LIBRE GRAPHICS

2.3

Index

Guesting/journeying

GINGER COONS

ginger coons is a member of the Libre Graphics Editorial team.

This issue marks the second time we've invited a guest editor to take over the pages of *Libre Graphics* magazine. The first time, issue 1.3, which we called "Collaboration, collaboratively," brought to light artists and designers whose work we hadn't seen before, which we encountered through the eyes of our guest editors, Loredana Bontempi, Emanuele Bonetti, Morgan Fortems, and Thibaut Hofer. That was an important moment for us. It's exceptionally comfortable to travel in our own circles, and is made so much easier by having a close-knit editorial team. In considering work to include in each issue, we often find ourselves trying to break away from a reliance on the usual suspects. Our first guest editors took on that task admirably, and delivered an issue that surprised and delighted us with its variety.

That's why we've decided that, once per volume, on the third issue, we want to keep up the tradition of getting a fresh set of eyes. In this issue, 2.3, those eyes belong to Manuel Schmalstieg of Greyscale Press. This time around, Manuel brings us an inventory of new ways of thinking about type design, with a couple side excursions into publishing.

Undeniably, the history of digital type design and typography has been a trip. Moving from the high-fidelity glyphs of pre-digital type, to the aggressively pixelated letters of early computation, we now see both a glut and a renaissance in fonts for the web and other digital applications. The projects covered in this issue pick up on that trip. From generative and parametric work to approaches for curating and classifying type, the work represented in this issue presents a window into the current state of F/LOSS type design and typography. From the taxonomies and ways of organizing embodied in some of the projects, to new ways of devising and designing type, this issue also goes beyond just looking at what's happening now, and imagines some of the futures of F/LOSS type design and typography. We invite you to explore the works and ideas collected by our guest editor, and to come along as we take a look at the trip Libre type is taking.

Type etc.

MANUEL SCHMALSTIEG/GREYSCALE PRESS

In this issue, I've tried to collect a range of voices from the Libre type community, with their various approaches as designers, curators, afficianados and hackers. Libre type design is evolving quickly, and I hope that the new and upcoming generation of type tools—generative, parametric, running in your browser and elsewhere—will empower many more (non-)designers to get their hands dirty.

Working with Libre and freely licenced type designs opens up many doors for original, enthusiastic and imaginative (re)/(ab)use. I remember the moment when I was told how NotCourier, a derivative of Courier, was "forked" in a couple of hours. Ludivine Loiseau, a member of OSP, the forking foundry, said "we simply cut off the serifs, and there was that completely new font."

A similar revelation occurred to me while I was attending a FontForge workshop, in a lovely village in the south of France, and realized that with just a couple of commands, I could produce the missing medium weight for this magazine's title font, PropCourier Sans.

Indeed, fonts—sorry, typefaces!—are an odd thing: on one hand, they can take years, lifetimes, to be refined and polished into perfection (fortunately, the Libre software community has come up with tools such as ttfautohint and Kernagic, that alleviate the most time-consuming parts). On the other, when granted the freedom to transform and to adapt, radically original work can emerge from old designs in a matter of hours, minutes, split-seconds.

While software has a relatively short history, with current systems and languages reaching back to the 1970s (the origins of UNIX), type design is a much older field, as the evolution of typography stretches over a few millennia. Today's modern typefaces (such as Linux Biolinum, used in this magazine) are based on centuries-old rules and canons. Breaking from those traditions means challenging our perception of legibility (the 0xA000 font showcased in this issue is a perfect example). To consider our work to be embedded in such a slow historical process is a refreshing idea, especially on the backdrop of a speed-obsessed digital culture favouring short release cycles, coding sprints, and technological obsolescence.

But even if our fonts are following old rules, the tools that we use for designing type are evolving, in rather amazing ways. Metapolator is the perfect example of a type design tool that opens up uncharted perspectives, while at the same time incorporating the concepts of Metafont, a command-line font software invented by Donald Knuth 40 years ago.

Julien Deswaef's contribution to this issue also tackles this paradoxical relation to time: on-screen type rendering technologies and hacks that were considered cutting-edge 5 years ago are already reminding us of artefacts of a forgotten dark age.

In that context, I see this issue as a snapshot, a moment in the fast and slow evolutions of type design and tools, that will be probably superseded in the forthcoming months by new breakthroughs and discoveries.

So please take your time, and enjoy this issue of *Libre Graphics* magazine.

Typographic Vagrancy

NOTEBOOK

Where
Paris

What
A workshop in two parts, and in many senses of the word: a session introducing Libre type design tools, followed by a weekend-long marathon applying those new-found skills to develop Libre typography.

