The Physical, the Digital and the Designer

LIBRE
GRAPHICS
1.4

OERT

Open Educational Resources for Typography

This project aims to create a high quality public learning resource on typography and type design, available to every student in the world. The result will be completely free and open, like Wikipedia.

We are teachers of typography at FADU/UBA, the Faculty of Architecture, Design and Urbanism from University of Buenos Aires. This is a free public university, one of the most important in Argentina.

We teach over 500 students each year, so we have been making typography learning resources since 1994. Updating and translating them into English, we will bring this knowledge to the web at **www.OERT.org**

Index

Masthead

Editorial Team

Ana Carvalho ana@manufacturaindependente.org
ginger coons ginger@adaptstudio.ca
Ricardo Lafuente ricardo@manufacturaindependente.org

Copy editor

Margaret Burnett

Publisher

ginger coons

Community Board

Dave Crossland
Louis Desjardins
Aymeric Mansoux
Alexandre Prokoudine
Femke Snelting

Contributors

Dave Crossland, Maria Figueiredo, Nelson Gonçalves, Eric de Haas, Richard Hughes, Jonathan Puckey, Eric Schrijver.

Printed in Porto by Cromotema (http://cromotema.pt) and in Toronto by Green Printer (http://www.greenprinteronline.com) in recycled paper.

Contacts

Write to us at enquiries@libregraphicsmag.com
HTTP://LIBREGRAPHICSMAG.COM

A Reader's Guide to Libre Graphics Magazine

In this magazine, you may find concepts, words, ideas and things which are new to you. Good. That means your horizons are expanding. The problem with that, of course, is that sometimes, things with steep learning curves are less fun than those without.

That's why we're trying to flatten the learning curve. If, while reading *Libre Graphics* magazine, you encounter an unfamiliar word, project name, whatever it may be, chances are good there's an explanation.

At the back of this magazine, you'll find a glossary and resource list. The glossary aims to define words that are unique to the world of Libre Graphics. The resource list provides valuable information about tools, licenses, whatever items we may be mentioning.

Practically, this means that if, for example, you're reading an article about Paper.js (see pages 30 to 33), you can always flip to the back of the magazine, look up Paper.js in the resource list and become quickly informed about it. This provides some instant gratification, giving you the resources you need to understand, in a moment, just what we're talking about.

We hope you like our system.

Images under a CC Attribution Share-Alike license
Photo of ginger coons by herself.
Photo of Ricardo Lafuente by Ana Carvalho.
Photo of Ana Carvalho by Luís Camanho.
Photo of Dave Crossland by Mary Crossland.
Photo of Eric Schrijver by himself.
Illustrations in "Type Design" based on "1032056", image by Flickr user fdctsevilla.
Illustrations in "Dispatch", page 20, by Natacha Cindy and Hugo Ferreira.
Photos in "Notebook", in order of appearance, by Pedro Caetano, Bruno Fernandes, ginger coons and Aloysio Araripe. All photos, with the exception of ginger coons' photo, can be found at Flickr, in the photostream of the user festivalculturadigitalbr.
All images in the "Showcase" section can be attributed to the creators mentioned therein. All are licensed CC BY-SA with the exception of "I heart lightning" in the article "The graphic side of 3D printing" which is licensed under CC BY.
Illustrations in "Folds, impositions and gores: an interview with Tom Lechner", by Tom Lechner.
Photos in "Colorhug: filling the F/LOSS colour calibration void" by Richard Hughes.
Map images in "Best of SVG" are from OpenStreetMap contributors.

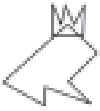
Images under other licenses
Photos in "Dispatch", page 21 by Fátima Barreiros are under CC-NC-SA. They were published here with the permission of their author.
Compass rose image used in "Best of SVG", by Wikimedia user Themadchopper is in the public domain and can be found at:
http://commons.wikimedia.org/wiki/File:Modern_Nautical_Compass_Rose.png

General
Advertisements, with the exception of those representing Libre Graphics Magazine, are not necessarily covered by the blanket CC BY-SA license. It is best to check with the projects they represent before reusing them.

OSP

Foundry 2012

spring

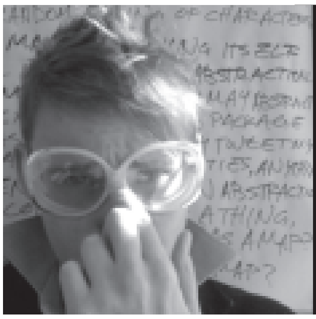


FUTURE

RELEASES

mixtape

ospublish.constantvzw.org/foundry



Editor's letter

ginger coons

We so often draw a strong distinction between the physical and the digital, acting as if the one is solid, staid and reliable, while the other is born in ether, unreal and untouchable. This is far from the case. Not only on occasion, but always, the physical and the digital are intimately intertwined.

The digital is merely a subset of the physical. It is a land we've come to view as different and distinct, despite its reliance on the physical. Regardless of our perceptions, the two tick along, happily co-operating and relying on one another. As the digital fails to escape the bounds of the physical, the physical comes to embrace its part in the success of the digital.

Graphic design and media arts are fields well acquainted with the obvious areas of overlap between the physical and the digital. From the days of air brushing and drafting by hand, to the bringing of those same metaphors into the realm of digital production, designers and media artists are at the forefront of both the conflicts and the embraces of the digital and the physical.

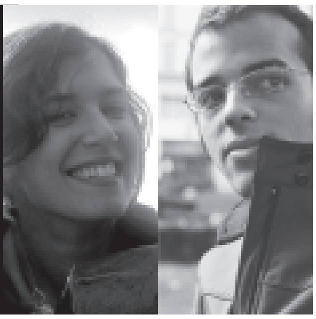
Whether it manifests itself in a workflow incorporating both digital and physical methods to different ends, or whether it is a transformation which takes place in the space between the two realms, the point of interaction between the digital and the physical is a special place. And it bears exploring. F/LOSS graphics and art fields are taking up that exploration in magnificent ways. In the world of F/LOSS, the space where the digital and the physical collide is becoming beautifully populated.

This issue of *Libre Graphics* magazine is about border cases, the role of intentionality and happy accident in the mingling of physical and digital, and any and all points of intersection. We're exploring the translation from the digital to the physical, the physical to the digital and the careful mapping of the two. We're looking at the place of history and the promise of the future in harmoniously melding our media.

We're looking at folds, sprays, cuts and prints. We're looking at mapping worlds onto each other. In short, in this issue, we're looking at collisions.

Branching out: journey notes

Ana Carvalho & Ricardo Lafuente



We now re-grab the reins after last issue's experiment with collaboration. It was a thrill to work with and feature the contributions of so many great practitioners; it was also a chance to go through and become an impromptu version control system, merging the various contributions into one coherent whole.

Version control is a theme that keeps popping up when discussing free and open creative projects, and it has, for good reason, been one of the main threads of discussion at the Libre Graphics Research Unit*. Not only does it provide useful metaphors for us to (re-)think our practice, but it is an effective practical tool for the management of complex editorial projects such as a magazine (though it is becoming unwieldy to manage a repository that takes up more than 8 gigabytes).

And now that we wrap up the first year-run of production with the release of the fourth issue, it is now time to sit down and wonder where we can go from here. This publication made evident that there is an emergent kind of artistic practice based on free culture that lacks proper stages where it can show itself, and the crossings between software tools, design theory and real-world practice remain underexplored.

There is still a lot to be said, and not just through words. The layout decisions and general graphic and artistic direction of Libre Graphics magazine are also a reflection of our unanswered questions and tentative answers. As an example, we've avoided exuberant experimentalism when conceiving the magazine's identity and graphic principles; this isn't because we're shy to experiment, but because we decided we'd be careful. Balancing reader expectations towards predictability and consistency with the designer's will to push the boundaries of graphic language is a frequent dilemma, and one that keeps popping up when we sit down to lay out a new issue: how far can we go without risking clarity and seriousness? And in what ways does our subject matter demand a new artistic approach? Our way to look for answers is to go on through slow, but steady steps.

We get closer to a satisfactory answer by taking risks and sometimes failing. We took note of our perceived failures (some of them pointed out by our readership), and are getting ready to sit down and, again, take a new risk in defining an improved identity for Libre Graphics magazine to be introduced in issue 5—the first issue of volume 2. We hope you're enjoying the journey, and we hope we can count on your curiosity and critical feedback since, in a good way, we're all lost together trying to figure out all the subtle ways that libre graphics culture is amplifying our collective possibilities and potential.

* From the LGRU web site: “The Libre Graphics Research Unit is a traveling lab where new ideas for creative tools are developed. The Research Unit is an initiative of four European media-labs actively engaged in Free/Libre and Open Source Software and Free Culture. This cross-disciplinary project involves artists, designers and programmers and is developed in dialogue with the Libre Graphics community.” (http://lgru.net)

LinuxFest Northwest

Bellingham, WA
April 28th & 29th

- Grassroots Linux gathering
- Exhibits of all flavors
- Presentations of all levels
- Prizes and after party
- FREE admission & parking
- FREE open source software
- Bring the whole family!

Hosted By

Bellingham TECHNICAL COLLEGE

linuxfestnorthwest.org

New releases

Blender 2.62

http://www.blender.org/download/get-blender

The latest version of Blender includes motion tracking, an improved interface for the game engine and a number of new uv editing tools, among other additions.

VLC 2.0 Twoflower

http://www.videolan.org/vlc/releases/2.0.0.html

A major new release from the ever-popular video player. Now with faster decoding, an enlarged list of formats (which includes HD) and higher quality subtitles.

GIMP 2.7.5

http://www.gimpusers.com/downloads

Call it a beta of the next GIMP 2.8. This devel version of GIMP gives a taste of what's coming next, namely faster performance, improved tablet support and a lot of bug fixes.

Ghostscript 9.05

http://www.ghostscript.com

With improvements to ICC colour rendering and font replacement, the newest version of Ghostscript has a number of design-sensitive additions.

Gnome 3.4

http://gnome.org

The popular desktop environment has come out with its first new release in six months, incorporating new menus and search functionality, as well as a more polished look.

What's new with you?
We're always eager to find out what designers, artists and others using and working with F/LOSS are up to. Tell us what you've done lately at ***enquiries@libregraphicsmag.com***

Upcoming events

We're very pleased to present a calendar of upcoming events which encompass all things graphic design, media art and F/LOSS. Given that there are few events which tackle all three subjects, we aim to offer you events where you can be the agent of change: the F/LOSS designer at a traditional design event, or maybe the designer at a predominantly software developer event.

28-29 APR

Linux Fest Northwest
BELLINGHAM, UNITED STATES
http://linuxfestnorthwest.org

2-5 MAY

Libre Graphics Meeting
VIENNA, AUSTRIA
http://libre-graphics-meeting.org/2012

18-20 MAY

SIGINT
COLOGNE, GERMANY
http://sigint.ccc.de

25-26 MAY

Flossie 2012
LONDON, UNITED KINGDOM
http://www.flossie.org

1 JUN

Drupal Business Summit
VANCOUVER, CANADA
https://www.drupalsummit.com

6-8 JUN

Linux Con Japan

YOKOHAMA, JAPAN
https://events.linuxfoundation.org/events/linuxcon-japan

8-10 JUN

Southeast LinuxFest

CHARLOTTE, NC, UNITED STATES
http://www.southeastlinuxfest.org

15 JUN

Ampersand 2012

BRIGHTON, UNITED KINGDOM
http://2012.ampersandconf.com

25-28 JUL

Fisl 2012

PORTO ALEGRE, BRAZIL
http://softwarelivre.org/fisl12

30 AUG 3 SEP

Ars Electronica Festival

LINZ, AUSTRIA
http://www.aec.at/thebigpicture/en

11-14 SEP

SVG Open & The Graphical Web

ZURICH, SWITZERLAND
http://www.svgopen.org/2012

9-11 NOV

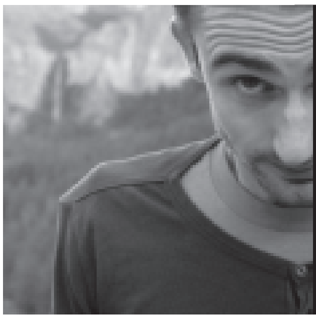
Mozilla Festival 2012

LONDON, UNITED KINGDOM
https://www.mozillafestival.org

10-18 NOV

HTMlles 10

MONTREAL, CANADA
http://www.htmlles.net/2012



Moral rights and the SIL Open Font License

Dave Crossland

Designers often express caution about Libre culture. Letting other people make modifications to their designs is unusual for them. This is especially true for type designers, since the harmony of a typeface design can easily unravel if changes are made carelessly. Copyright is the major legal restriction that stops others from making modifications. In some places, though, there is another restriction: moral rights.

Moral rights are explained on the English Wikipedia as “the right of attribution, the right to have a work published anonymously or pseudonymously, and the right to the integrity of the work.” In many places such as France, authors are not allowed to transfer, sell or give up these rights to other people or parties.

Recently I was discussing the SIL Open Font License with France's highest profile type designer, Jean Francois Porchez. He asked me if these moral rights conflict with Libre licensing. The SIL Open Font License is the most popular license for Libre fonts and—I believe—it does respect moral rights. Let me walk you through the OFL's compliance measures.

Section 2 of the OFL requires that “each copy contains the above copyright notice.” As a copyright holder, your copyright notice (which can include your URL and contact email) will remain with all copies and derivatives. This ensures that your attribution is made available. Additionally, there are FONTLOG text files commonly distributed with OFL fonts. These have a detailed description of the fonts, a timeline of releases, and details about all copyright owners and authors including those of any “parent” fonts. This is not required, but it is recommended by the OFL as a best practice and it really helps with good attribution.

Another side to attribution is a case in which the author *wishes* to be misattributed, in a way: anonymously or with a pen name. In this case, designers can simply use their handle, or “Anonymous,” in their copyright notice and other fonts' metadata.

A third side of attribution, what we might call bad attribution, involves the name of the designer of a parent font being used to promote derivatives. Section 4 of the OFL provides a guard against such misuse to “promote, endorse or advertise” a derivative.

Such attribution issues are quite straightforward. The concern is really the moral right to the integrity of a work. English Wikipedia explains that “preserving of the integrity of the work bars the work from alteration, distortion, or mutilation.” Allowing others to make alterations is fundamental to Libre culture.

