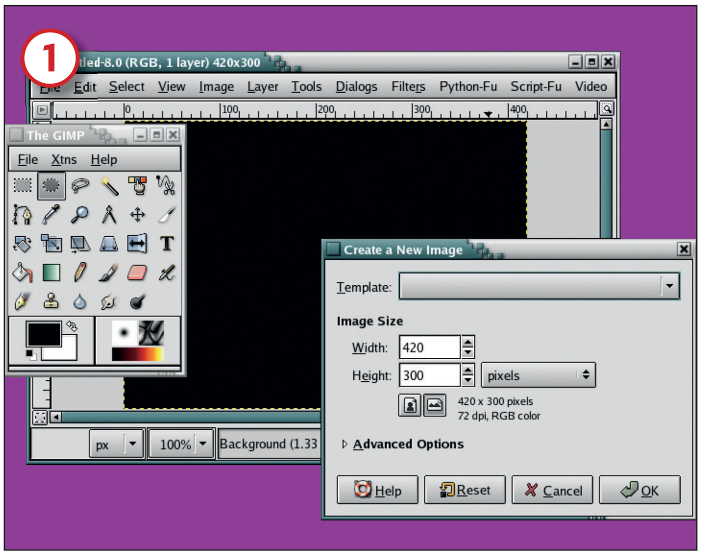
A brief tour of the Flames filter is all that's needed to get started with this project. It's located in the Filter > Render > Nature menu. Opening the filter, you'll find a preview window with the current flame design. A two-tabbed page provides sliders for adjusting the rendering options and camera position, and a menu at the bottom lets us set the colour map for the rendering. Any gradient can be used, but if the Custom gradient setting is used, any changes in the Gradient Editor will take effect when any other rendering option causes the preview to be updated.

Click on the Edit button to edit the current design. A second dialog window opens with an interface very similar to the QBist filter (Filters > Render > Pattern > QBist), where three rows of three previews each are displayed. The central preview is the current design. Clicking on any other button makes that preview the current design and updates all the other previews. Below the previews you'll find a slider, which adjusts the speed at which the previews are rendered. Generally speaking, the faster the render the less complex the design, but the definition of complexity with this filter is really a matter of style.

Below the slider is a randomise button, which will generate a completely new design based on the variation specified in the menu to the button's right. There are nine variations we can use, and switching from one to the other will usually generate a design completely different from the current design. This means you can't make minor modifications to the current design using the Variations menu. It's mostly trial and error to determine how to use the previews to generate something you're happy with. As I said previously, you don't have a huge amount of control with this filter, but that isn't really the point of using it. Find a good design – something appropriately bizarre – and save it.

CREATE A STANDALONE IMAGE

To get a feel for using the Flame filter, I'm going to walk you through a simple space-themed scene. The majority of this work will be in creating the space background. The flame itself is so easy it only takes one step.


1

Create a New Image

Template:

Image Size

Width: 420

Height: 300 pixels

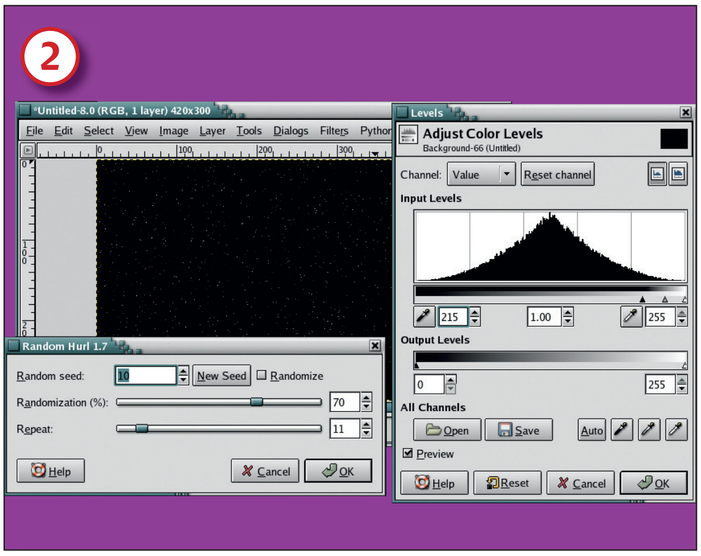
420 x 300 pixels
72 dpi, RGB color

▷ Advanced Options

Untitled-8.0 (RGB, 1 layer) 420x300
File Edit Select View Image Layer Tools Dialogs Filters Python-Fu Script-Fu Video
The GIMP
File Xtns Help
px 100% Background (1.33
Help Reset Cancel OK

