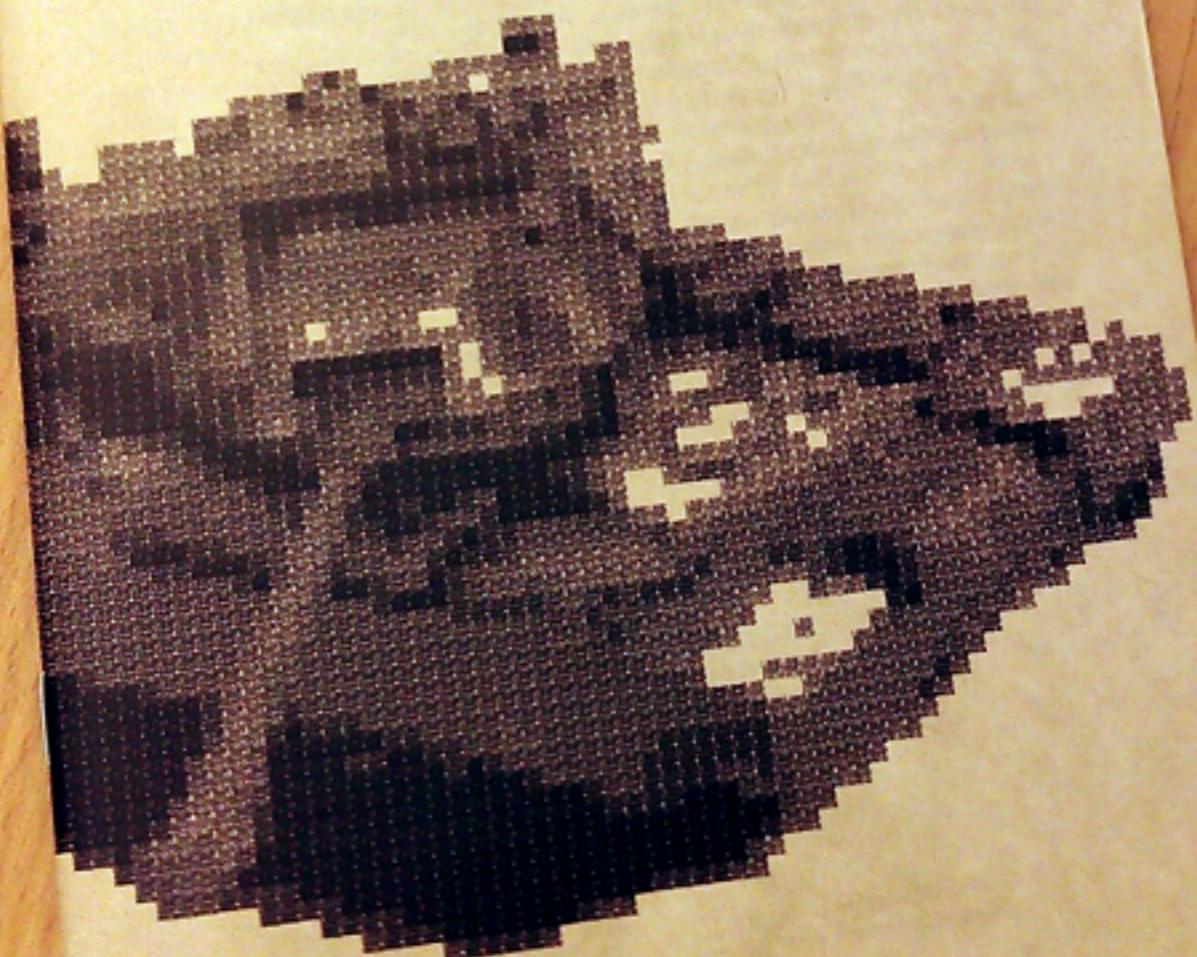
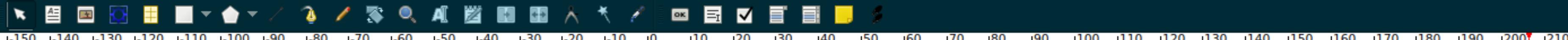


LIBRE
GRAP
HICS
#0!







C D E F G H I
L M N O P Q
T U V W X Y

! i c k
o w n f o x
m p s o v e r
e l a z y

Before webfonts

JULIEN DESWAFF

Fonts seem to have conquered the web. The @font-face CSS property is everywhere and its use has reached far beyond the display of characters.¹ But there was a time, not so long ago, when web designers had to struggle with complex processes just to display the title of a blog article in the typography of their choice.