Personally, I believe that saying an alteration of a typeface is a “distortion, or mutilation” is an entirely personal opinion. If I chop off the serifs from a typeface and respace it, I think it would be strange for anyone to say that I *mutilated* it. Others can say I did it poorly, that making a related sans is really a new design project and many other details of the typeface must be changed to re-harmonize it. But for my personal identity to be so wrapped up in my work that I could have hurt feelings in this way seems bizarre to me. Yet many artists and designers do feel strongly that the work they have made should not be altered by other people at all, even in the countries where moral rights are not instiuted in law, like the United States.

The identity of a designer is associated with his work. In type, there is a common association of the name of a typeface with the name of its primary designer. Making a change naturally invites altering the name to reflect differentiation. Using the original name in part could cause the association with the original designer to carry over too far. The OFL has a solution to this. It allows designers to set Reserved Font Names, which act in a similar way to trademarks.

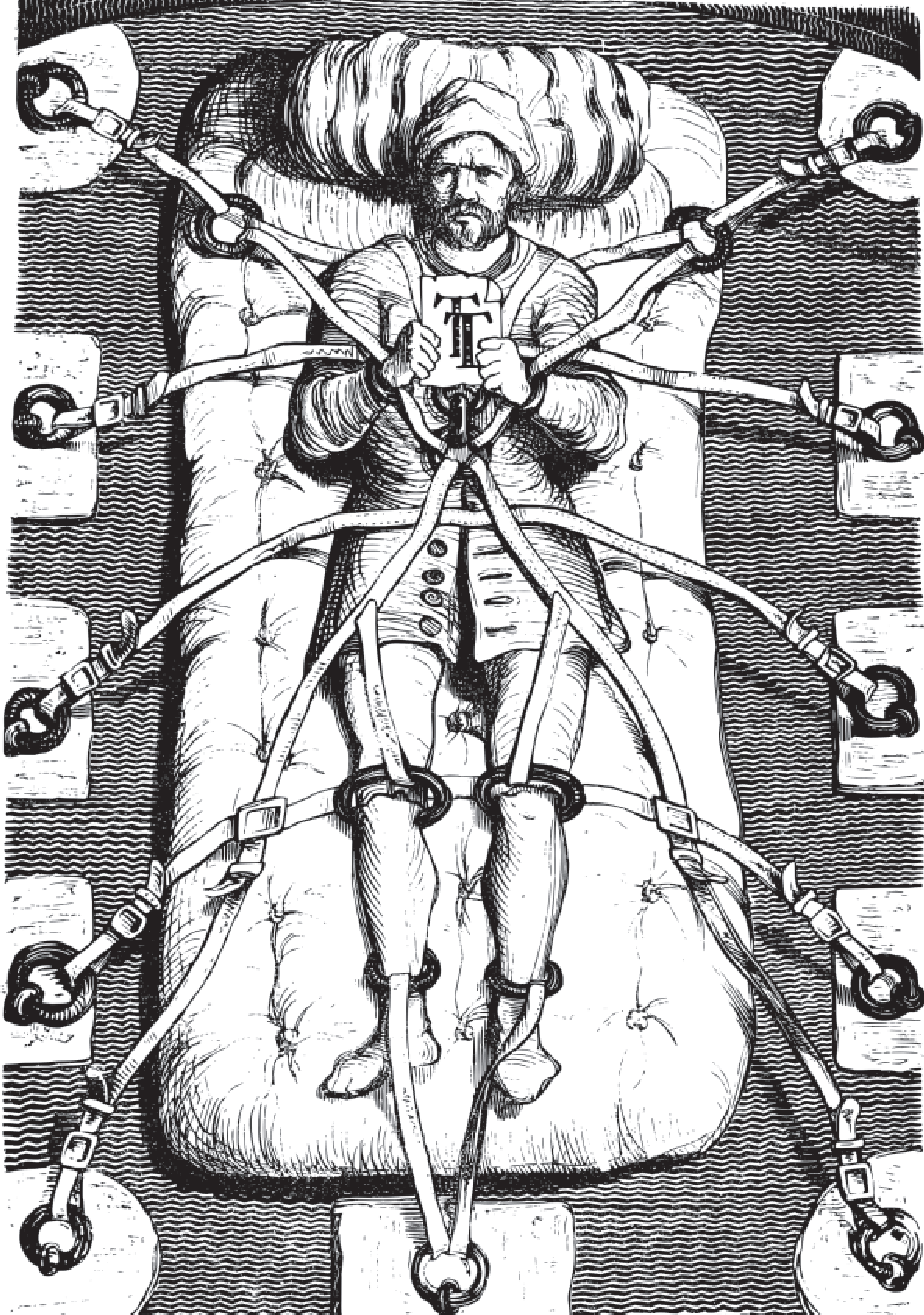
If you declare a Reserved Font Name, the OFL ensures that when your font is modified then the name you chose for the font will *not* be used—unless you give additional permission.

This naming restriction encourages people who make ‘corrections,’ who adjust minor details in the font—tweaking a kerning pair here, adding some metadata there—to contribute their changes upstream, back to the original designer who maintains the font under its most well known name.

In all of these ways, among others, I believe the OFL respects moral rights, while maintaining the principles of Libre culture.

The author of this column is not a lawyer. This column is meant as a set of guidelines and opinions, not as legal advice.

Dave Crossland believes anyone can learn to design great fonts. He is a type designer fascinated by the potential of software freedom for graphic design, and runs workshops on type design around the world. http://understandingfonts.com


TT

Will these hands never be dirty?

Eric Schrijver

Graphic design is a nostalgic field.
Even in the art schools, the students want to make books and posters. Designing for the web has little prestige. I could say that I want students to design for the screen, and to actively engage with their digital tools, but I first need to know what it is that makes it so attractive to design for the printing press.

Books and posters nowadays start their lives on a computer in proprietary software. Many of my colleagues see the software as a neutral tool, subservient to their creativity. Therefore, the software can be used as-is. For me, software is a piece of culture, an embodiment of a certain way of thinking. The software partakes in the creation. A truly rich visual culture can only come about if designers manipulate, appropriate and subvert the software technology they use.

Hacking analogue technology is a physical affair. Cracking open software requires a different sort of interaction, with programming interfaces and computer files. It requires a new set of skills that takes time and enthusiasm to attain. It is this enthusiasm that is often lacking. In fact, my students often seem scared of digital technology.

So what are today's nascent designers scared of? The comparison with their attitude to printing technology shows it is not necessarily technology in general that designers are frightened of, nor is it geekiness. Designers actually take pride in the geeky details of their craft when they are related to the printing process, knowing about things like spot colors, paper stocks and binding methods.

Is it a matter of differing cultures?
Even though design is applied mathematics, most design majors study the humanities in high school.
Code seems to belong to this other world, the world of the kids who choose mathematics. The other geeks.

If the divide is social, then a gentle introduction to the other culture, the culture of programming, itself embedded in the culture of science and mathematics, should form part of a contemporary design curriculum.

Or another strategy, can we force the students to get their hands dirty?
With code? No more mockups.
That's an efficient way to introduce the nature of the digital. There is always the question of whether designers should learn to code. I think they should.

As a student, I came across a printing press that worked with movable type.
I spent a day setting a simple poem.
It's dirty, precise, frustrating work.
At the end of the day I printed my poem, and only after I had cleaned the press, I spotted the spelling error.

As tedious as the process had been, this day taught me so much about the nature of printing technology.
My understanding of my profession really deepened. I know why uppercase is called uppercase (uppercase letters are stored above the lowercase ones); why leading, the space between lines, is called leading (it is strips of lead).
I have an understanding of how all of the classical book layout conventions are related to the process of setting a block of movable type.

Even though I never set anything in movable type again, I understand printing technology to a further extent.
The same is true for code.
Going through the tedious process of writing a computer program will change your understanding of the medium you work with all the time. Your dirty hands will forever influence any interaction you have with programmers.

I say “getting your hands dirty” and it is obvious that it is a metaphor. Or is it? At any rate, anything I do, I do it with my body. My head is but a part of my body (I'm glad that HTML5 acknowledges this). And it is not necessarily so that my interaction with a computer is more detached than that with another device. To see what is going on, we need to move beyond tight distinctions between analogue and digital, between concrete and abstract. Like drawing, when you think and you do at the same time [sic-ed].

I don't know how physical programming actually is. What kind of software interaction is it, that is the equivalent of prying something open, or to turning it on its head? I do know that all my descriptions of what I want include this physical dimension. What I want is a relation to software that is more daring, and ultimately more intimate.

Eric Schrijver (Amsterdam, 1984) is a graphic designer and a performance artist. He is inspired by programming culture. Eric teaches Design for new media at the Royal Academy of Art in The Hague, and is a member of the design collective Open Source Publishing.
http://ericschrijver.nl

Notebook

Where

CulturaDigital.Br Festival,
Rio de Janeiro, Brazil.

What

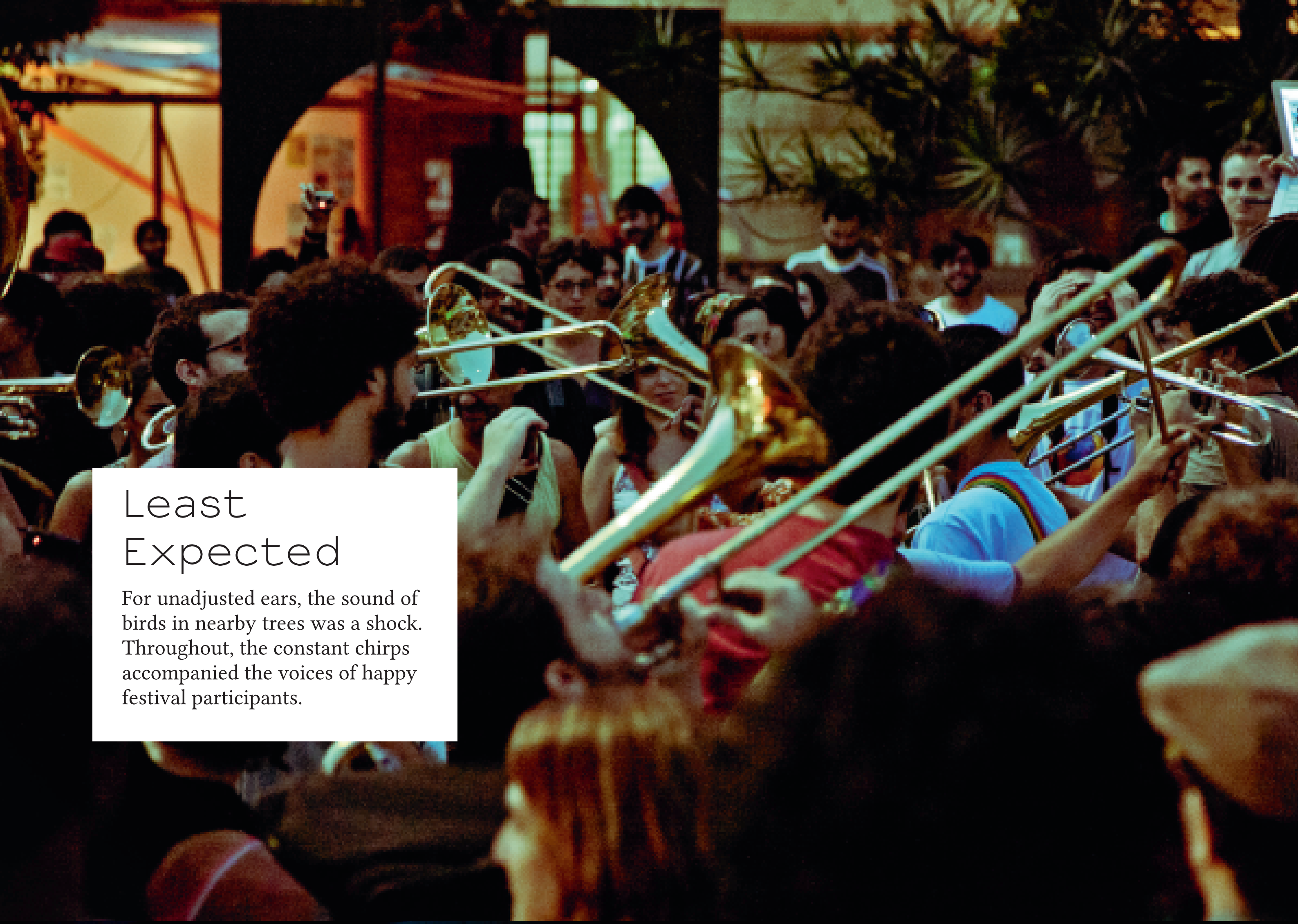
Brazilian free culture festival, bringing together creators and community organizations.

Best Use of Foam

The festival's logo was rendered in giant letters, made of foam and perfect for sitting. At the end of the weekend, some participants went home with parts of this unusual type treatment.

Best Ambience

The festival took place almost entirely outside, on the grounds of the Museum of Modern Art in Rio de Janeiro. The museum itself is raised off the ground, with space below. Crowds of festival-goers wandered and worked in the shadow of a floating gallery.



Least Expected

For unadjusted ears, the sound of birds in nearby trees was a shock. Throughout, the constant chirps accompanied the voices of happy festival participants.

Easiest How-To

Mimi Hui, a member of NYC Resistor took the opportunity during a talk about RFID technology to explain the construction of aluminum foil Faraday cages.

Small & useful

There's an adage in the software world: programs should do one thing very well. In that spirit, we offer you a round-up of small and useful programs and resources which do one thing particularly well.

Not a tool, but a resource, ***The Public Domain Review*** is a gateway to interesting works in the public domain. Saving you from slogging through all the public domain work available, the *Review* presents curated collections of particularly interesting work.

http://publicdomainreview.org

GimpLensfun is a handy GIMP plugin using the LensFun library and database. GimpLensFun brings easy correction of lens distortion right into your raster graphics editor.

http://lensfun.sebastiankraft.net

Unpaper is a command line tool designed to correct issues in scanned book pages. It removes excess darkness around edges, rotates and de-skews.

http://unpaper.berlios.de

Pdf Chain is a convenient addition to another small and useful tool, Pdftk. PDF Chain provides a graphical interface for the command line power of Pdftk, allowing users to conveniently merge, split and make other modifications to multi-page PDF documents.

http://pdfchain.sourceforge.net

Gtk Vector Screenshot taking module does what it says on the tin: it takes vector-based screenshots. No more flat, un-scalable screenshots.

https://gitorious.org/gtk-vector-screenshot

The finished and unfinished business of OpenLab ESEV

by Nelson Gonçalves and Maria Figueiredo

OpenLab started in an environment characterized by a lack of knowledge of the existing Libre alternatives and work habits exclusively built around proprietary software. The desire to promote informed choices was the motivating challenge for the founders. Today, OpenLab activities are implemented on a voluntary basis and users include a core group of five teachers, several students and former students and other collaborators.