Prepare your canvas

Type Ctrl+N in the Toolbox to open a new canvas window. Use the default 400x320 size. If this isn't set in the New Image dialog, just click on the Reset button. Then click on OK to open the canvas. Type 'X' in the Toolbox to reset the foreground and background colours, then drag the foreground (black) into the canvas.


2

Random Hurl 1.7

Random seed: 10 ☐ Randomize

Randomization (%): 70

Repeat: 11

Levels

Adjust Color Levels
Background-66 (Untitled)

Channel: Value

Input Levels: 215 1.00 255

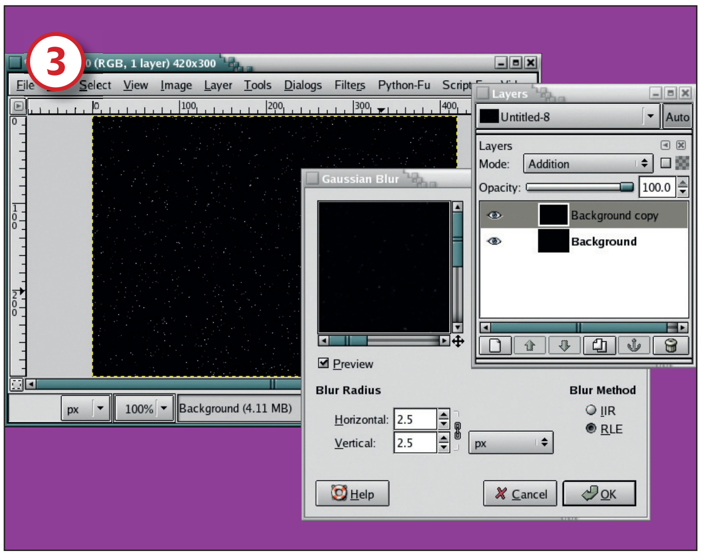
Output Levels: 0 255

☑ Preview

Untitled-8.0 (RGB, 1 layer) 420x300
File Edit Select View Image Layer Tools Dialogs Filters Python
New Seed
Reset channel
All Channels
Open Save Auto
Help Cancel OK
Help Reset Cancel OK

Add some stars

Open the Hurl filter (Filters > Noise > Hurl). Set the Random Seed to 10, the Randomization to 70% and the Repeat to 11. Click the OK button to apply this to the background layer. Desaturate the image (Layer > Colors > Desaturate), then open the Levels dialog (Layer > Colors > Levels) and adjust the Input levels as shown here.