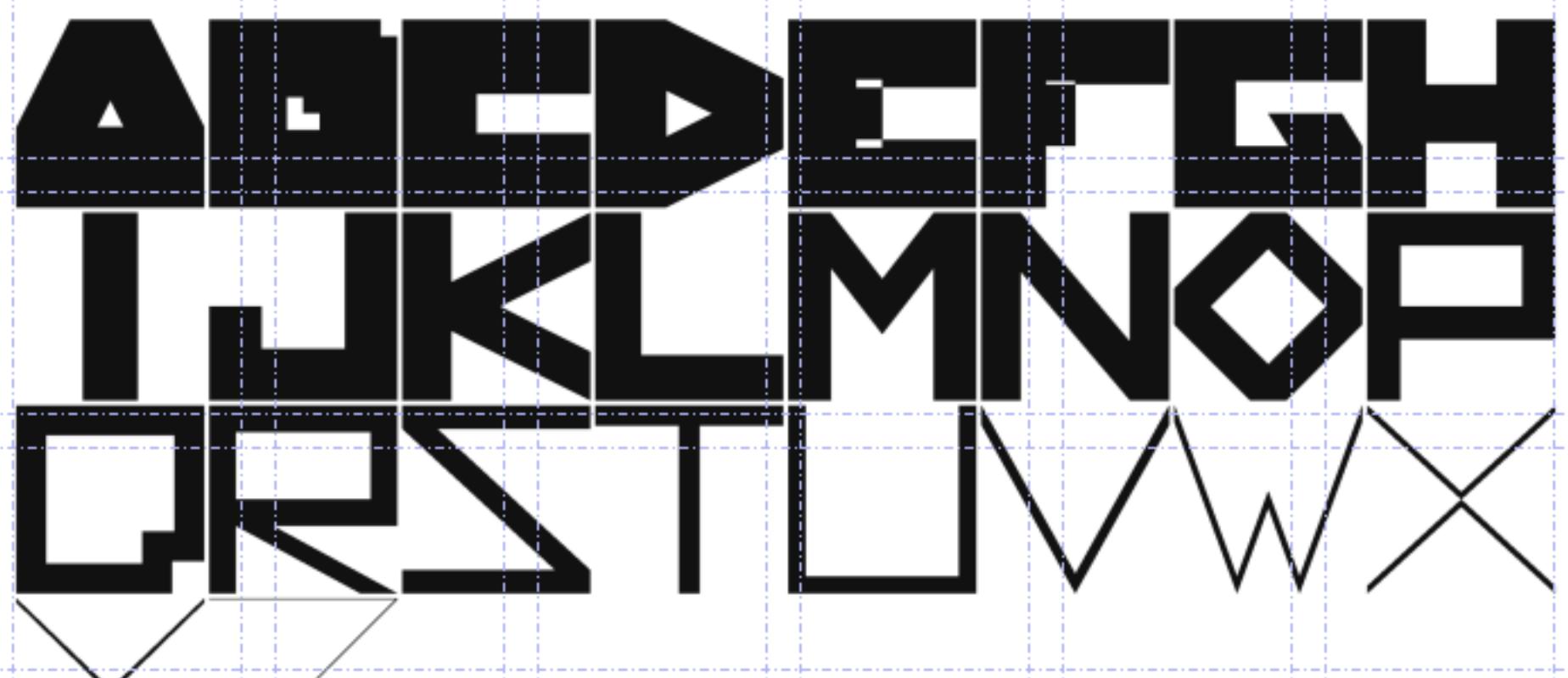
At its core, the web has always been text. But the glyphs to display it had to be local. When designers embraced the medium, their only options were the fonts most people had on their machines (Arial, Courier New, Times New Roman, Webdings...).² The obvious solution was to replace text with images, either pre-cooked by the designer himself, or baked on the fly and cached by the server. This technique, although widely used, was not elegant. It was heavy on the server and on the bandwidth, plus those little rendered texts weren't selectable or scalable.

To circumvent these flaws, an approach called sIFR³ was developed. It built a small Flash file into the page, used to load the desired font. Selected pieces of text would then dynamically be replaced by Flash-rendered text, all done in

the browser. Text could be scalable and selectable. But it required a proprietary plugin to be displayed correctly—and don't even dream about printing it.

When I started to design my web portfolio, none of these solutions really appealed to me. I wanted to stick with text-based open standards and no "image tricks."