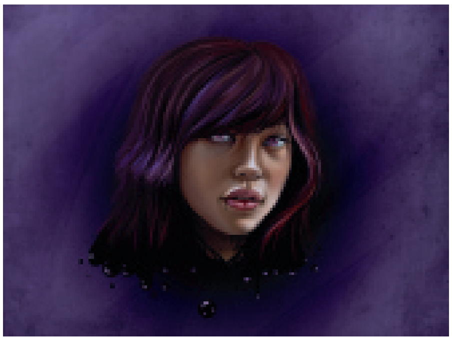
Since OpenLab started, we've witnessed kindergarten children playing with MyPaint and Luciole, elementary school students playing with Toonloop and Pencil, secondary school students playing with Blender and Audacity. Our own teachers and students have a place to look for support and answers. Concepts like F/LOSS and Creative Commons are now familiar to many students and teachers, even to those who don't use them. Several F/LOSS projects like Zotero, Blender, GIMP, Scribus, Inkscape, Audacity, Processing and Aptana Studio have made their way into the classroom.

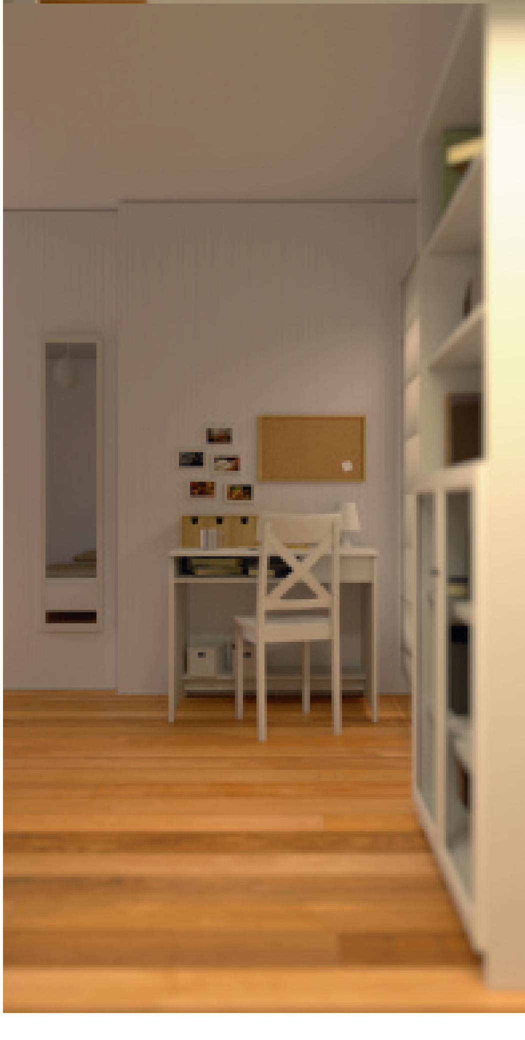
In all the workshops ESEV has run, one of the earliest stands out and exemplifies our aims and how we want to work.

Saturday morning, May 8, 2010, the school looks empty. Not a surprise: no classes on Saturday means no students. But is it really empty? Maybe not. Two dozen students are gathered at a classroom door. We have graduate and postgraduate students from teacher education, arts & multimedia and media. Many of them don't know each other. Most of them have brought their own laptops and some have even brought graphic tablets. They've come for the "Introduction to digital painting with Free Software" workshop.

Based in the School of Education of the Polytechnic Institute of Viseu (ESEV), OpenLab ESEV aims to establish a platform to foster and support the use of F/LOSS, Free Culture and more flexible licenses for creative and educational purposes. ESEV, located in Viseu, a city in the northern-center of Portugal, has over 1500 students and 105 teachers and offers nine undergraduate and 13 graduate programs. Programs are related to education, cultural animation, arts & multimedia, media, sports and public relations.

http://www.esev.ipv.pt/openlab





The trainer is already inside the classroom, checking to make sure the graphic tablet and the interactive whiteboard are working. Natacha Cindy, arts & multimedia student, discovered F/LOSS some months earlier. Armed with a passion for drawing, she immediately fell in love with Alchemy and MyPaint. She embraced the challenge of sharing her learning with other students without hesitation. It's her first experience as a trainer, the first of several.

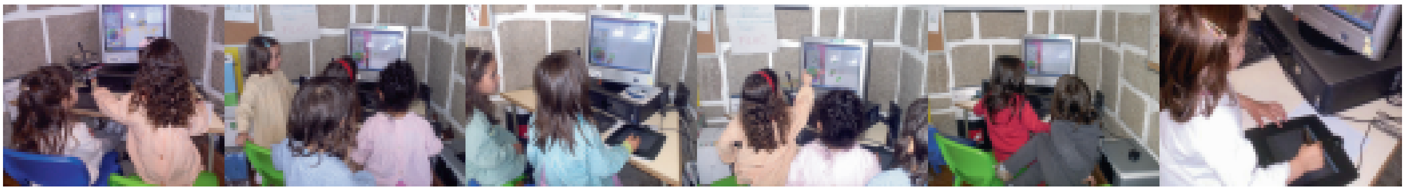
For the next three hours, inspired by Natacha's skill and enthusiam, the students explore Alchemy and MyPaint. For most of them, it's the first time using those programs. For some, it's even the first time using a graphic tablet. For all of them, it's the first peer-guided workshop they've attended. Natacha has prepared a short tutorial for those who want to train more later or work at their own pace. However, most of the time they try to keep up, individually or in pairs, asking questions and requesting personalized support. It's tiring, mostly for Natacha, but rewarding for everyone.

The results went far beyond the drawings produced during or after the workshop. The most important outcome was that Alchemy and MyPaint started to really exist for several students. The teacher education students took MyPaint to their kindergarten and used it in classroom activities. The students more related to artistic and creative activities installed these tools and still use them. Today, dozens of the students know about or have tried MyPaint and Alchemy. These tools found their way into the classroom and students' laptops. The best part is that this also happened with other tools, other students and other workshops.

This story can't be finished without some last words about Natacha. Today, Natacha is professionally working as an illustrator and doing her masters degree. She still uses MyPaint and Alchemy on a daily basis, although the last time we talked, she was very impressed with Krita. She's even managed to bring MyPaint to her workplace (a large book and multimedia publishing company) and integrate it in her workflow.

She's probably going to read these lines so it's only fair that we tell her: thank you and we're very proud to have you as an OpenLab ESEV friend and collaborator.

It's true that proprietary software is still predominant and that there's still a lot of awareness-raising and capacity building to do but we're proud of what we've accomplished and learned so far. To us, using F/LOSS is a statement about the world we live in and how we choose to live in it.



MAGNETIC
VAR 4°15'W (1985)
ANNUAL DECREASE 8'
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
0
30
60
90
120
150
180
210
240
270
300
330

Best of SVG

Even blank maps can tell a history.

Just browse the colletion of Blank World Maps offered by Wikimedia Commons and OpenStreetMap, set your eyes on the ever changing boundaries of countries and nations.

Being SVG, they are doubly useful: one, you get resolution-independent maps, allowing you to scale them without mercy. Two, SVG makes them perfect for including in a web application or visualization, creating experimental programs around geographic data, and for clever maps that embed additional cartographic data in the file itself.

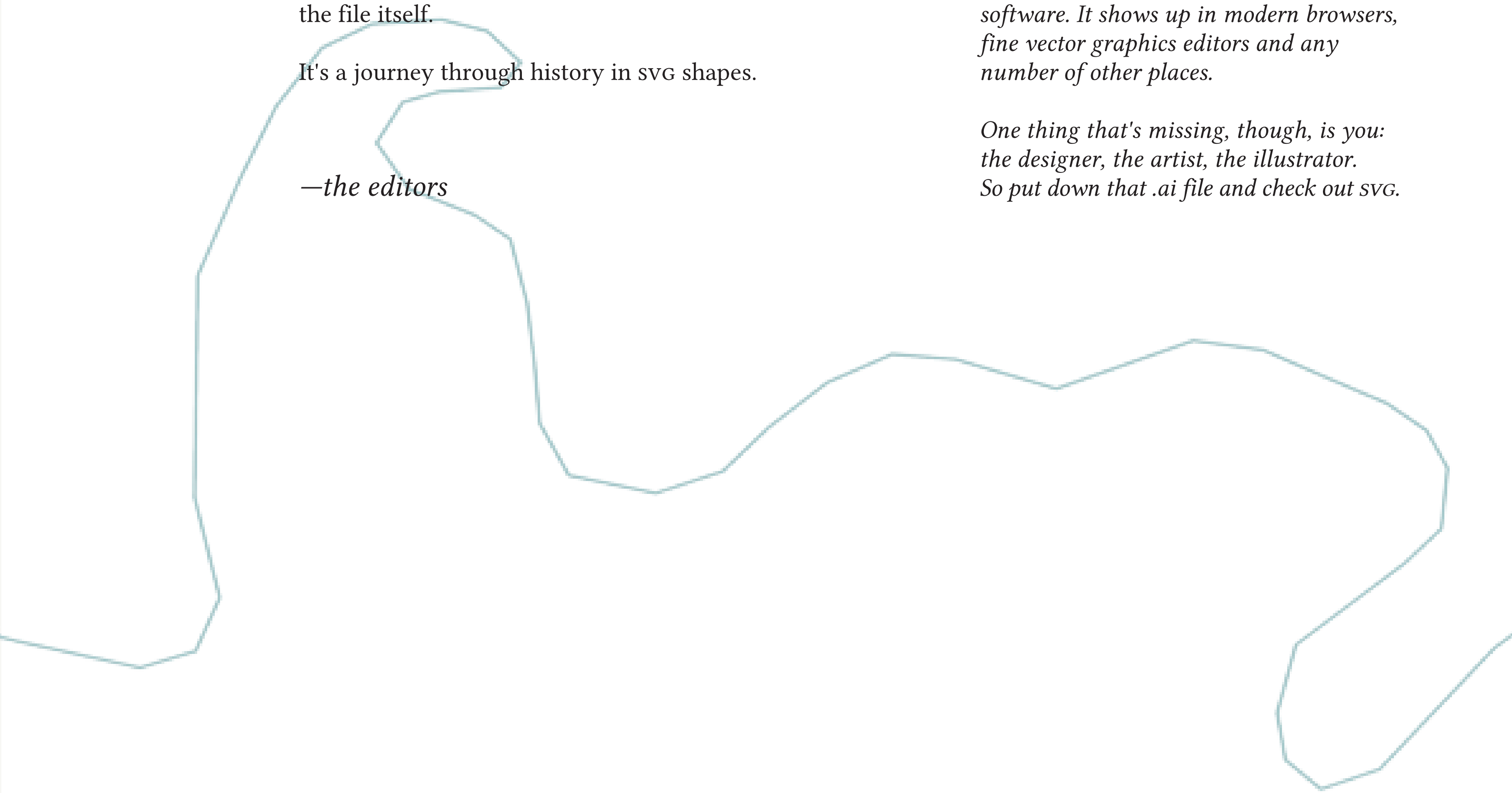
It's a journey through history in SVG shapes.

—the editors

We at Libre Graphics magazine have a thing for open standards. We like their transparency and their interoperability. We like that, with a well documented standard, everyone has an equal chance to play nicely together.

That's why we like SVG so much. It's a well developed, well supported standard brought to us by the World Wide Web Consortium (W3C). It's available for implementation by anyone developing software. It shows up in modern browsers, fine vector graphics editors and any number of other places.

One thing that's missing, though, is you: the designer, the artist, the illustrator. So put down that .ai file and check out SVG.



ABCDEFGHIJKLM
NOPQRSTUVWZ
abcdefghijklmno
pqrstuvwz
0123456789
!?@#%&*

EXO

&

In such a country, you will perceive at once tha
should be anything of what you call a "solid" ki
suppose that we could at least distinguish by s
and other figures, moving about as I have desc

EXO THIN

Place a penny on the middle of one of your tables in Space; and leaning over it, look down upon it. It will appear a circle.

EXO EXTRA LIGHT

When I was in Spaceland I
heard that your sailors have
very similar experiences wh
they traverse your seas and
discern some distant island
coast lying on the horizon.

EXO LIGHT

The far-off land may have bays, forelands, angles i
and out to any number and extent; yet at a distanc
you see none of these (unless indeed your sun
shines bright upon them revealing the projections
and retirements by means of light and shade),
nothing but a grey unbroken line upon the water.

EXO BOOK

Well, that is just what we see when one of our triangular or other acquaintances comes toward us in Flatland.

EXO MEDIUM

As there is neither sun with us, nor any light of such a kind as to make shadows, we have none of the helps to the sight that you have in Spaceland.

EXO DEMI BOLD

If our friend comes closer to us we see his line becomes larger; if he leaves us it becomes smaller: but still he looks like a straight line; be he a Triangle, Square, Pentagon, Hexagon, Circle, what you will—a straight Line he looks and nothing else.

EXO BOLD

You may perhaps ask how under these disadvantageous circumstances we are able to distinguish our friends from one another: but the answer to this very natural question will be more fitly and easily given when I come to describe the inhabitants of Flatland.

EXO EXTRA BOLD

For the present let me defer this
subject, and say a word or two abo
the climate and houses in our coun

EXO BLACK

Excerpts from *Flatland: A Romance of Many Dimensions*, by Edwin A. Abbot

Natanael Gama talks techno-fonts and the benefits of Libre

by Dave Crossland

Dave Crossland: *Could you describe Exo?*

Natanael Gama: Exo is a contemporary geometric sans font family with a sleek, technological look. It has a peculiar look, but it can be very versatile. It is a very complete font family, with over 700 characters supporting most Western, Central and Eastern European languages—and it can still go further!

This coverage has been reached for all 9 weights, the maximum possible on the web. Each weight has an upright style and the corresponding true italic style. Many techno fonts have only the simple slanted oblique companion styles, but for the highest typographic quality it is essential to have true italics that are drawn specifically.

The OpenType features include small caps, standard and discretionary ligatures, stylistic alternates, old-style figures, tabular figures, fractions, and more. These features will be available both as true OpenType fonts and also as simple fonts for use on the web.

Do you feel you achieved what you wanted from the design?

I feel I did, but I think in the end, only the users can say if a type designer succeeded or not.

I designed it for use in magazines and magazine-style websites, advertising. I made it for display use, and it can work to typeset small amounts of text. It could even work well as a corporate identity typeface for the right company.