3

Layers

Untitled-8

Mode: Addition

Opacity: 100.0

Background copy

Background

Gaussian Blur

☑ Preview

Blur Radius

Horizontal: 2.5

Vertical: 2.5 px

Blur Method

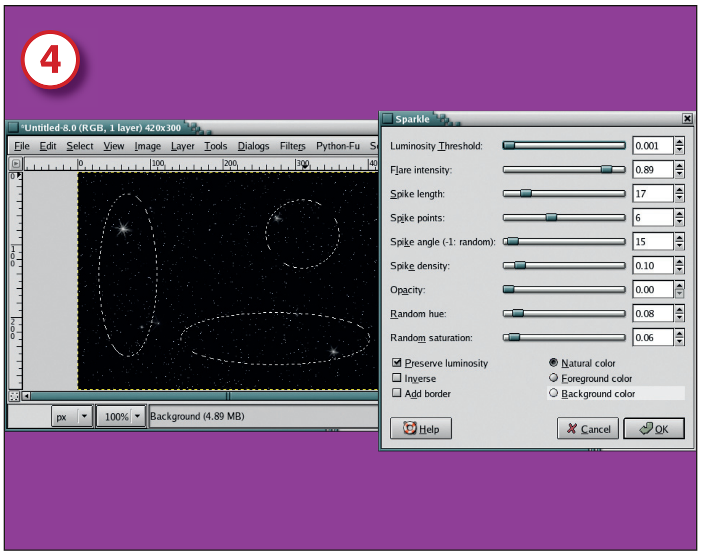
○ IIR

● RLE

0 (RGB, 1 layer) 420x300
File Select View Image Layer Tools Dialogs Filters Python-Fu Script
Auto
px 100% Background (4.11 MB)
Help Cancel OK

Blur the background

Duplicate the background layer (Layer > Duplicate Layer). Click on the original background layer in the Layers dialog to make it the active layer. Open the Gaussian Blur filter (Filters > Blur > Gaussian Blur) and set the Blur Radius to 2.5 for both the Horizontal and Vertical directions, then click on OK to apply the settings to the background layer. Click on the duplicate layer to make it active, and set that layer's blend mode to Addition. Finally, merge the two layers together (Layer > Merge Visible Layers).


4

Sparkle

Luminosity Threshold: 0.001

Flare intensity: 0.89

Spike length: 17

Spike points: 6

Spike angle (-1: random): 15

Spike density: 0.10

Opacity: 0.00

Random hue: 0.08

Random saturation: 0.06

☑ Preserve luminosity

☐ Inverse

☐ Add border

● Natural color

○ Foreground color

○ Background color

Untitled-8.0 (RGB, 1 layer) 420x300
File Edit Select View Image Layer Tools Dialogs Filters Python-Fu S
px 100% Background (4.89 MB)
Help Cancel OK

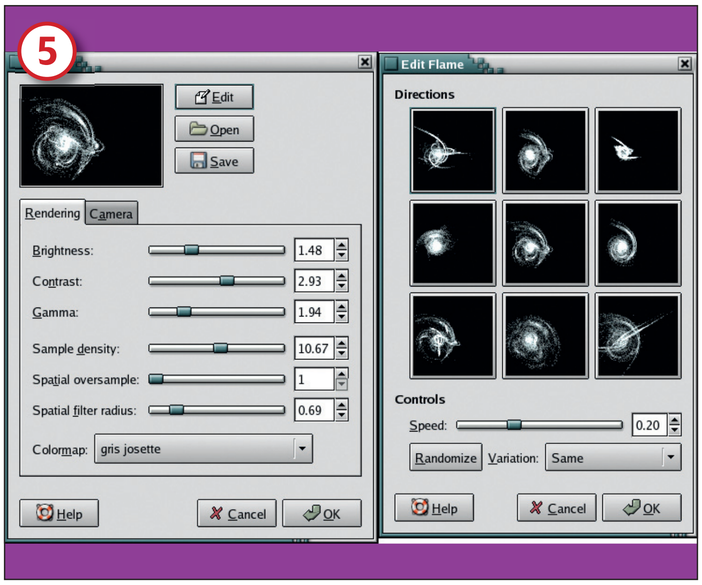
Make some stars sparkle

Make several oval selections (hold down the Shift key to add a new oval to existing selections). Feather the selections by 10–35 pixels (Select > Feather). Open the Sparkle filter (Filters > Light Effects > Sparkle). Set the sliders to something similar to what is shown here. The Spike Density should be set low – that slider determines how many stars get enhanced.