The <canvas> tag was then only supported by half of the browser market. And I was starting to get interested in "everything parametric." The font⁴ I was working on used a single closed polygon for each of its glyphs. Each letter was just a set of coordinate relations that would allow me to change the weight and proportions at will. To display it in the browser, I found some obscure Javascript library created by Walter Zorn⁵ that exposed a set of vector drawing functions. The library worked by creating a coloured <div> for each pixel of the drawing using a fast algorithm to prevent too much repetition and to combine as many pixels possible into one <div>. It worked remarkably well across browsers and still works in recent ones because it just uses a basic element from HTML. The "pixelated" look of it also turned out to be very pleasing to me.



<PAGEOBJECT OwnPage="22" PTYPE="4" XPOS="737.801574802994" YPOS="9850.69135647754" = "641.795632618734" RADRECT="0" FRTYPE="0" CLIPEDIT="0" PWIDTH="1" PCOLOR="None" PCOLOR2="N EDLST="" SHADE="100" SHADE2="100" GRTYP="0" ROT="0" PLINEART="1" PLINEEND="0" PLINEJOIN="0" LX="0" LOCALY="0" PICART="1" PLTSHOW="0" BASEOF="0" textPathType="0" textPathFlipped="0" FL PE="1" RATIO="1" PRINTABLE="1" ANNOTATION="0" ANNAME="" TEXTFLOWMODE="0" TEXTFLOW="0" TEXTF = "0" EXTRA="0" TEXTRA="0" BEXTRA="0" REXTRA="0" FLOP="0" PFILE="" PFILE2="" PFILE3="" PRFIL DED="1" LOCK="0" LOCKR="0" REVERS="0" TransValue="0" TransValueS="0" TransBlend="0" TransBl e="0" LeftLine="0" RightLine="0" BottomLine="0" isGroupControl="0" NUMDASH="0" DASHS="" DAS 0 0 260.79 0 260.79 0 260.79 0 260.79 641.796 260.79 641.796 260.79 641.796 260.7 641.796 0 641.796 0 0 0 0 " NUMCO="16" COCOOR="0 0 0 0 260.79 0 260.79 0 260.79 0 260.79 0 260.79 641.796 260.79 641.796 0 641.796 0 641.796 0 641.796 0 641.796 0 641.796 0 0 0 0 " NUMGROUP="0 endArrowIndex="0" OnMasterPage="" ImageClip="" ImageRes="1" Pagenumber="0" isInline="0" fi s="1.02318153949454e-12" gYpos="49.9691517531301" gWidth="260.79023622063" gHeight="691.764 rier Spaced" LAYER="0" BOOKMARK="0" NEXTITEM="199" BACKITEM="-1">

<ITEXT CH="Design inspiration is often fuelled by constraints. In that spirit, I spent some time in trying to create a pixel font with greyscale squares for anti-aliasing. The initial idea was to create a font from a vector image in GIMP and a custom C program to turn the image into a UFO file where each set pixels were converted into a single character.">

<ITEXT CH=" For easier editing, the program was changed to output an XPM-inspired character mapping for different greyscale pixel values can be controlled."

<para PARENT="Text PropCourier Spaced"/>

<I>By coding/creating/designing a new set of components—puzzle pieces characters used to design the glyphs—new visual variations for the family can be created. Wh

<nbhyphen/>

<ITEXT CH="aliased drawing, known as pixeling, one imagines the curvature of the pixel's geometric area. The scope and power of expression "/>

<ITEXT CH="o

<підприємств>

<TEXT CH= high lower-case grid, thus the 3x3 set for drawing a lower-case 'o' >mac 2.3 also

```
def filter_itext(textbit):
    content = textbit["CH"]
    cparent = textbit.get("CPARENT")
    # parent = textbit.get("PARENT")
    if cparent:
        if cparent in IGNORED_STYLES:
            return ""
        elif cparent == H1_STYLE:
            return "# " + content
        elif cparent == H2_STYLE:
            return "## " + content
        elif cparent == SMCAPS_STYLE:
            return content.upper()
        elif cparent == BOLD_STYLE:
            return "**%s**" % content
        elif cparent == ITALIC_STYLE:
            return "_%s_" % content
    return content
```

```
def main(filename):
    output = ""
    sla_xml = open(filename, 'r').read()
    soup = bs4.BeautifulSoup(sla_xml, 'xml')
    for obj in soup.findAll("PAGEOBJECT"):
        if obj.findAll("ITEXT"):
            ...
            for item in obj.contents:
                if type(item) == bs4.Tag:
                    print item
```

Title: 0xA000 font family

Author: Øyvind Kolås

Section: Showcase

[](/images/2.3/at1.png)

Design inspiration is often fuelled by constraints. In that spirit, 0xA000 started out as a pixel font with greyscale squares for anti-aliasing. The initial ASCII set was made with a custom C program to turn the image into a UFO file where each set pixel referenced a component. The program was changed to output an XPM-inspired text file description of the font, where the current greyscale pixel values can be controlled.