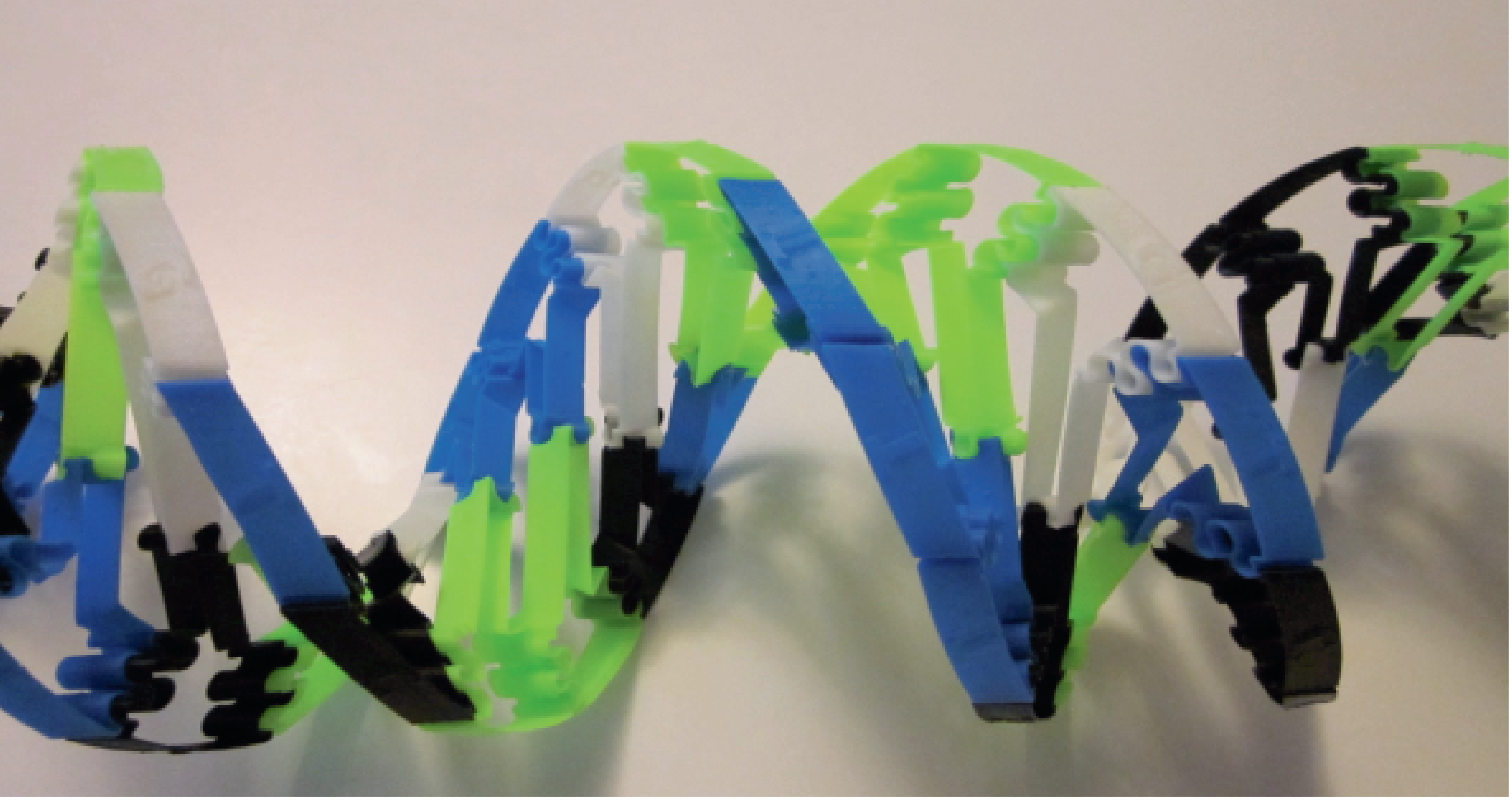
It's an impressive project. What's the story behind it?

When I started my career as a freelance graphic designer I faced the problem of not having enough good fonts to do decent work. When you are starting out, you don't have that much money to invest in good typography, so many people break the license agreements of the fonts they use. Since I wanted to be as legal as possible, I got into the world of Libre fonts—and I am so thankful to those designers who are generous enough to share their work with the world!

But along the way I have found that right now there are very few Libre fonts you can count on. This was a problem for me, and for sure it is the same for all those serious designers who avoid pirate fonts. The cheap stuff very often lacks important features to do a good job, but it can't be changed or fixed. The amazing thing about Libre fonts is that they are both available at zero price and can also be improved, like Wikipedia!

Having this situation in mind, I decided to invest my own time into the design of a good font family to help all designers. So I made Exo for myself and designers like me—a font family for those who want to do a good job, but may lack the fonts to do it properly. I decided to make this a Libre font so the designers can do great work with a legal font family, and learn the value of high quality type.

Showcase

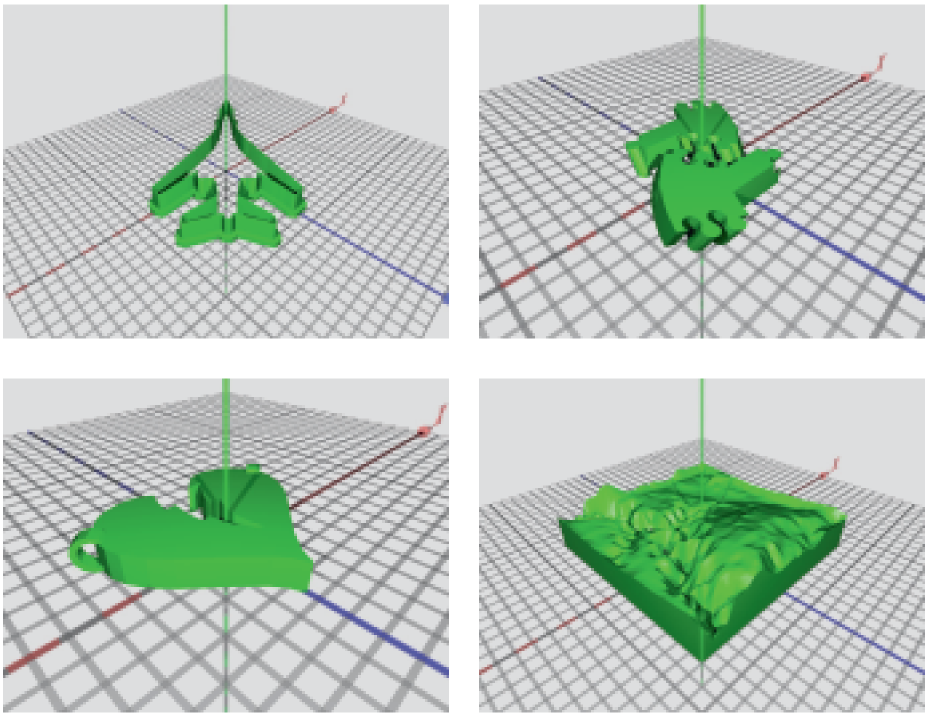


The graphic side of 3D printing

There's a little revolution taking place in basements, garages and hackerspaces. 3D printers—machines which turn digital meshes into physical objects—are getting cheaper, better and more popular. We've collected samples of some strikingly graphic 3D prints, designed with F/LOSS tools.

All of the projects featured in this showcase are available from Thingiverse, under permissive licenses, for printing and modification.

http://thingiverse.com

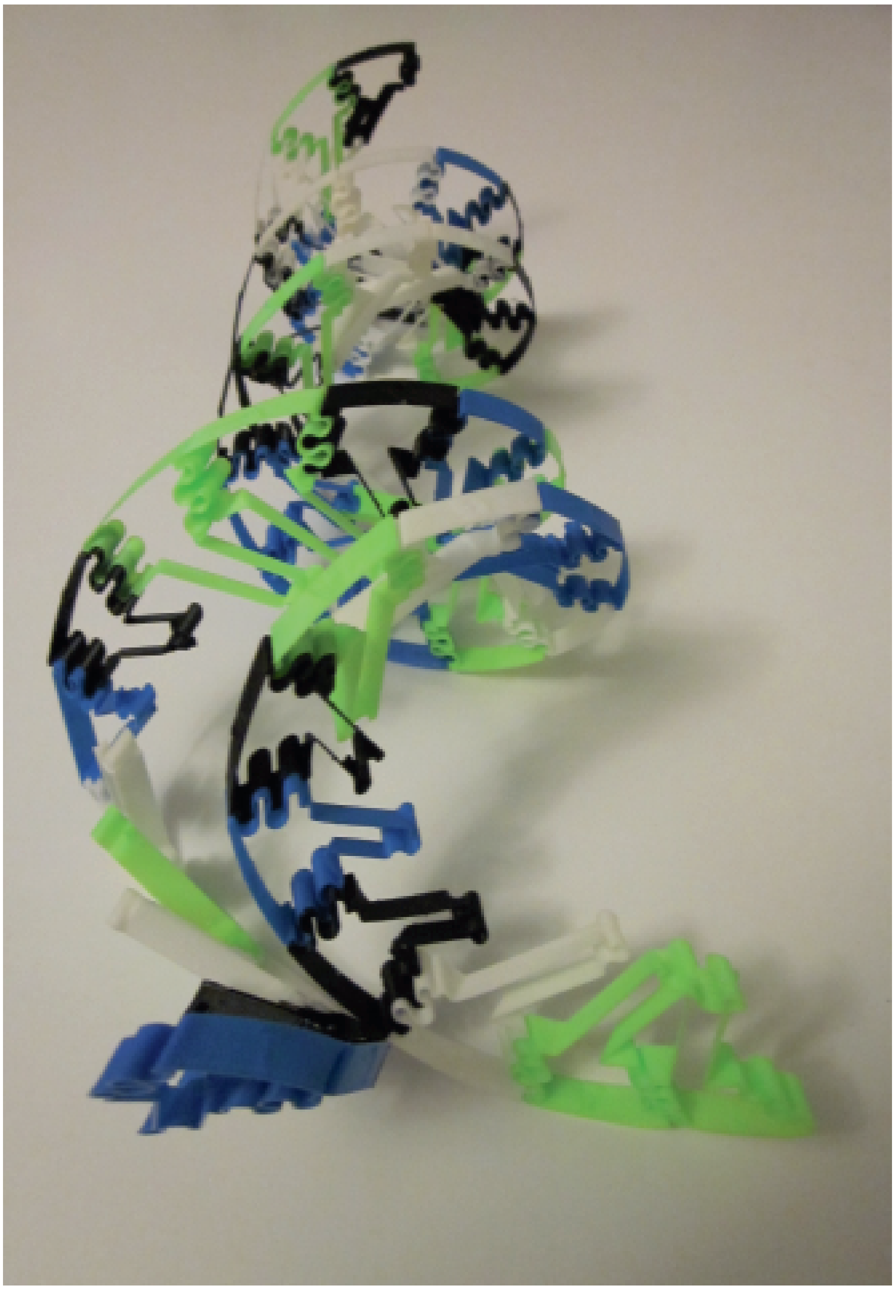


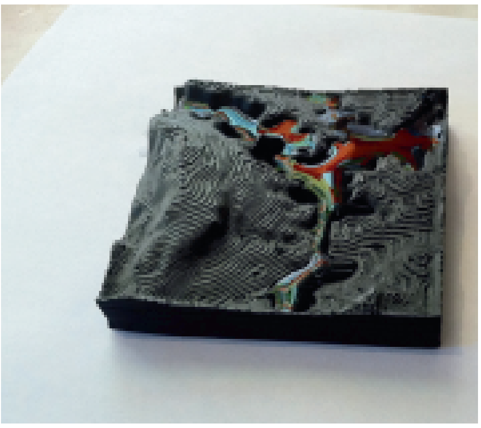
Airplane cookie cutters (Inkscape), user: emmett on Thingiverse: http://www.thingiverse.com/thing:11098

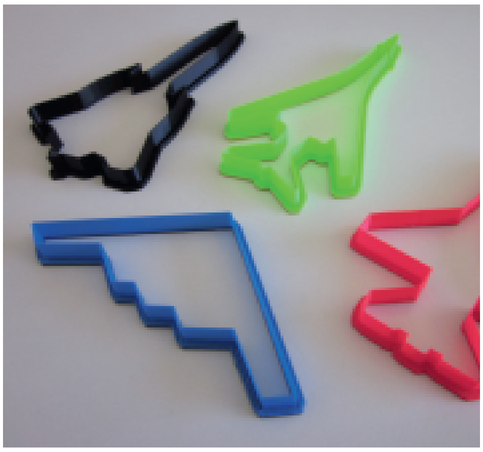
DNA Playset (OpenSCAD), Emmett Lalish: http://www.thingiverse.com/thing:17343

I heart lightning (Inkscape & OpenSCAD), Amy Hurst: http://www.thingiverse.com/thing:17495

Lake and mountain topography (Blender), Luke Chilson: http://www.thingiverse.com/thing:7207







You need great tools to make great tools.

Paper.js: building, designing and the browser

An interview with Jonathan Puckey by Manufactura Independente

Manufactura Independente: *Your approach to digital design tools is rather specific, in the sense that you mostly create actual, very specific graphic tools instead of filters: user input is thus a very strong part of the outcome. How do you see the issue of control in the tools you make?*

Jonathan Puckey: When I find myself doing filters, I work mainly with the programmer part of my brain. I rewrite the code, press a button, check the result, rewrite the code, press a button, check the result. When I make tools, I am able to split the process in two: first I get to design an environment to work within, then I (or others) get to push those boundaries to their limits by working within it. It becomes a collaboration once more.

What's your criteria for publishing a tool vs. keeping it for your own use?