»

«

5

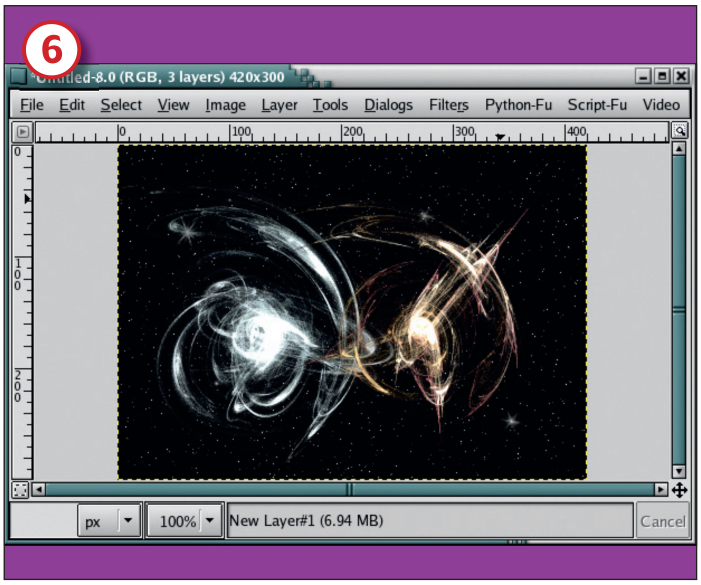


Brightness: 1.48
Contrast: 2.93
Gamma: 1.94
Sample density: 10.67
Spatial oversample: 1
Spatial filter radius: 0.69
Colormap: gris josette
Speed: 0.20
Variation: Same

Design your flame

Now we're ready for the flame. Since Flame will overwrite the contents of the current layer, you should start by adding a new transparent layer (Layer > New Layer). Open the filter. You can use the default design or create one of your own. Again, the nine previews in the Edit dialog allow you to change the shape.

6



Untitled-8.0 (RGB, 3 layers) 420x300

px 100% New Layer#1 (6.94 MB)

Save the result

Once you've chosen a suitable shape and colour map, click on OK to apply the effect to the background layer. If you find you really like the design, use the Save button and save to any directory. The saved designs are what I'll be using in the next part of this project: creating a series of images using *Flam3*.

QUICK TIPS

- **It's easiest to create a flame in the *Gimp* filter at a high resolution to start with, then save that Flame file. This is better than scaling the results with the *Flam3* tools or with *FFmpeg* or *MPlayer*.**
- **You may need to adjust the brightness when rendering at higher resolutions with the *Gimp* Flame filter. You'll definitely need to adjust the zoom.**
- **Also for higher resolutions, try increasing the Sample Density. This slider doesn't affect the preview, but will change the amount detail in the image rendered to the *Gimp* canvas.**
- **Centre any Flame filter file you save on the Camera preview using the Camera options. If you don't do this any animation generated from that file will be shifted off-centre.**
- **The easiest way to affect the playback speed is to keep the default setting under *Flam3* and *FFmpeg* and use *MPlayer* or *Xine* to set the frames per second setting.**

A FLAM3 ANIMATION

Once you've saved at least one design to a file you're ready to generate your flame series. The saved file is first converted to an XML file by *Flam3-convert*, which is then fed to *Flam3* to produce a series of images. You'll need to download and compile the *Flam3* software for this part of the project. The software is available for Linux from the *Flam3* website *(see Links box, below right)*. Unpack the software with

```
tar xvzf Flam3-2.6.tar.gz
```

Change into the **Flam3** directory. The software has no prerequisites so compiling it is extremely simple, thus:

```
cd Flam3-2.6
./configure
make
```

This produces a set of programs: *Flam3-render*, *Flam3-convert*, *Flam3-animate* and *Flam3-genome*. *Flam3-render* can be used to generate a single image but we've already done that with *Gimp*'s Flame filter so there's no need to concern ourselves with that program. You can optionally install the software too, with **make install**. But this requires root access unless you're familiar with setting the installation directory using *Autoconf*.

You can just as easily run the programs from the same directory in which you compiled them. A note of warning before you dive in to these programs: they aren't documented for the average user. There are numerous command line options with some being undocumented and their use and meaning is not always clear. But keeping in mind you're trying to be creative, don't be afraid to just experiment. At worst, you'll end up with several JPEG images you don't like. Just try, try again.