By coding/creating/designing a new set of components—puzzle pieces corresponding to the different glyphs—new visual variations for the family can be created. When doing manual anti-aliasing, one imagines the curvature of the shape intersecting with the pixel's geometric area. The function of the font creation pipeline expands when adding puzzle pieces corresponding to these intersections to create semi-legible pixel fonts with a 3px-high lower-case grid, thus the 3x3 set for defining the starting point.

The small set of printable ASCII characters constrained how many puzzle pieces could comfortably fit.

Within the current constraints there's much room for refinement, and the overall project has many paths—including both serifs and improvements to the tooling for experimenting with the design.

<<http://pippin.gimp.org/0xA000>>

[^1]: Many font file formats support components, reusable vector shapes. These are normally defined as vector shapes, like the undecorated base glyphs "A," "E," and "0" for "A" and "E".

[](/images/2.3/at2.png)

[](/images/2.3/at3.png)

Title: 0xA000 font family

Author: Øyvind Kolås

Section: Showcase

Tags: type design, bitmap

[](/images/2.3/at1.png)

Design inspiration is often fuelled by constraints. In that spirit, 0xA000 started out as a pixel font with greyscale squares for anti-aliasing. The initial ASCII set was made with a custom C program to turn the image into a UFO file where each set pixel referenced a component. The program was changed to output an XPM-inspired text file description of the font, where the different greyscale pixel values can be controlled.

By coding/creating/designing a new set of components—puzzle pieces corresponding to the different glyphs—new visual variations for the family can be created. When doing manual anti-aliasing, one imagines the curvature of the shape intersecting with the pixel's geometric area. The font creation pipeline expands when adding puzzle pieces corresponding to these inclusions to create semi-legible pixel fonts with a 3px-high lower-case grid, thus the 3x3 set for defining a starting point.

The small set of printable ASCII characters constrained how many puzzle pieces could comfortably fit.

Within the current constraints there's much room for refinement, and the overall project has—~~including~~ both serifs and improvements to the tooling for experimenting with the design.

<<http://pippin.gimp.org/0xA000>>

[^1]: Many font file formats support components, reusable vector shapes. These are normally represented as vector shapes, like the undecorated base glyphs "A," "E," and "0" for Á¥Á¢Á¤ÁºÁ½Á£ and Á».

[](/images/2.3/at2.png)

Recipes for some of our steps

Extracting image paths from Scribus files

```
cat 01-original-scribus-files/lgmag-1.1-p* | \
grep -oe '"[^"]*\.(jpg|png)"' | \
sort | uniq | \
sed 's://"//g; s/\.\.\.\///g' > images-1.1.txt
```

Re-downloading images from repos

After extracting the image lists from the previous recipe, we did this (example for 2.4):

```
mkdir -p images/2.4
cd images/2.4
while read f; do wget "https://gitlab.com/libregraphicsmag/vol2issue4/raw/master/$f"; done < ../../image-lists/images-2.4.txt
```

Resizing images to a max width and height

We wanted max 1800px width and 1400px height, while keeping smaller images as they are.

```
cd images/2.4
for f in *; do echo $f; mogrify -resize 1800x1400> $f; done
```

or, to alter all images in subdirectories:

```
cd images
find . -name "*" | xargs mogrify -resize 970x4000>
```

Review text files with a particular string

We needed to check every instance of a lowercase acronym (e.g. "svg") and edit it manually if it needed to be made uppercase again.

```
grep '\bsvg\b' issue* -lr | xargs vim -p
```

Turn all images into image links

We wanted to change all images into a thumbnail which would then link to the original image.