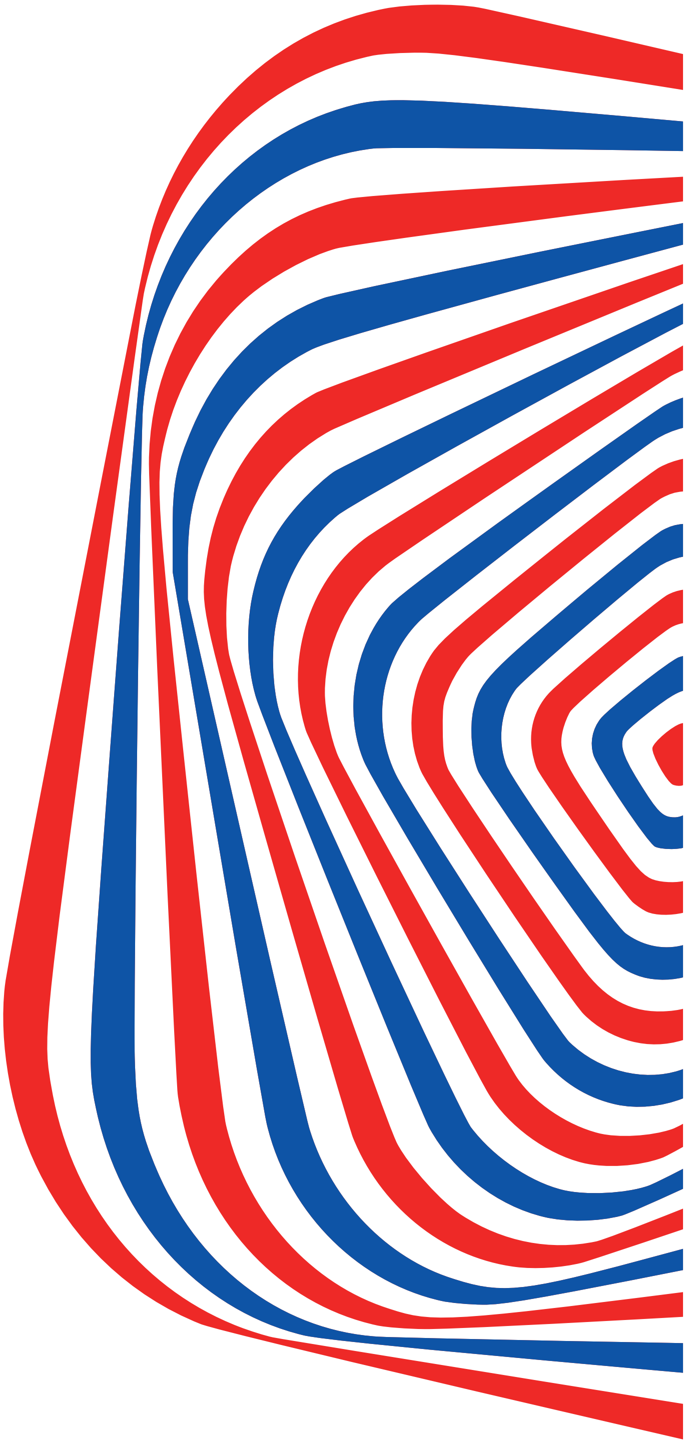
It depends. I am happy to publish a tool when I feel that there might be other people out there who could use it in different ways than I could. So there is something missing, that needs to be filled in by the designer. This also means that the user of the tool is in the forefront, not the tool itself.

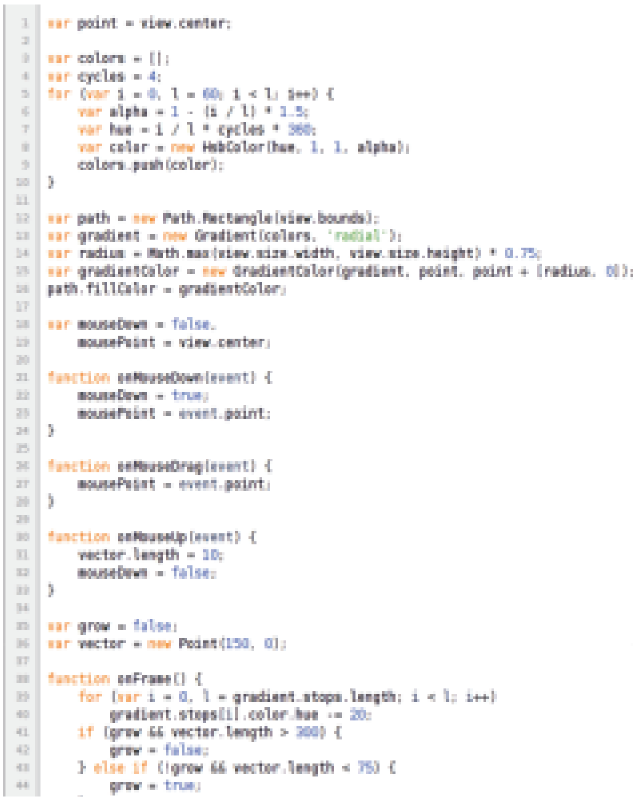
What defines whether you'll build a single-use tool or a re-usable one? Do you find yourself creating a base set of more general-purpose scripts, or do you start from scratch for a new project?

I really believe in investing time into contributing to the platforms that I use to build tools in. For graphic design tools this is Scriptographer, for web based vector graphics this is Paper.js and for backend server and automization stuff I am currently using Node.js, which has a very rich community. The code I reuse tends to be built into the libraries I work with. If something is missing in my tool belt, I am happy to contribute missing features. In the end, you need great tools to make great tools.

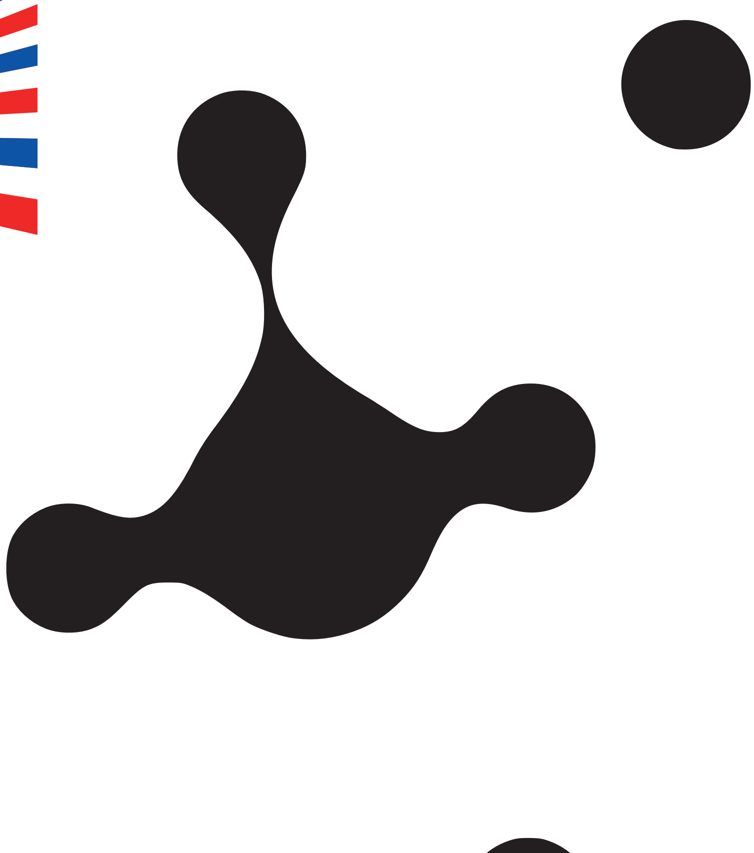
Would you agree there is a gap between the mindset of the tool creator and the tool user? How do you deal with it? Or does that distinction make no sense to you?

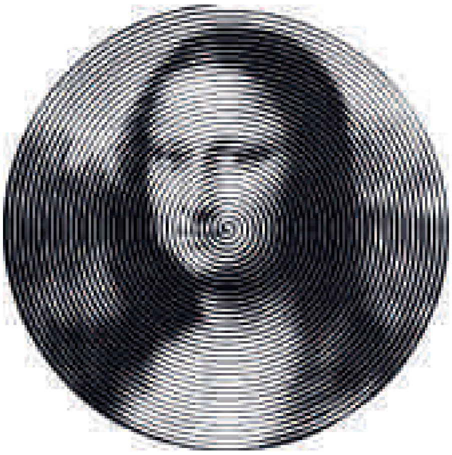
Sometimes, as a tool creator, it can be difficult to put the same amount of attention into using the tool as you put into developing it. Then it can be very enjoyable to see others working with your tool, pushing it to where it

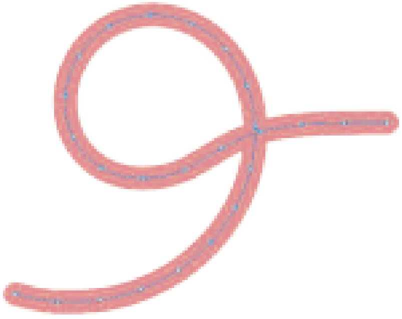




```
var point = view.center;

var colors = [];
var cycles = 4;
for (var i = 0, l = 60; i < l; i++) {
    var alpha = 1 - (i / l) * 1.5;
    var hue = i / l * cycles * 360;
    var color = new HsbColor(hue, 1, 1, alpha);
    colors.push(color);
}

var path = new Path.Rectangle(view.bounds);
var gradient = new Gradient(colors, 'radial');
var radius = Math.max(view.size.width, view.size.height) * 0.75;
var gradientColor = new GradientColor(gradient, point, point + [radius, 0]);
path.fillColor = gradientColor;

var mouseDown = false,
    mousePoint = view.center;

function onMouseDown(event) {
    mouseDown = true;
    mousePoint = event.point;
}

function onMouseDrag(event) {
    mousePoint = event.point;
}

function onMouseUp(event) {
    vector.length = 10;
    mouseDown = false;
}

var grow = false;
var vector = new Point(150, 0);

function onFrame() {
    for (var i = 0, l = gradient.stops.length; i < l; i++)
        gradient.stops[i].color.hue -= 20;
    if (grow && vector.length > 300) {
        grow = false;
    } else if (!grow && vector.length < 75) {
        grow = true;
```







deserves to go. In the end, we are all tool creators on different meta levels. The end user of the software develops new ideas on how to apply the tool (also in combination with other tools) and therefore redefines the tool itself.

You've recently been involved in the development of the Paper.js project with Jürg Lehni, which brings the concepts of Scriptographer (a plugin to script Adobe Illustrator using JavaScript) to the web and the browser. How does the switch from a desktop environment into a browser canvas change your perceived boundaries of what you can build?

It changed more than we at first thought it would. When we did the first tests, we were only really thinking of creating generated static images. All at once we were seeing 60fps animations come out of our little framework. That was very exciting and opened up possibilities for all kinds of things we weren't thinking of in the first place. We have also plugged Paper.js into the server side using Node.js, which allows for things like video compositing and maybe in the future, PDF generating.

Would you say it makes possible new aesthetic directions?
Have you found, or come up with, surprising and unexpected uses for Paper.js so far?

Currently, I am quite interested in the possibilities of developing design tools for clients. I find the idea of a client working with a design tool within a controlled environment very interesting. Balancing freedom and restriction, etc. Paper.js allows me to develop design tools for clients, without the need for them to purchase expensive and complicated professional software.

What are you busy with now?

I recently started a new graphic design studio called Moniker (http://studiomoniker.com) together with Roel Wouters and Luna Maurer. I am currently working on some meta-ball Paper.js code for a cultural website. We have also started development of a text-file based CMS, which we aim to use within our studio for small portfolio websites for clients.
We just received funding to work on a music video which uses some interesting backend video compositing technology. What else... Oh yes, we are currently looking for some amazing programmer/designer interns who would like to collaborate with us on these things.

The Baltan Cutter Or the thin line between intuitive and generative

by Eric de Haas

In 2010, I began to work on a new visual identity for Baltan Laboratories, a lab for art, design and technological culture based in Eindhoven, the Netherlands. During this research process, more and more examples of generative graphic design started popping up, which I started to investigate as an approach that could be used for Baltan. Since Baltan is an organisation that profiles itself at the intersection of art, design and technological culture it seemed very appropriate to look into these new ways of dealing with a visual identity. Whereas a traditional graphic identity is defined by a logo, choice of colors, typeface and a set of rules for applying these elements, we are witnessing a strong shift towards more organic/random identities. This shift has been triggered by a change in available tools and mediums, and a move from print to screen.

Although fascinated by a lot of good generative design, such as the MIT Media Lab identity created by TheGreenEyl (with E. Roon Kang), I found myself less interested in the generative process and more in the moment when the design is deemed to be finished. This is something that is determined, in the case of generative design, by the script. Where in generative design creativity takes place in the very beginning and middle, creating a concept and writing the code, it leaves little room for intuitive decisions at the end of the process. This endphase of a design process, where you can decide to redo a whole layout or change all the colours, is what ultimately speaks to your intuition as a designer and what can be part of a signature. A logical step from there was to design a tool which would have some of the same features as a generative tool, but still needed the human hand and eye to complete the outcome, a tool that would need the specific skill of the user.

Good examples of this type of tool can be found in many Scriptographer tools, where the hand or the operation of a mouse/pen can strongly determine the output. Inspired by this, I contacted Jonathan Puckey, a co-developer of the Scriptographer platform, and asked him to help us create a tool. The topic of generative versus intuitive triggered some interesting conversations with Jonathan, which fortunately made him decide to work with us on this project. We then entered the most difficult phase of deciding what this tool should do.

Jonathan and I started a blog that was used to document a period of correspondence between the two of us as a way to develop our thoughts around the Baltan tool. I would write a letter/post that would be answered by Jonathan in the form of

a script/tool. (This correspondence can be found at http://baltanlaboratories.tumblr.com)

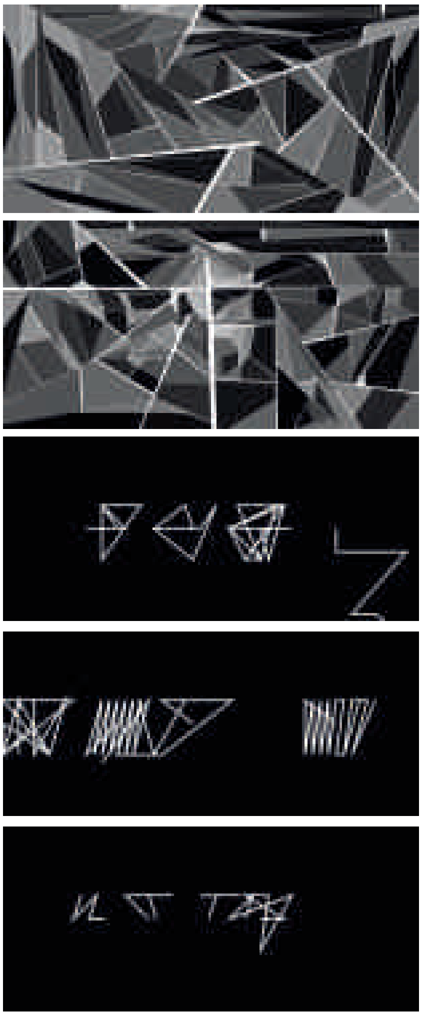
We started by looking at the first tool known to mankind, the bone or stick that was used for beating, the simple function of hammering, which could be used either as a weapon or as a tool to crack things. We found out that the old saying "all tools can be used as a hammer" didn't quite work for digital tools, but tried to simulate this by developing a version that could create but also destroy. From here we drifted off into the weird evolution of the mouse as a drawing tool. What the mouse basically did was separate the hand from the eye. After then doing some experiments where the keyboard would function as a drawing tool, and subsequently finding out that this would minimize the chance of making mistakes, we started questioning whether or not mistakes are essential in a drawing process or if this is a romantic idea triggered by the feeling that we have too much control over our toolset.