With the programs compiled, the next step is to convert the saved Flame filter files to *Flam3* XML files. This is easily accomplished with the following command:

```
cat ~/.gimp-2.2/flame/fractured-glass |./Flam3-convert >
glass.xml
```

The Flame filter saves its configurations in an ordinary text file. In this example the file was saved to a directory called **flame** in the default *GIMP 2.2* user's directory. The converter program, *Flam3-convert*, reads the filter file via standard input and outputs an XML file called **glass.xml**. The converted XML file will be used to generate your image series. Unlike many Linux command line programs, *Flam3* programs get their options from environment variables. The most obvious way to set these is on a single command line, such as

```
template=vidres.Flam3 repeat=10 Flam3-genome > new.
Flam3
```

The environment variables **template** and **repeat** are options for the *Flam3-genome* program. For a full list of options, use the **-?** command line option with any of the programs mentioned.

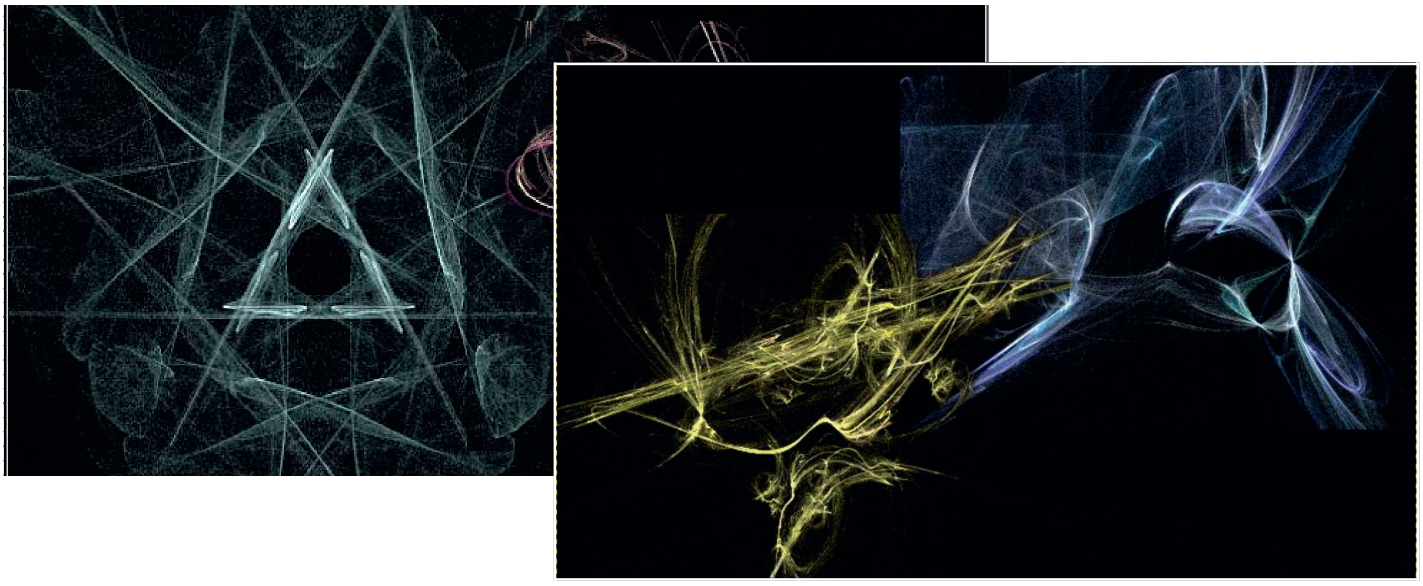
Creating a series

Now that you have the basic background for using the *Flam3* tools, the image series can be created. The first step here is to process the original configuration into a set of control points. This is done using the *Flam3-genome* program. The genome program mutates the design in a manner, according to Draves, similar to genetic algorithms. The command format will be similar to this:

```
verbose=1 aspect_ratio=1.6 gs=3.5 ss=2.5 template=article.
xml repeat=20 ../Flam3-genome > control.Flam3
```

The **verbose** option is useful to watch that something is actually happening when the command is run, otherwise you won't see anything. The aspect ratio is set for my display, which is 1280x800. Leaving this option out won't change the resolution you set in the original *Gimp* filter (420x300). The **gs** and **ss** options increase the quality and size of the images that will be rendered. The amount chosen is arbitrary for this example – I was just shooting for enlarging the images and improving the quality of the rendering to compensate.