So this

```

```

Quickstart

Installing Pelican

Writing content

Publish your site

Settings

Creating themes

Plugins

pelican-themes

Importing an existing site

Frequently Asked Questions (FAQ)

Tips

Contributing and feedback guidelines

Pelican internals

Some history about Pelican

Release history

Pelican 3.7.1

Pelican is a static site generator, written in [Python](#). Highlights include:

- Write your content directly with your editor of choice in [reStructuredText](#) or [Markdown](#) formats
- Includes a simple CLI tool to (re)generate your site
- Easy to interface with distributed version control systems and web hooks
- Completely static output is easy to host anywhere

Ready to get started? Check out the [Quickstart](#) guide.

Features

Pelican 3 currently supports:

- Articles (e.g., blog posts) and pages (e.g., "About", "Projects", "Contact")
- Comments, via an external service (Disqus). If you prefer to have more control over your comment data, self-hosted comments are another option. Check out the [Pelican Plugins](#) repository for more details.
- Theming support (themes are created using [Jinja2](#) templates)
- Publication of articles in multiple languages
- Atom/RSS feeds
- Code syntax highlighting
- Import from WordPress, Dotclear, or RSS feeds
- Integration with external tools: Twitter, Google Analytics, etc. (optional)
- Fast rebuild times thanks to content caching and selective output writing

Why the name "Pelican"?

"Pelican" is an anagram for *calepin*, which means "notebook" in French. ;)

Source code

master	archive / content / issue1.4 /	+ ▾	Lock	History	Find file	Web IDE	QP ▾
	Turn images into links	Ricardo Lafuente authored 3 minutes ago	1bb0f630				
Name	Last commit	Last update					
..							
 04-masthead.md	Add missing titles	6 days ago					
 06-editors-letter.md	Capitalise acronyms everywhere	6 days ago					
 07-production-colophon.md	Added photos to columns, editor's letter, prod coloph...	6 days ago					
 09-new-releases.md	Edit layout of best-of, new-releases and small&useful	6 days ago					
 10-upcoming-events.md	Capitalise acronyms everywhere	6 days ago					
 12-column-dave.md	Turn images into links	3 minutes ago					
 14-column-eric.md	Turn images into links	3 minutes ago					
 16-notebook.md	Turn images into links	3 minutes ago					
 19-small-useful.md	Turn images into links	3 minutes ago					
 20-openlabesev.md	Turn images into links	3 minutes ago					
 22-best-of-svg.md	Turn images into links	3 minutes ago					
 24-interview-natanael.md	Turn images into links	3 minutes ago					
 28-3dprint-thingiverse.md	Turn images into links	3 minutes ago					
 30-baltan-cutter.md	Turn images into links	3 minutes ago					
 34-interview-paperjs.md	Turn images into links	3 minutes ago					