This feeling of too much control made us wonder if this could be the reason that for the last 10 or 15 years there have been a lot of projects appearing that try to have unpredictable or generative output. Could the lack of traces or textures common to analogue tools, like the inconsistent ink flow of a pen or a brush, trigger an urge for unpredictability in relation to their digital counterparts? Do we actually need to turn things around? Do we need a tool that has to learn how to complement its user, a tool with a memory, a tool that learns? But memory is also created by wielding the tool. Each outcome created using the tool shifts its potential and therefore becomes part of its memory. This can only happen when the tool provides us with a logical framework to work within, and only when the outcome can be considered to be a layered series of decisions with traceable influences.

BEYOND DATA
Exploring new ways of visualising and embodying digital data
The Tools Series

This was why drawing with a keyboard felt unnatural: all decisions were already made by the engineer. We needed to find a way to emphasise the decisions, maybe by limiting them. Maybe the prehistoric cutting tool contained more than enough choices. Maybe it represents the perfect example of a tool where the person operating it creates a memory and learns along the way, learning to work quicker or more detailed.

The final version of the Baltan Cutter stayed very close to one of the first sketches and can be best described as a 2D sculptor's tool. It basically has two functions: cutting and chipping. Both functions operate in a very clear way so that you are actually able to sculpt if you learn to operate the tool, but it also leaves room for intuitive cutting. Although we've just started using the Baltan Cutter for Baltan's identity-related items, we are finding out that the simplicity, limitations and the learning process of handling the tool create fresh outcomes and leave a lot of space for intuition and creativity. By consistently applying this output in different ways it also has a strong effect on the identity because it generates graphic imagery which have a certain level of autonomy and authenticity which makes it very recognisable in its appearance.



HAVE YOUR SAY NOW THAT WE'VE FINISHED VOL.1 AND ARE GETTING READY FOR VOL.2!

Submit your work, essay,
article or random piece.
you're the community,
you're our feature.

SUBMISSIONS@LIBREGRAPHICSMAG.COM

We want your feedback.
Tell us what you think.
Where we could do better.
Where you could help.

FEEDBACK@LIBREGRAPHICSMAG.COM

Distribute the magazine.
Get your library to order a subscription.
Spread the word.
Help us reach out.

ENQUIRIES@LIBREGRAPHICSMAG.COM

Folds, impositions and gores: an interview with Tom Lechner

In May 2011, Femke Snelting, Pierre Marchand and Ludivine Loiseau interviewed Tom Lechner, the creator of Laidout.

Femke Snelting: *What is Laidout?*

Tom Lechner: Well, Laidout is software that I wrote to lay out my cartoon books in an easy fashion. Nothing else fit my needs at the time, so I just wrote it.

FS: *It does a lot more than laying out cartoons?*

TL: It works for any image, basically, and gradients. I guess it's two forms of laying out. It's laying out pieces of paper that remain whole in themselves, or you can take an image and lay it out on smaller pieces of paper. Tiling, I guess you could call it.

FS: *Could you say something about your interest in moving from 2D to 3D and back again? It seems everything you do is related to that.*

TL: I've been making sculpture of various kinds for quite a long time. I've always drawn. Since I was about eighteen, I started making sculptures, mainly mathematical woodwork. I don't quite have access to a full woodwork workshop anymore, so I can't make as much woodwork as I used to. It's kind of an instance of being defined by what tools you have available to you. I don't have a wood shop, but I can do other stuff. I can still make various shapes, but mainly out of paper. Since I had been doing woodwork, I picked up photography. I made a ton of panoramic images. It's kind of fun to figure out how to project these images out of the computer into something that you can physically create, for instance a T-shirt or a ball, or other paper shapes.

FS: *Is there ever any work that stays in the computer, or does it always need to become physical?*

TL: Usually, for me, it is important to make something that I can actually physically interact with. The computer I usually find quite limiting. You can do amazing things with computers, you can pan around an image—that in itself is pretty amazing—but in the end I get more out of interacting with things physically than just in the computer.

FS: *But with Laidout, you have moved folding into the computer! Do you enjoy that kind of reverse transformation?*

TL: It is a challenge to do and I enjoy figuring out how to do that. In making computer tools, I always try to make something that I can not do nearly as quickly by hand. It's just much easier to do in a computer. Or in the case of spherical images, it's practically impossible to do it outside the computer. I could paint it with airbrushes and stuff like that but that in itself would take a hundred times longer than just pressing a couple of commands and having the computer do it all automatically.

FS: *Do you think that for you, the program itself is part of the work?*

TL: I think it's definitely part of the work. That's kind of the nuts and bolts that you have to enjoy to get somewhere else. But if I look back on it, I spend a huge amount of time just programming and not actually making the artwork itself. It's more just making the tools and all the programming for the tools. I think there's a lot of truth to that. When it comes time to actually make artwork, I do like to have the tool that's just right for the job, that works just the way that seems efficient.

FS: *How is the material imagined in the tool?*

TL: So, far not really completely. When you fold, you introduce slight twists and things like that. And that depends on the stiffness of the paper and the thickness of the paper. I've not adequately dealt with that so much. If you just have one fold, it's pretty easy to figure out what the creep is for that. You can do tests and you can actually measure it. That's pretty easy to compensate for. But if you have many more folds than that, it becomes much more difficult.

FS: *Are you thinking about how to do that?*

TL: I am.

FS: *That would be very interesting. It is very strange to imagine the material in the digital space, to give an idea of what might come out in the end. Then you really have to work your metaphors, I think. You talked about working with physical input, having touchpads... Can you talk a bit more about why you're interested in this?*

TL: You can do a lot of things with just a mouse and a keyboard. But it's still very limiting. You have to be sitting there, and you have to just control those two things. Here's your whole body, with which you can do amazing things, but you're restricted to just moving and clicking and you only have a single point up on the screen that you have to direct very specifically. It just seems very limiting. It's largely an unexplored field, just to accept a wider variety of inputs to control things. A lot of the multitouch stuff that's been done is just gestures for little tiny phones. It's mainly for browsing, not necessarily for actual work. That's something I would like to explore quite a lot more.

FS: *Your folding simulations made me think about gestures. Do you have any fantasies about how this could work for real?*

TL: There's tonnes of sci fi movies, like *Minority Report*, where you wear these gloves and you can do various things. Even that is still just mainly browsing. I saw one, it was a research project by this guy at Caltech. He had made this table and he wore polarized glasses so he could look down at this table and see a 3D image. And then he had gloves on, and he could sculpt things right in the air. The computer would keep track of where his hand is going. Instead of sculpting clay, you're sculpting this 3D mesh. That seemed quite impressive to me.

FS: *You're thinking about 3D printers, actually?*

TL: It's something that's on my mind. I just got something called the Eggbot. You can hold spheres in this thing and it's basically a plotter that can print on spherical surfaces or round surfaces. That's something I'd like to explore some more.

I've made various balls with just my photographic panoramas glued onto them. But that could be used to trace an outline for something and then you could go in with pens or paints and add more detail. If you're trying to paint on a sphere, just paint and no photograph, laying out an outline is perhaps the hardest part. If you simplify it, it becomes much easier to make actual images on spheres. That would be fun to explore.

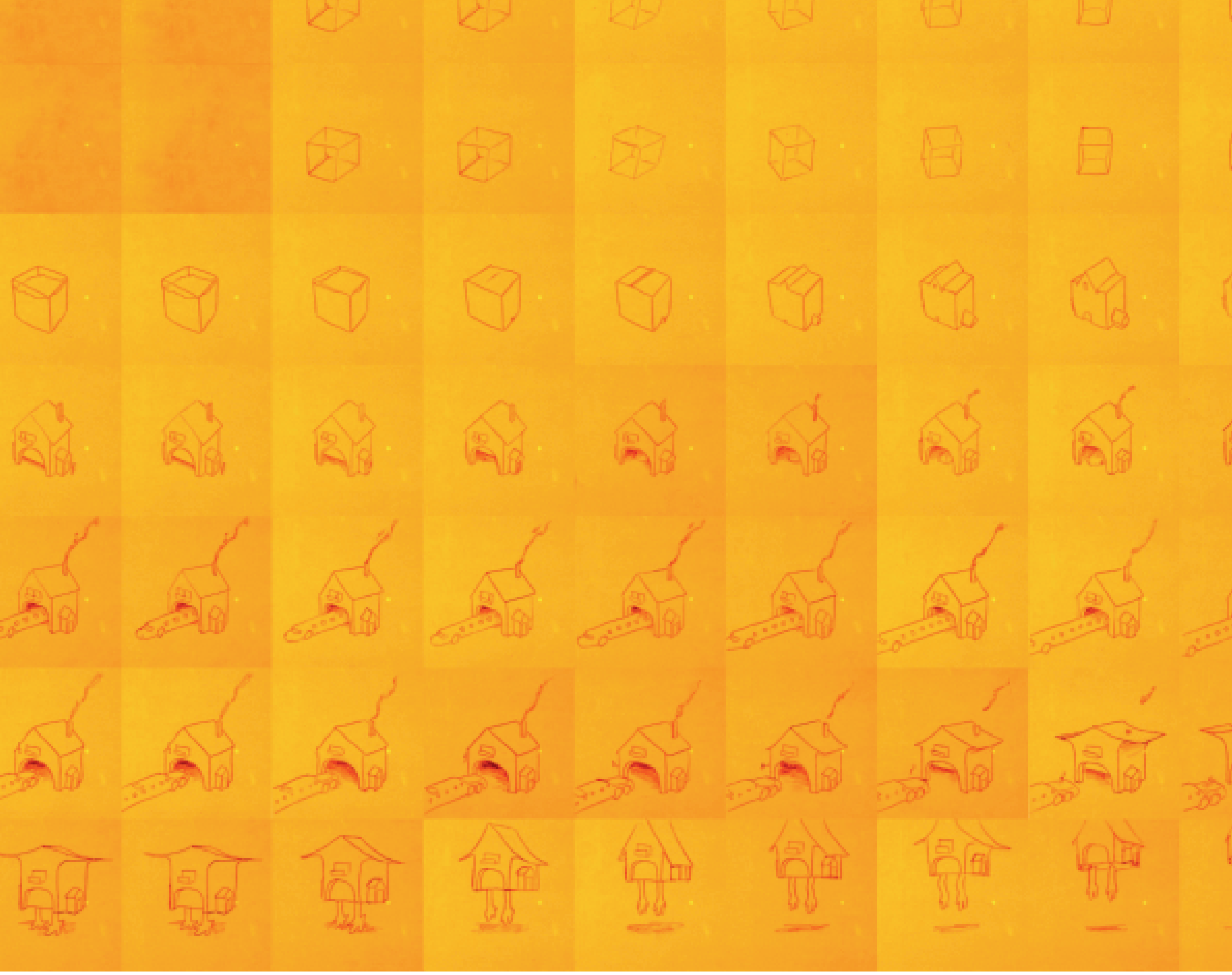
Pierre Marchand: *I'd like to come back to the folding. Following your existing aesthetic, the stiffness and the angles of the drawing would be very beautiful. Is it important you, preserving the aesthetic of your programs, the widgets, the lines, the arrows...?*

TL: I think the specific widgets, in the end, are not really important to me at all. It's more just producing an actual effect. So if there is some better way, more efficient way, more adaptable way to produce some effect, then it's better to just completely abandon what doesn't work and make something that's new, that actually does work. Especially with multitouch stuff, a lot of old widgets make no more sense. You have to deal with a lot of other kinds of things, so you need different controls.

PM: *The way someone sets up his workshop says a lot about his work. The way you made Laidout and how you set up its screen, it's important to define a spot in the space of the possible.*

Ludivine Loiseau: *What is nice is that you made the visualisation so important. The windows and the rest of the interface is really simple, the attention is really focused on what's happening. It is not like shiny windows with shadows everywhere, you feel like you are not bothered by the machine.*

FS: *Are there many other users of the program?*



TL: Not that I know of (laughter). I know that there is at least one other person who actually used it to produce a booklet. So I know that it is possible for someone other than myself to make things with it. I've gotten a couple of patches from people to not make it crash at various places but since Laidout is quite small, I can just not pay any attention to criticism. Partially because there isn't any, and I have particular motivations to make it work in a certain way and so it is easier to just go forward.

FS: *Can you describe again these two types of imposition, the first one being very familiar to us. What is the difference with the plan for a 3D object? A classic imposition plan is also somehow about turning a flat surface into a three dimensional object?*

TL: It is almost translatable. I'm reworking the 3D version to be able to incorporate the flat folding. It's not quite there yet, the problem is the connection between the pages. Currently, in the 3D version, you have a shape that has a definitive form and that controls how things bleed across the edges. When you have a piece of paper for a normal imposition, the pages that are next to each other in the physical form are not necessarily related to each other at all in the actual piece of paper. Right now, the piece of paper you use for the 3D model is very defined, there is no flexibility. Give me a few months!

FS: *So it is very different actually.*

TL: It is a different approach. One person wanted to do flexagons, it is sort of like origami I guess, but it is not quite as complicated. You take a piece of paper, cut out a square and another square, and then you can fold it and you end up with a square that is actually made up of four different sections. Then you can take the middle section, and you get another page and you can keep folding in strange ways and you get different pages. Now the question becomes: how do you define that page that is a collection of four different chunks of paper? I'm working on that!

FS: *We talk about the move from 2D to 3D as if these pages are empty. But you actually project images on them and I keep thinking about maps, transitional objects where physical space is projected on paper which then becomes a second real space and so on. Are you at all interested in maps?*

TL: A little bit. I don't really want to because it is such a well-explored field already. Already for many hundreds of years the problem is how do you represent a globe onto a more or less two-dimensional surface. You have to figure out a way to make globe gores or other ways to project it and then glue it on to a ball for example. There is a lot of work done with that particular sort of imagery.

PM: *Too many people in the field!*

TL: Yes. One thing that might be interesting to do though is when you have a ball that is a projection surface, then you can do more things, like overlays onto a map. If you want to simulate earthquakes for example.