Most vernacular:
The photo-walk that inspired the type created during the event—getting out, looking at signage around the venue, and taking cues from the locale.

Best engagement:
The outcome of the workshop was a Libre type pack for local use. Neighbours, associations and businesses in the surrounding area are now free to use their own, hyper-localized type.

Best use of walls:
Photos of the neighbourhood mingled with in-progress type in a glorious mess of glyphs, presiding over the event from a back wall in the communal work room.

http://velvetyne.fr/marathongeneral

Vagabondage Typographique Festif

Conférence et formation aux outils de design typographique libres

Marathon de création typographique sur 2 jours

25, 26, 27 juillet

La Générale
14, av. Parmentier
Paris XI° Ⓜ Voltaire

Organisé par la fonderie Velvetyne.fr

LA GÉNÉRALE
www.lagenerale.fr

http://velvetyne.fr/marathongeneral

Design : Raphaël Bastide

BEYOND THE FIRST DECADE

LIBRE GRAPHICS
MEETING 2015
TORONTO
29 APRIL – 2 MAY
WWW.LIBREGRAPHICSMEETING.ORG

New Releases

Beautiful Open

A gallery of gorgeous F/LOSS project websites, just to show it can be done.

http://beautifulopen.com

Brick

A collection of high-quality web fonts, for your hyperlinking pleasure.

http://brick.im

Birdfont

The up and coming FontForge alternative is about to release its 2.0 version, with a beta package already available for those too eager to wait.

http://birdfont.org/betaversion.php

copyleft.org

Describing itself as "a collaborative project to create and disseminate useful information, tutorial material, and new policy ideas," copyleft.org aggregates existing and new resources on copyleft licensing.

http://copyleft.org

Flat/Even

A library for creating PDF layouts using Python code. It can be used as a code-based tool, or along its GUI companion, Even.

http://xxyxyz.org/flat

http://xxyxyz.org/even

Glyphr Studio

A web-based font editor with a modern interface for rapid tweaking, with exciting features like dual-screen editing and SVG import.

http://glyphrstudio.com

Opentype.js

Parses fonts and glyphs, going so far as to show contour parameters. From displaying fonts and their component parts to inspecting metadata, Opentype.js offers a compelling solution for online font display and inspection.

http://nodebox.github.io/opentype.js

Plain Pattern

Harnesses the awesome power of SVG to generate tiles for seamless patterns.

http://www.kennethcachia.com/plain-pattern

Prototypo

A parameter-based font generator, currently under development. While the hosted version is, at present, only accessible to those who supported the crowdfunding campaign, source code is available.

http://www.prototypo.io

Scri.ch

An elegantly minimalistic online sketch pad. Scri.ch presents you with a totally blank page, waiting for a drawing, and allows you to save your work for later reference and distribution.

http://scri.ch

The Missing Scarf

Not strictly a new release, because it appeared on the film festival circuit in 2013, but the award-winning animated short, made with Blender, has recently been made available for all to watch.

http://themissingscarf.com

http://vimeo.com/107395294

ttfautohint 1.2

The latest version of ttfautohint offers minor bug fixes and some emergent features, which promise to be elaborated on further in future releases.

http://www.freetype.org/ttfautohint

Moving kinetic typography

ANTONIO ROBERTS

Antonio Roberts is a digital visual artist
based in Birmingham, UK.
He produces artwork that takes
inspiration from glitches in software and
hardware, errors and the unexpected.
http://hellocatfood.com

The title sequence for the 1959 Alfred Hitchcock film *North By Northwest*, made by Saul Bass, is credited as the first extensive use of kinetic typography in film. As we move through time there begin to be more examples of its use, such as the opening sequence for *Star Wars*. Although essentially scrolling text, by animating the text to disappear into a vanishing point it reemphasizes the futuristic, space-age nature of the film.

With the widespread availability of digital animation and video editing tools there are now more artists and filmmakers using kinetic typography in their work. Despite this, few tools exist specifically to create kinetic typography, and those that do have not yet matured. Quite often, those interested in kinetic typography are forced to circumnavigate and hack together programs to trick the software into doing something it wasn't actually designed for. These approaches usually fail because the software has limited capabilities to edit the many detailed typographic properties that may need to be manipulated.

After Effects is the most widely used and documented tool designers use to create kinetic typography. A search for the term "Kinetic Typography Software" usually brings up tutorials and templates available for use with After Effects. The problem with this approach is twofold—not only is After Effects proprietary, but the use of templates, whether in proprietary or open source software, results in predictable and similar results.