BROWSE BY

ISSUE ▾

AUTHOR

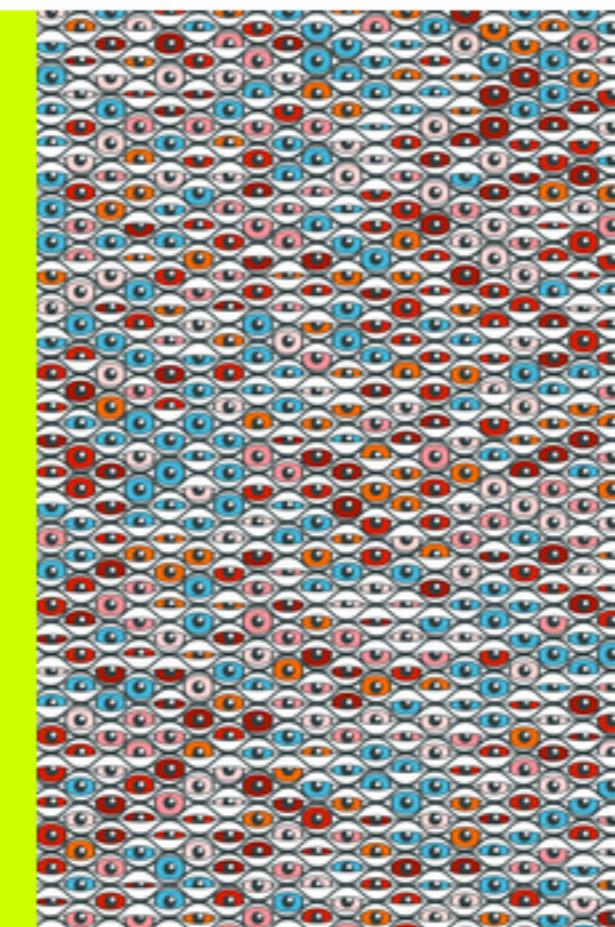
THEME



ISSUE 1.1
FIRST ENCOUNTERS—TAKING FLIGHT

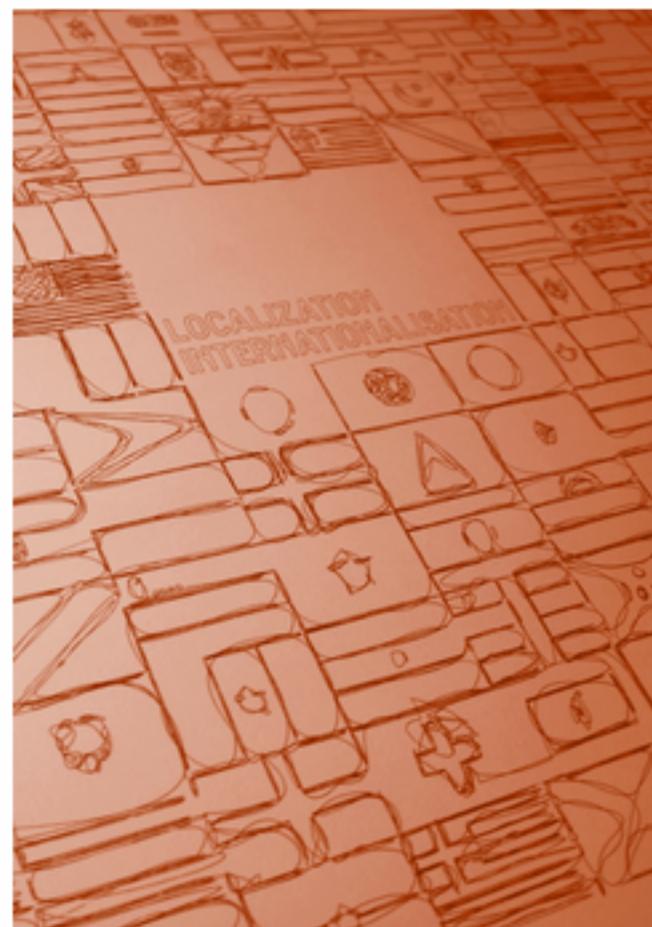


ISSUE 1.2
USE CASES AND AFFOR-
DANCES



ISSUE 1.3
COLLABORATION, COLLABO-
RATIVELY

The Physical,
the Digital
and the Designer



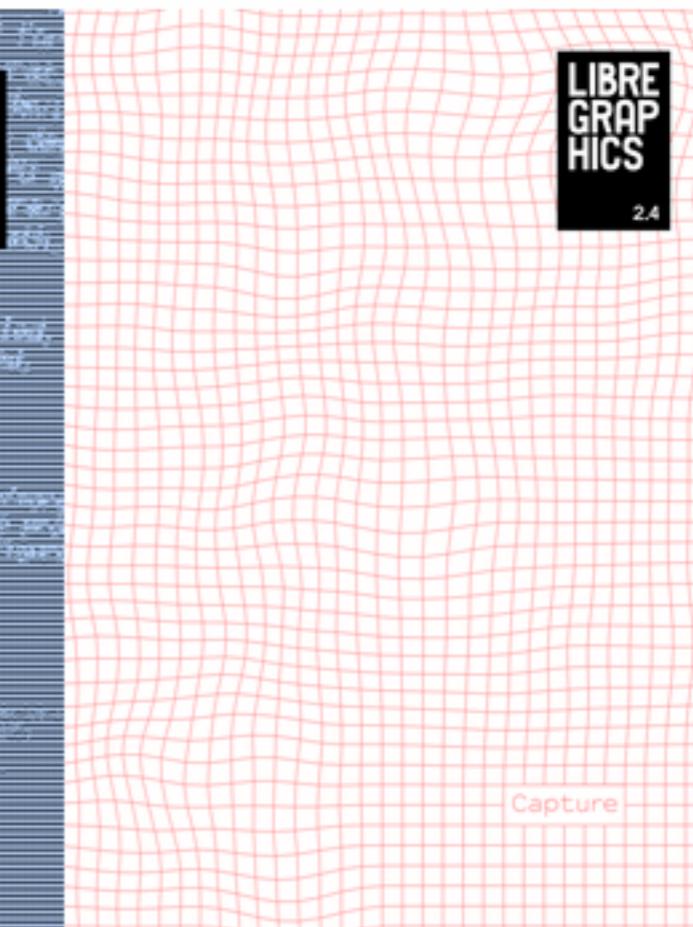
ISSUE 2.1
LOCALISATION/INTERNATION-
ALIZATION



ISSUE 2.2
GENDERING F/LOSS



ISSUE 2.3
THE TYPE ISSUE



ISSUE 2.4
CAPTURE

Issue 1.4 – The Physical, the Digital and the Designer

The Physical, the Digital and the Designer

[Source](#)[↓ PDF high-res](#)[↓ PDF low-res](#)