FS: *And the panoramic images you make, do you use special equipment for this?*

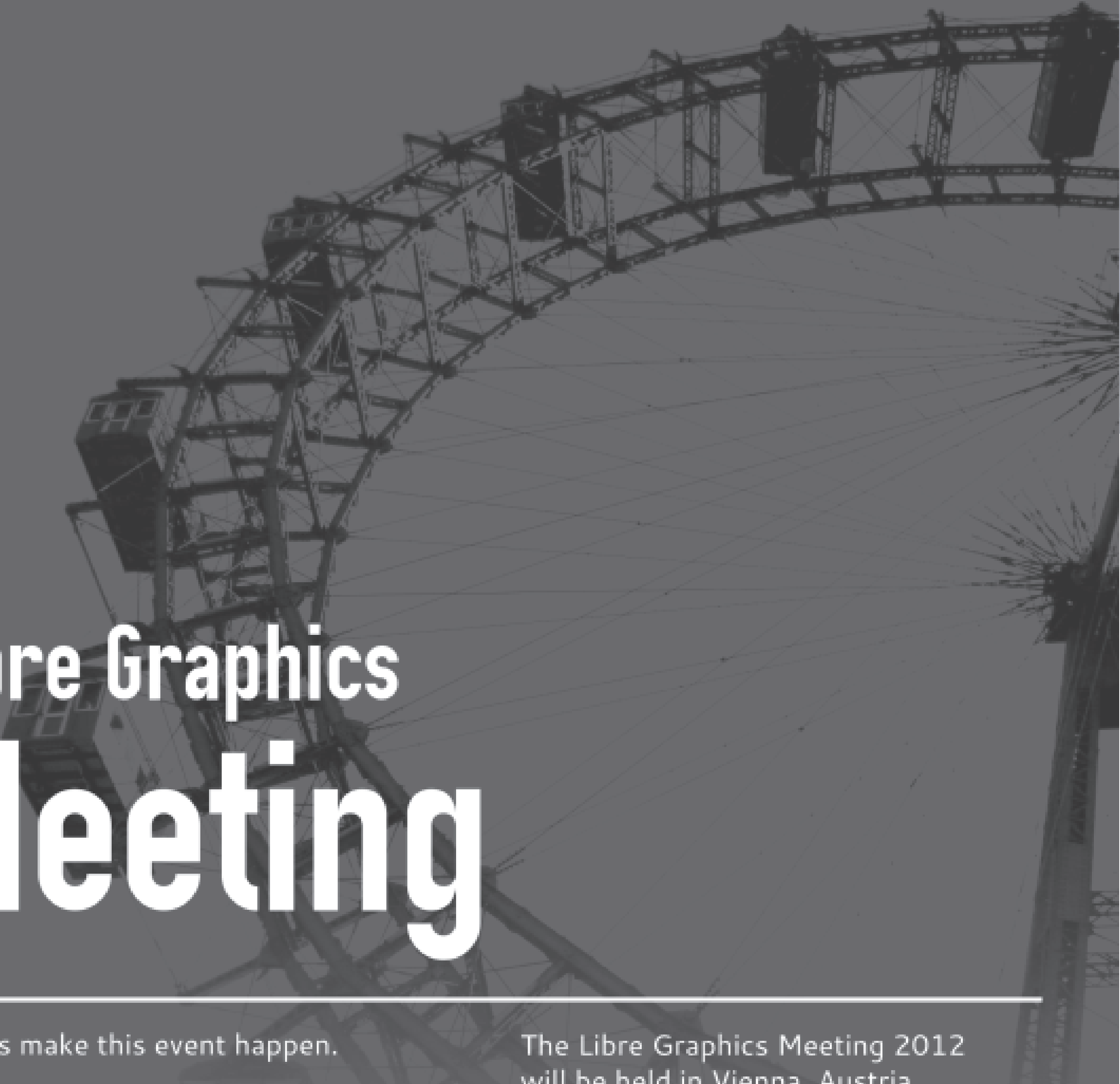
TL: For the first couple, I made this 30-sided polyhedron that you could mount a camera inside and it sat on a base in a particular way so you could get thirty chunks of images from a really cheap point-and-shoot camera. You do all that, and you have your thirty images and it is extremely laborious to take all these thirty images and line them up. That is why I made the 3D portion of Laidout, it was to help me do that in an easier fashion. Since then I've got a fisheye lens which simplifies things quite considerably. Instead of spending ten hours on something, I can do it in ten minutes. I can take 6 shots, and one shot up, one shot down. In Hugin you can stitch them all together.

FS: *Panoramic images are usually spherical or circular. Do you take certain images with a specific projection surface in mind?*

TL: To an extent. I take enough images. Once I have a whole bunch of images, the task is to select a particular image that goes with a particular space. Like cubes—there are few lines and it is convenient to line them up to an actual rectangular space like a room. The tetrahedron made out of cones, I made one of Mount St. Helena, because I thought it was an interesting way to put the two cones together. One thing I would like to do is to extend the panoramic image to be more like a progression. For most panoramic images, the focal point is a single point in space. But when you walk along a trail, you might have a series of photographs all along. I think it could be an interesting work to produce, some kind of ellipsoidal shape with a panoramic image that flows along the trail.

LIBRE GRAPHICS
MEETING 2012
VIENNA
2-5 MAY

Programmers of free and open source graphics software, artists and users from around the world are headed to Vienna for the 7th annual Libre Graphics Meeting



Libre Graphics
Meeting

Help us make this event happen.

Donate to support the LGM:
http://pledgie.com/campaigns/16614

Talk submissions until the 1st of March:
http://libregraphicsmeeting.org/2012/

The Libre Graphics Meeting 2012
will be held in Vienna, Austria
at the UAS Technikum Wien,
from Wed. 2nd to Sat. 5th of May 2012.

Contact: Ale Rimoldi ale@ideale.ch

ColorHug: filling the F/LOSS colour calibration void

Richard Hughes

WHAT IS COLORHUG?

The ColorHug is a small USB device that connects to a computer and measures colours very accurately. It allows animators, designers and artists to calibrate their screens. It builds a profile that lets colour-managed applications adapt colours automatically, helping videos and photos to look as originally intended. Measuring the screen with an external device is the only way to determine the difference between what the display thinks it's showing the user and what the user is actually being shown. For instance, most Thinkpad displays are very blue, which makes photos look overly cold. This isn't even something you can measure and fix once, as when the LCD display panel gets older, it gets dimmer and more yellow due to the backlight ageing. Even the phosphors in CRTs and plasmas degrade at different rates over time. The ColorHug device allows you to check your screen every few months or so, and takes a few minutes to profile and correct the screen profile. The device has drivers for the colord color management framework (also written by me) and will even be supported by argyllcms in the next release. The firmware interface is available for people to play with for no charge, and of course all the firmware, bootloader and hardware design is open source and freely available.

I developed the ColorHug to really scratch an itch. In doing this, I had no intention of becoming a hardware manufacturer. I enjoy writing Free software too much. The main motivation for designing and building this hardware was that at conferences I would showcase my software framework, and surprise designers with the effect of calibrating their screen. The main sticking point was that I was telling people to go and buy some expensive hardware from a proprietary color

company that doesn't care one little bit about Linux support. The proprietary hardware isn't really any good, and the profit margins must be huge on each device. So I began thinking. All I really needed to do was build a USB device that could count pulses really fast and do a few matrix calculations on a set of numbers. Luckily for the ColorHug, my university degree was in electronics and my previous job was working for a defence contractor prototyping new stuff. So really it wasn't that much work to build a functional prototype, and then build 50 units for people to play with.

I know the worldwide market of these devices is going to be small, but I'm hoping it'll kickstart other people in trying to do something like this themselves.

ColorHug is cheap. It's £48 plus postage, and so it's really aimed at the low end of the market. I could have built something much more accurate and a lot faster, but that would mean selling a device for several hundred pounds. I'm aiming this at people who want to “fix” the colour on their screen and who want to start playing around with colour managing their workflow. If you're already a colour geek, you really want to spend about £500 on a spectrophotometer device. If you're just a Linux user who likes to look at photos but don't know what “blood red” is in XYZ co-ordinates, the ColorHug might be the device for you.

THE BUSINESS OF COLORHUG

Setting up any business is difficult. Right from the start I decided to set up the business as an independent company, so my personal liability would be minimised. I injected a small amount of my own savings to be able to buy stock for 50 devices. The first 50 devices I soldered myself, not with machines, which turned out to be less fun than it sounds. The amount of profit for the first batch was tiny (less than £100) but the fact that I've got 50 users who say the device works means that I can be confident spending more money to make a larger, more profitable batch. The hardest part about actually making stuff in the UK is the sheer complexity of the tax situation and all the different registrations you have to get before you can even start. A lot of the companies I contacted to build the PCB wanted huge minimum orders, and without a few hundred thousand pounds in capital I'm forced to build up the order size organically using the profit from the last batch to buy the stock for the next, slightly larger, batch.

So, a lot of people want to know what the point is, when it all sounds like so much work for not a lot of cash: I want people to try making open hardware themselves. There are so many talented engineers in software that could make simple devices that outperform or undercut devices from proprietary companies. I've already got a fairly good-size community built around ColorHug, where people post patches for new functionality and fixes for bugs. The hard part for most people is releasing control and trusting that people will actually help the project grow.

ColorHug is only something I do late at night and at the weekends. This means the project is going to move more slowly than if I had a few full time staff. The short term prognosis is batch number 2, which is 100 units. Then a few months after that I'll probably double the batch size and build some more. I know the worldwide market of these devices is going to be small, but I'm hoping it'll kickstart other people in trying to do something like this themselves.

Lumiera is looking for developers.

Lumiera is an Open Source project to create a professional, non-linear video editor. The NLE is geared towards the rigours of a professional environment, but it should equally be at home to the casual, hobby user. Born from the the Cinelerra project, Lumiera strives to be innovative and offer a challenging and instructive ground for developers.

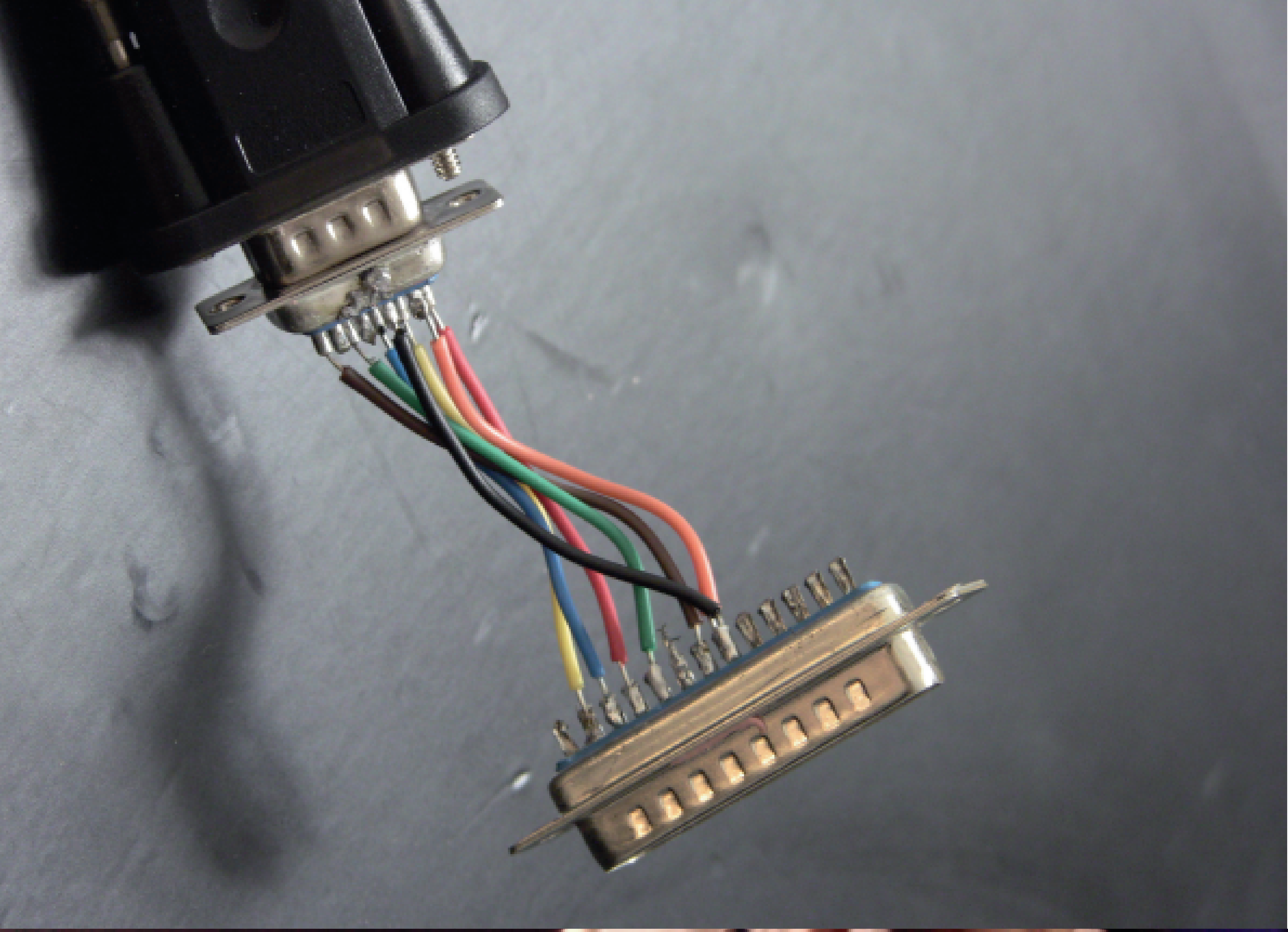
While the project has an active community and is steadily evolving, developers are few.