Outside of proprietary software a number of experimental approaches exist. Blender, kdenlive, Synfig and other video editing and animation software can be used, but these still face the same specialized problems with handling text. For those willing to delve into programming more, there are some options. Processing, especially when combined with external libraries such as Geomerative, provides many opportunities for experimenting with animated and kinetic typography. In 2002, Johnny C. Lee published *Kinetic Typography Engine*, written in Java, which goes to great lengths to provide a taxonomy for creating kinetic typography. Once the user has navigated the complexities of the program they will find that it provides opportunities to animate many attributes of text.

The problem with these programmatic solutions is that they can't easily be exported or reused outside of their applications. And that's on top of being less accessible to

those without programming knowledge. This clearly points
to a need for further work in F/LOSS-based kinetic
typography, as well as a taxonomy for describing
the changes to the appearance and movement of text over
time that is software-independent and an open standard
to support implementation.

When compared to animation, video editing,
and filmmaking, kinetic typography is very new
and, as such, a common language and set of standards have
not yet been developed. Improving the medium's
accessibility by developing new tools for doing kinetic
typography will go some way towards maturing the
standards, style and voice of moving type.
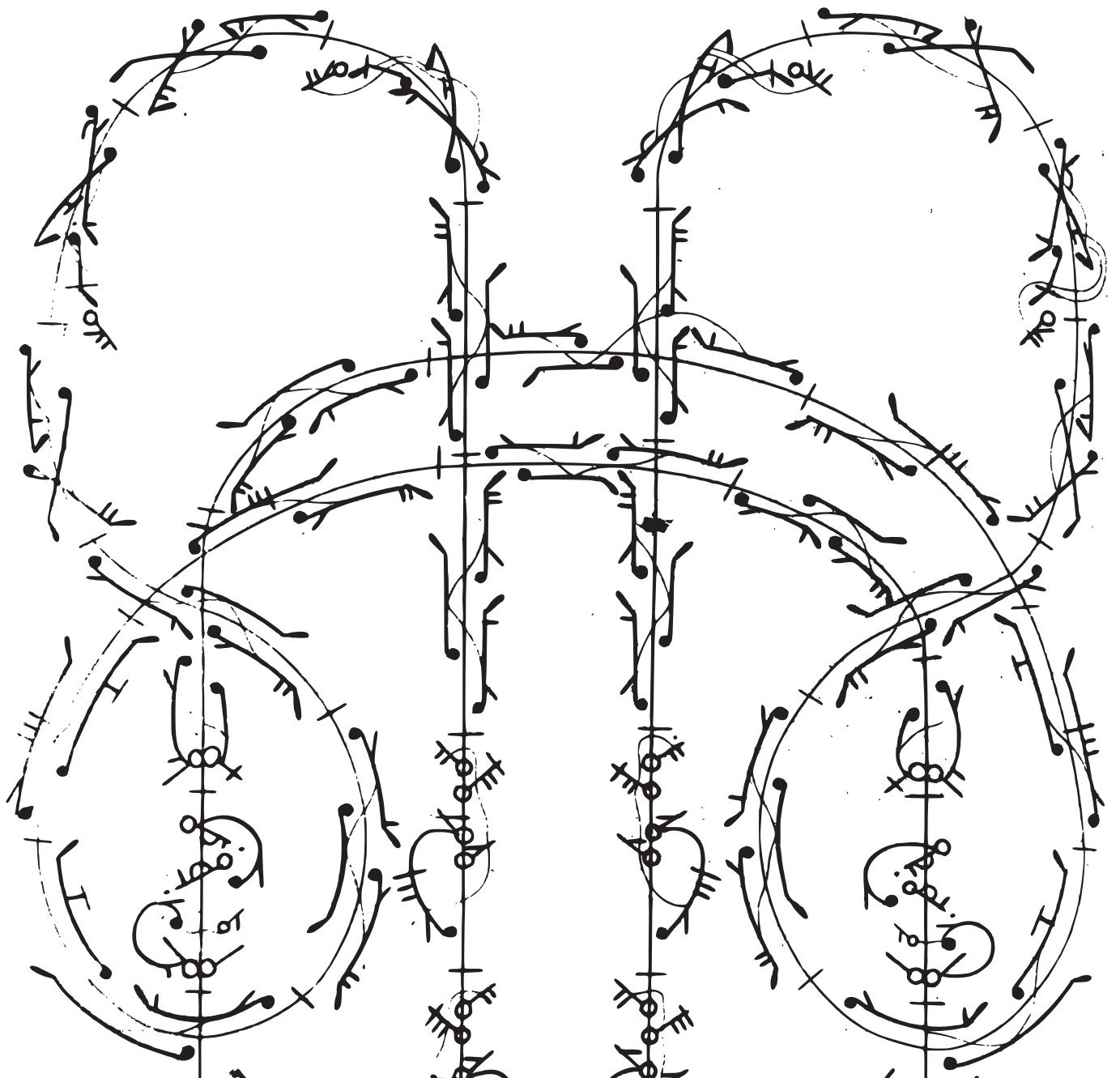
Friendlier forks