Masthead

[EDITOR'S LETTER](#)

— ginger coons

[PRODUCTION COLOPHON](#)

[Branching out: journey notes](#)

— Ana Isabel Carvalho, Ricardo Lafuente

[NEW RELEASES](#)

[New Releases](#)

Upcoming Events

[COLUMN](#)

[Moral rights and the SIL Open Font License](#)

— Dave Crossland

[COLUMN](#)

[Will these hands never be dirty?](#)

— Eric Schrijver

[NOTEBOOK](#)

[CulturaDigital.Br Festival](#)

[SMALL AND USEFUL](#)

[Small and Useful](#)

[DISPATCH](#)

[The finished and unfinished business of OpenLab ESEV](#)

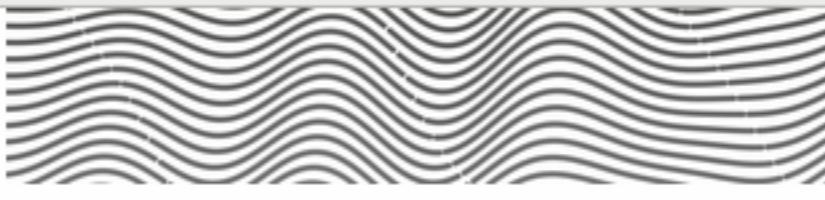
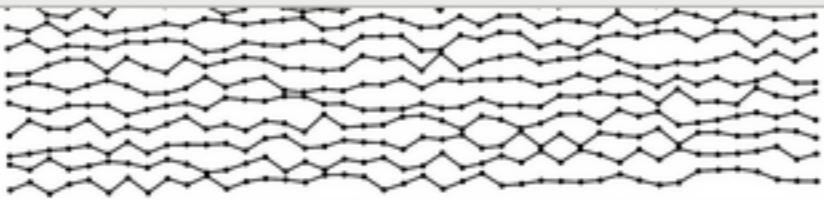
— Nelson Gonçalves, Maria Figueiredo

[BEST OF](#)

[Best of SVG](#)

[INTERVIEW](#)

[Natanael Gama talks techno-fonts and the benefits of Libre](#)



The impact of sketchPatch

Projects built on sketchPatch are fertile many times over, even before and after they exist. They are fertile ground for the people who play with them as users. They learn things, they have some fun. They don't necessarily come back, but a project like sketchPatch touches hundreds of people, even if for maybe only five minutes and to a limited degree. We do know that we've influenced a lot of people, even if only ever so slightly, and gave them a taste of first-hand creative coding. But that's just one level. Sophie and I learned a great deal from it, and met and dealt with dozens of great new people because of it. And the outcome is not just a website. It's an open source platform that keeps giving even in the form of entirely new incarnations, such as the LiveCodeLab project and workshops at MZTEK. It is an absurd amount of work and stress at times, but I confess, I still look at this IE6-compatible hobby horse from 2009, which is beginning to show its age with subtle and not so subtle cracks, and I think: I don't know how long this will be able to stand up for, but isn't this just the best site ever.

[« PREVIOUS ARTICLE: A JOURNEY THROUGH FORM FIELDS](#)

[NEXT ARTICLE: GENDERING F/LOSS »](#)

[Read in the original PDF ↗](#)

[Get a printed copy ↗](#)

[Source Markdown file ↗](#)

OTHER ARTICLES IN "DISPATCH"

[F/LOSS in the classroom](#)

— Ludivine Loiseau

[Coding pictures](#)

— Ricardo Lafuente

[The finished and unfinished business of OpenLab ESEV](#)

— Nelson Gonçalves, Maria Figueiredo

[Mothership HackerMoms: Forging Open Source community for](#)

ISSUE 2.1

[COLOPHON](#) [Colophon](#)

[EDITOR'S LETTER](#) [Localisation and internationalization](#)

[PRODUCTION COLOPHON](#) [Pruning branches](#)

[NOTEBOOK](#) [Mozilla Festival](#)