The only two options you can't live without here are the **template** and **repeat** options. The **template** option specifies



Move over Jean-Michel Jarre: here's a collection of stills taken from a short movie made by rendering an image from one of 20 control points, rotating it and morphing it for a number of frames until it becomes the next control point. If you watch the movie closely you'll see each of these four control points show up, though only briefly as they get morphed into the next control point. You need to use your imagination if all you're doing is looking at still images – go to www.graphics-muse.org/source/lxfi73.avi to see it as it should be.

the XML input file for the *Flam3-genome* program. This is the XML file created earlier by *Flam3-convert*. The **repeat** option specifies how many control points to generate. Each control point is a variation on the original design.

The control points define a set of designs in the animation, each based on the original design from *Gimp*'s Flame filter. Rendering one image for each design and putting that into a movie would produce a very short movie that moved wildly between images. A better solution would be to produce a set of variations on each design that move smoothly into the next design. This is done by filtering the control points file through *Flam3-genome* again to generate a sequence of rotations and transitions. The command format is very similar to the last one:

```
verbose=1 sequence=control.Flam3 nframes=120 ../Flam3-genome > seq.Flam3
```

The new options are **sequence**, which is the input file to use, and **nframes**, which is the number of frames to generate for each control point. Each frame is a small rotation of the design along with a slow mutation into the next design. The output from this command will be the input into the animation program, *Flam3-animate*:

```
verbose=1 ../Flam3-animate < seq.Flam3
```

When this command is run, prepare yourself: it will generate a huge number of images. Each image will be a frame in the movie, but at 30–60 frames per second that isn't a very long movie. Changing the **nframes** option in the second *Flam3-genome* command will reduce the number of images, as will changing the number of control points you generated. Again, playing with the tools is the best way to learn them.

Generating a movie

Once the huge set of frames is generated you're ready to produce your fractals movie. This is almost a no-brainer with the use of *FFmpeg*, the Swiss Army knife of video conversion. The simplest command to get from a directory of thousands of JPEG images to an AVI-formatted movie suitable for playback by *MPlayer* or *Xine* is

```
ffmpeg -i %04d.jpg movie.avi
```

The **04** in this code means that each JPEG image file is prefixed with a four-digit number. If we had used the **prefix** option to animate the series we could have added some other prefix to the filenames, but for this example a four-digit number is sufficient. Generating the movie from a thousand 420x300 JPEG images doesn't take long – a few minutes at most on my laptop (Intel Pentium M 1.5G and 512MB memory).

Playback is simple enough as well:

```
mplayer -fs -vo x11
```

The **-fs** option puts the playback in full screen mode and the **-vo** option just provides a fairly standard video interface that doesn't require hardware acceleration (which this example doesn't really need).

> **"KEEPING IN MIND THAT YOU'RE TRYING TO BE CREATIVE, DON'T BE AFRAID TO EXPERIMENT."**

You can see the movie of my series at my own website *(see Links box, below)*, and I hope you go on to produce some colourful, trippy movies of your own. And what good is all this work with fractals? Well, not much, really. Some of Draves's work has moved into the video DJ world, where walls of images just like these throb to the sounds of modern music marvels. But it's something that only a few artists take to a commercial level – for most of us it's just a creative outlet, a way to have fun. And before too long, we're back to replacing teeth and generating logos. The work must go on, after all.

LINKS

- **Fractal galleries: www.arcanefractals.com/index.shtml**
  www.renwebdesign.com/fractals/index2.html
  http://occipital.net/main.php
- ***Flam3* website: http://Flam3.com**
  ***Flam3* Linux source code: http://Flam3.com/Flam3-2.6.tar.gz**
  Scott Draves's personal website: http://draves.org
  Sample movie generated for this article: www.graphics-muse.org/source/lxfi73.avi

NEXT MONTH

I'll take a look at the best image viewers and content managers for cataloguing your *Gimp* projects..