ERIC SCHRIJVER

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

The metaphors we live by are different in each era, and tell us about the social movements shaping the moment. In the time of Snowden and Zuckerberg, we get our metaphors from Silicon Valley. The "fork" is one such concept, originating in this case from the world of Open Source software development.

Initially considered a negative occurrence, a fork is when someone creates a new version of an existing project, taking a different direction from the original. A new style of Open Source collaboration, embodied in the popular code sharing platform Github (and Git, the system on which it is based), encourages forking. On Github, collaboration starts by creating a fork of a project and adding changes to this fork. Then one either contributes these changes back to the original repo (if the maintainer accepts it), or one goes one's own way—a fork in the traditional sense.

In the spirit of the fork, there is a lot to be said about how a pragmatic view on originality and authorship and an embracing of redundancy can make for a culturally rich ecosystem. These ideas are inspiring enough to see how they could also work outside the realm of software development. But when it comes to type design, one need not look to software development. To see how building upon existing creations makes typographic sense, one can look at type design history itself.

One of the seminal typefaces of post-war graphic design is described by its own creator as an improvement upon an existing font. Gill Sans is considered Britain's national sans serif, as seen on Penguin books and in the BBC logo. The typeface was designed by Eric Gill, who described it as an attempt to improve Edward Johnston's typeface made for the London Underground. Though one could go to great lengths enumerating the differences between the two typefaces, it is the idea that Gill Sans represents a proposition of improvement over an existing typeface that interests me.

In software, it can be confusing to have many forks of one project existing, because it is probably practical to use just one version of the package. Cultural artefacts like typefaces, however, can more easily exist in an abundance of similar guises, because they coexist. Both the inspiration and the inspired can be used by a contemporary designer. Gill Sans can exist comfortably alongside Johnston's typeface, without diminishing from its functionality. The fork is an addition, rather than a diminishment.

As more and more typefaces are becoming available under various open licenses, a type design culture of the digital fork becomes more and more feasible. Yet for this to happen, type designers might first need to give themselves permission to modify, to edit instead of re-making, and to tweak instead of reviving.
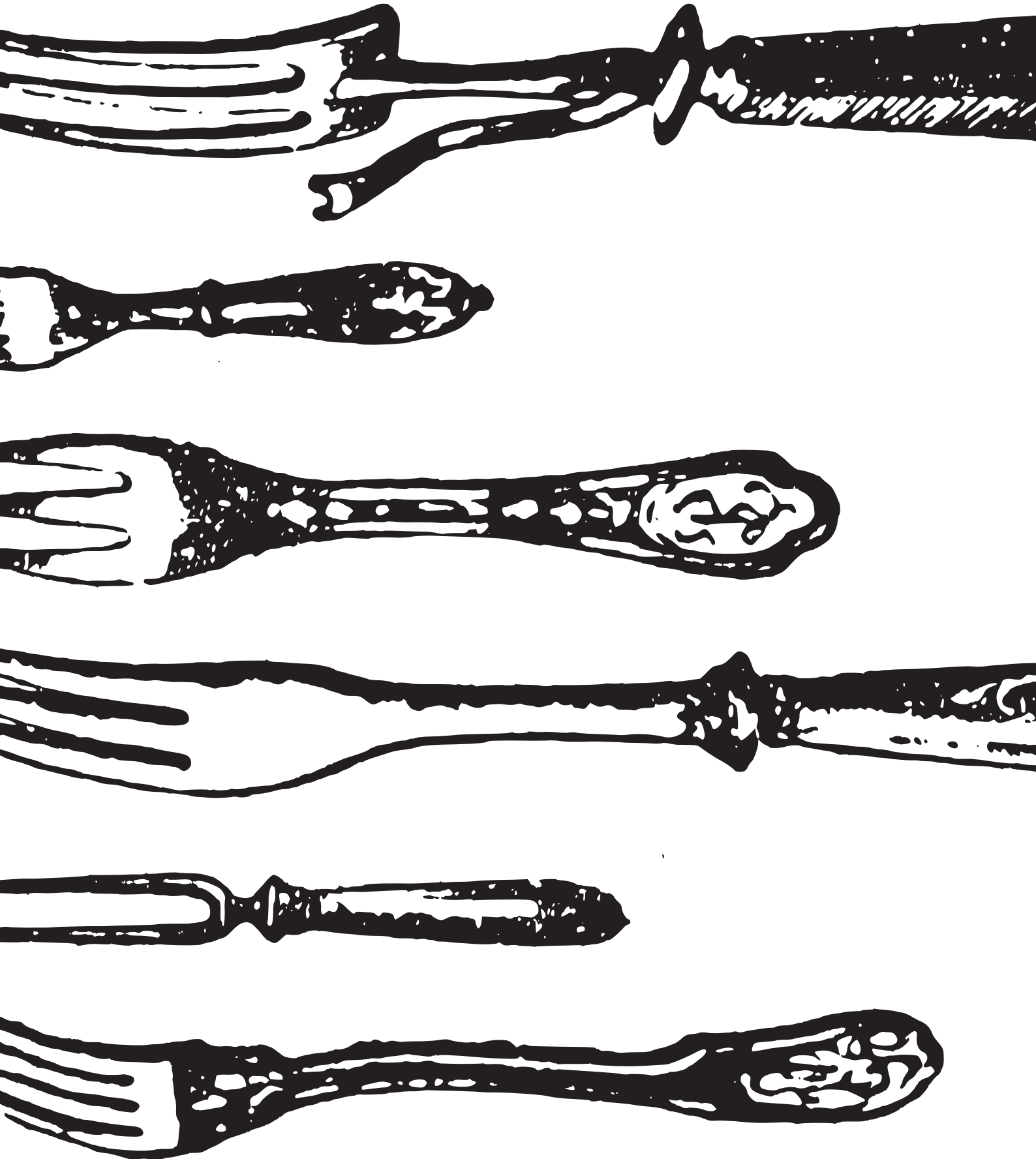
капуччино • латте маккиато