[NEW RELEASES](#) [New releases](#)

[COLUMN](#) [The transnational glitch](#)

[COLUMN](#) [A journey through form fields](#)

[DISPATCH](#) [Sketching creative code](#)

[CALL FOR SUBMISSIONS](#) [Gendering F/LOSS](#)

[SMALL & USEFUL](#) [Small & useful](#)

[COLUMN](#) [Type design, around the world](#)

Authors

ALEXANDRE LERAY (1)

SHOWCASE <stdin>

ALEXEI VANYASHIN (1)

DISPATCH Building Cyrillic fonts together

ALLISON MOORE (1)

SHOWCASE Papercut

ANA ISABEL CARVALHO (12)

PRODUCTION COLOPHON Production Colophon

PRODUCTION COLOPHON Versions under control

INTERVIEW Interview with Oxygen's Nuno Pinheiro

PRODUCTION COLOPHON Branching out: journey notes

INTERVIEW Paper.js: building, designing and the browser. An interview with Jonathan Puckey

FEATURE Resurrecting the noble plotter

PRODUCTION COLOPHON Pruning branches

PRODUCTION COLOPHON Making the switch

DISPATCH Forging foundries

INTERVIEW The League of Movable Type – an interview

PRODUCTION COLOPHON Dry Layout

No title yet

ANNA CARRERAS (1)

INTERVIEW Open Color 3D Scan

ANNEMIEKE VAN DER HOEK (1)

LAF KON (1)

SHOWCASE Bash scripts for generative posters

LAURA C. HEWITT (1)

SHOWCASE Camouflage and Mimicry Illuminated Drawing / Warfly 1

LILA PAGOLA (1)

FEATURE Applying F/LOSS as a final user and not dying in the attempt

LORAINE FURTER (1)

FEATURE SPECIMEN

LOREDANA BONTEMPI (1)

GUEST EDITOR'S LETTER Look at my camel. My camel is amazing.

LUDIVINE LOISEAU (2)

DISPATCH F/LOSS in the classroom

INTERVIEW Folds, impositions and gores: an interview with Tom Lechner

MAKEMAKE (1)

SHOWCASE Open source recipe for organic logos

MANUEL SCHMALSTIEG / GREYSCALE PRESS (1)

GUEST EDITOR'S LETTER Type etc.

MAR CANET (1)

SHOWCASE Knitic

MARIA FIGUEIREDO (1)

DISPATCH The finished and unfinished business of OpenLab ESEV

Themes

BITMAP (1)

SHOWCASE [0xA000 font family](#)

BITMAPS (1)

COLUMN [The heritage of our pixels](#)

COMMAND LINE (2)

FIRST TIME [Setting a book with Scribus](#)

COLUMN [The voice of the shell—in collaboration with my computer](#)

COMMONS (2)

BEST OF [Wayfinding and warnings from Wikimedia Commons](#)

BEST OF [Isn't Open Clip Art Library handy?](#)

COMMUNITY (1)

FEATURE [Managing artist communities: the case for Ubuntu Artists](#)

COPyleft (1)

COLUMN [Copyleft Business](#)

CUSTOM INTERFACES (1)

FEATURE [AdaptableGIMP: user interfaces for users](#)

CUSTOM WORKFLOWS (1)

FEATURE [Making your workflow work for you](#)

DEJAVU FONT (1)

FEATURE [Interview with Ben Laenen of DejaVu](#)

EVENTS (3)

LIBRE CULTURE (3)

NOTEBOOK [Make Art 2010](#)

COLUMN [This is the first day of my life](#)

FEATURE [Applying F/LOSS as a final user and not dying in the attempt](#)

LIBRE GRAPHICS (1)

NOTEBOOK [Libre Graphics Meeting 2011](#)

LINES (1)

COLUMN [The heritage of our pixels](#)

PROPCOURIER (1)

PRODUCTION COLOPHON [Versions under control](#)

RECIPES (1)

SHOWCASE [Open source recipe for organic logos](#)

SCHOOL (1)

INTERVIEW [Parallel School—an interview](#)

SCRIPTING (1)

SHOWCASE [Bash scripts for generative posters](#)

SCRIPTING IMAGES (2)

DISPATCH [Coding pictures](#)

PRODUCTION COLOPHON [Cover process](#)

SVG (2)

BEST OF [Wayfinding and warnings from Wikimedia Commons](#)

LIBRE

GRAP

HICS

<http://archive.libregraphicsmag.com/>