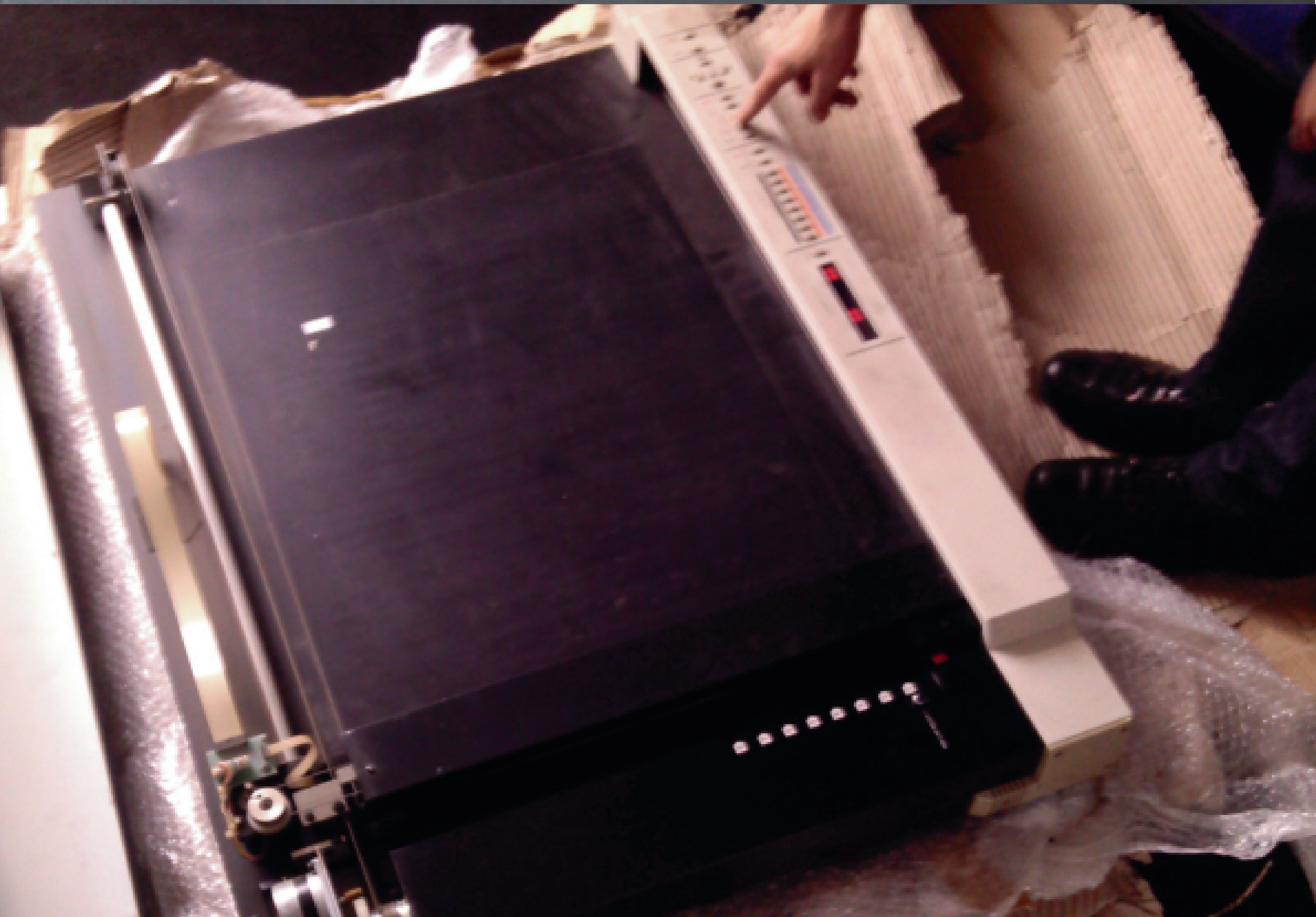
We are looking for developers. We believe that Lumiera can cater from the novice to the more experienced developers.

We'd be delighted to hear from you.

Lumiera.org

edit your freedom





Resurrecting the noble plotter

Manufactura Independente & Diogo Tudela

Pen plotters are already a somewhat distant artefact of early mass-production digital devices. Designed to meet the needs of technical draftspeople by tracing CAD and architectural schematics, they were made obsolete by large-format inkjet printers right before the turn of the millennium.

The appeal of a pen plotter was its precise ability to trace contours in a way that no other kind of printer could. Its transparent operation looks almost magical: after the document is sent to the device, a mechanical arm grabs one of the available pens and quickly traces a drawing, switching pens as it goes if the document requires more than one colour. Unfortunately for the plotter, breakthroughs in printing resolution made it possible to reproduce (and surpass) the quality of a plotter's accurate lines. Thus, for technical and professional purposes, pen plotters have become obsolete and now pop up regularly at bargain prices on auction sites, with the cost of shipping the bulky device sometimes more than that of the used pen plotter itself.

The fast-paced upgrade cycle of modern operating systems, with the inevitable consequence of dropping support for older devices, has resulted in the absence of any kind of recent drivers for most pen plotters in Windows and OS X. It has also become significantly harder to find software that can work with the plotters' page description language, HPGL.

As you would expect, given this dire scenario, F/LOSS comes in to save the day. Chiplotle is a small and powerful Python library which provides a command-line interface that allows sending HPGL files to a plotter. Even if it isn't supported directly by the library, you can probably get your plotter working using Chiplotle's generic profile.

What about HPGL? It is indeed hard to find support in modern software, but again, as expected, our favourite vector editor provides the answer. Inkscape is able to export HPGL files, right out of the box. And since HPGL files are plaintext, there's little to stop us from making our own scripts to fine-tune and correct the document before sending it for drawing. We've documented the process in some detail in our blog (http://tinyurl.com/plotterhacking).

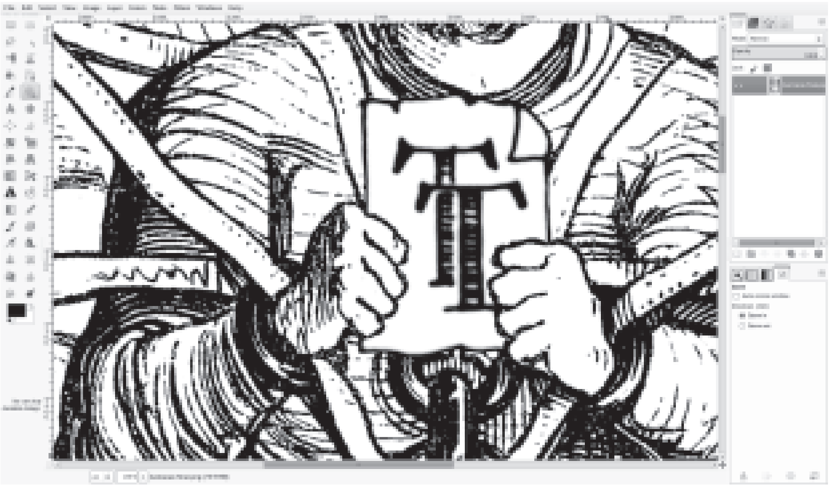
In exploring plotter hacking, we worked with Diogo Tudela, who acquired a used Roland DPX-2200 A2 plotter and wanted to hook it up. Using Chiplotle and hacking through the hardware adapters in order to get the right configuration, the three of us managed to resurrect this 1985 device into full operation. It was a thrill to be able to experiment with things that regular printers cannot do: we can fit almost any pen in its robot hand, as well as use fancier kinds of paper that would jam a regular printer. The process of hand-tweaking the HPGL files and making pre-processing scripts brought us a lot closer to the peculiarities of a plotter's operation principles, far from the usual black-box experience that regular inkjets give us, with their sealed print cartridges and hard-to-hack interfaces.

With F/LOSS developers providing the necessary tools, and with plotters available at bargain prices, there's no better time for hacking and exploration with the noble pen plotter.

Resource list 1.4

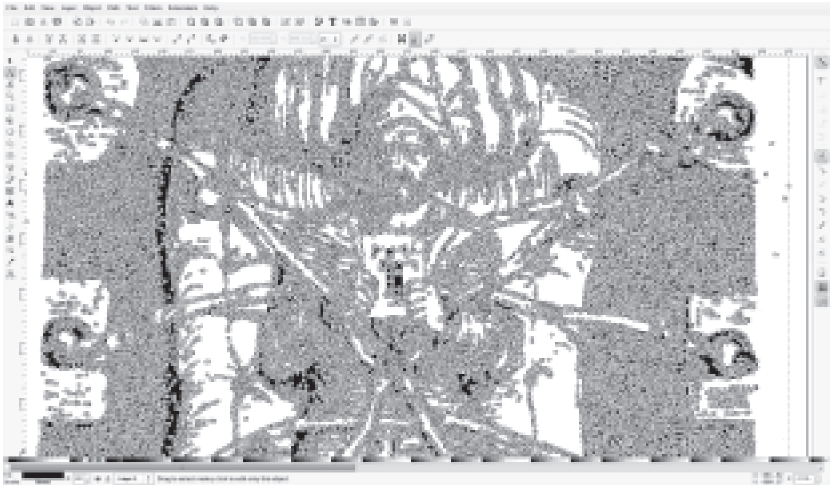
GIMP

A raster based image editor for GNU/Linux, Mac OS X and Microsoft Windows.



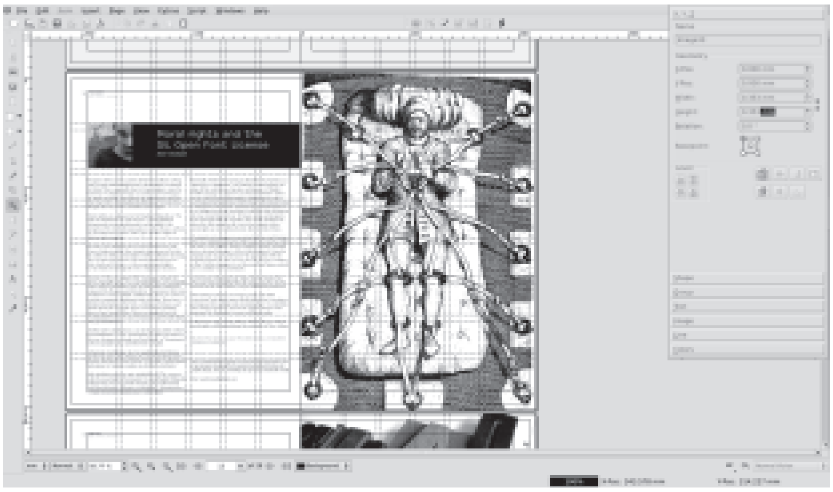
INKSCAPE

A vector graphics editor for GNU/Linux, Mac OS X and Microsoft Windows.



SCRIBUS

A desktop publishing program for GNU/Linux, Mac OS X and Microsoft Windows.



Glossary 1.4

Alchemy:
A sketching environment designed for visual experimentation. Available for GNU/Linux, Mac OS X and Microsoft Windows.

Aptana Studio:
An IDE for creating web applications. Available for GNU/Linux, Mac OS X and Microsoft Windows.

ArgyllCMS:
A colour management system with ICC profile compatibility for a variety of display and capture devices. Available for GNU/Linux, Mac OS X and Microsoft Windows.

Audacity:
A sound editing application for GNU/Linux, Mac OS X and Microsoft Windows.

Blender:
A powerful 3D animation application for GNU/Linux, Mac OS X and Microsoft Windows.

Chiplotle:
A Python API for pen plotters. Implements HPGL. Available as a Python library.

CMS:
Stands for Content Management System. Software installed on a webserver in order to provide a simple framework for editing webpages. WordPress and Drupal are examples of content management systems.

command line:
A text only user interface, allowing users to input commands without the intervention of cursors and graphical user interfaces.

colord:
A tool for installing, managing and creating colour profiles. For GNU/Linux.

Creative Commons:
A suite of licenses designed to allow creators and users of works flexibility beyond that offered in traditional copyright.

F/LOSS:
Stands for Free/Libre Open Source Software. Software which has a viewable, modifiable source. It can be modified and redistributed.

firmware:
Software designed to be permanently installed on a computer/device.

Free software:
A term describing software which is made available under licenses permitting users to not only run it, but to examine its code, redistribute it and modify it.

GIMP:
A raster based image editor for GNU/Linux, Mac OS X and Microsoft Windows. (Sometimes spelled Gimp)

Hugin:
Panoramic photo stitching software for GNU/Linux, Mac OS X and Microsoft Windows.

Inkscape:
A vector graphics editor for GNU/Linux, Mac OS X and Microsoft Windows.

JavaScript:
A scripting language commonly used on websites.

Krita:
A drawing application supporting both vector and raster images. Available for GNU/Linux, FreeBSD and Microsoft Windows.

Laidout:
A layout and desktop publishing application for GNU/Linux.

Luciole:
A stop motion animation program for GNU/Linux.

MyPaint:
A digital painting application with a focus on fluid workflow. Available for GNU/Linux, Mac OS X and Microsoft Windows.

Node.js:
A JavaScript-based system for creating internet applications.

open hardware:
Hardware which follows the same principles as F/LOSS, including publicly available, freely licensed schematics.

Paper.js:
A web-friendly JavaScript version of Scriptographer.

PCB:
A printed circuit board, the base of most electronics, generally consisting of a number of electronic components linked together to form certain types of circuits.

Pencil:
2D animation software with support for both vectors and rasters. Available for GNU/Linux, Mac OS X and Microsoft Windows.

Processing:
A programming language and development environment predominantly used for visually-oriented or media-rich projects. Available for GNU/Linux, Mac OS X and Microsoft Windows.

Python:
A popular interactive programming language. Available for GNU/Linux, Mac OS X and Microsoft Windows.

"scratching an itch":
A term used in many F/LOSS communities to mean that a developer has created a program primarily to meet his/her own needs, instead of the needs of others.

Scribus:
A desktop publishing application for GNU/Linux, Mac OS X and Windows.

Scriptographer:
A scripting plugin for Adobe Illustrator which gives the user the possibility to extend Illustrator's functionality by the use of the JavaScript language. www.scriptographer.org. Scriptographer is a GPL-licensed plugin for proprietary software.

SIL Open Font License (OFL):
A license intended for use with fonts and font related software. Dictates terms which allow modification and redistribution of fonts.

ToonLoop:
A stop motion animation application for GNU/Linux. A minimal version is available for Mac OS X and Microsoft Windows.

Zotero:
A reference management application with extensions for browsers and word processors. Available GNU/Linux, Mac OS X and Microsoft Windows.

For the love of letters!

Designers have a simple way to track progress and keep different versions of their work: "Save As." Developers use version control systems so their project files are kept in a less messy way. Their motivation is to work together, by working separately and sharing their progress.

Pablo Impallari is the type designer behind hits like Lobster, Kaushan Script and Cabin. Before teaching himself type design, Pablo worked as a web developer, so creating his typefaces in a public, participatory way seemed natural. Developers have a motto for this, "Release early, release often."

As Pablo began collaborating with other type designers on libre fonts, both those across his desk and across the world, he saw the need for a simple, designer-friendly version control system. To see Pablo's current projects, and find out how easy it is to join them, visit **impallari.com/projects**

Libre Graphics Magazine 1.4
March 2012
ISSN: 1925-1416