У єгипетських кав'ярнях розповідали арабську легенду про те, як в Ефіопії один пастух помітив, що кози, які поїли ягід з кавового куща, не сплять, а всю ніч граються, стрибають.

способы заваривания кофе

Размер зёрен кофе играет важнейшую роль в определении его сортности — считается, что чем крупнее зерно, тем спелее была ягода, из которой его получили, и соответственно, тем лучше настой кофе из такого зерна. Зерна кофе сортируются по размеру с помощью нескольких сеток с ячейками разного размера.

абвгдеёжзийклмнопрстуфхцчшщъыьэюя

---

*

(2013)

Bitter Cyrillic

а б в г ѓ ґ д е ѐ ё ж з и й й́ к ќ л м н о п р с т у у̌
ф х ч ц ш щ я ъ ь ы љ њ ѕ є з і ї ј ћ ю ѹ
а б в г ѓ ґ д е ѐ ё ж з и й й́ к ќ л м н о п р с т у у̌
ф х ч ц ш щ я ъ ь ы љ њ ѕ є з і ї ј ћ ю ѹ

а б в г ѓ ґ д е ѐ ё ж з и й й́ к ќ л м н о п р с т у у̌
ф х ч ц ш щ я ъ ь ы љ њ ѕ є з і ї ј ћ ю ѹ
а б в г ѓ ґ д е ѐ ё ж з и й й́ к ќ л м н о п р с т у у̌
ф х ч ц ш щ я ъ ь ы љ њ ѕ є з і ї ј ћ ю ѹ

А Б В Г Ѓ Ґ Д Е Ѐ Ё Ж З И Й Й́ К Ќ Л М Н О П Р С Т У У̌
Ф Х Ч Ц Ш Щ Я Ъ Ь Ы Љ Њ Ѕ Є З І Ї Ј Ћ Ю Ѹ Ұ
А Б В Г Ѓ Ґ Д Е Ѐ Ё Ж З И Й Й́ К Ќ Л М Н О П Р С Т У У̌
Ф Х Ч Ц Ш Щ Я Ъ Ь Ы Љ Њ Ѕ Є З І Ї Ј Ћ Ю Ѹ Ұ

А Б В Г Ѓ Ґ Д Е Ѐ Ё Ж З И Й Й́ К Ќ Л М Н О П Р С Т У У̌
Ф Х Ч Ц Ш Щ Я Ъ Ь Ы Љ Њ Ѕ Є З І Ї Ј Ћ Ю Ѹ Ұ
А Б В Г Ѓ Ґ Д Е Ѐ Ё Ж З И Й Й́ К Ќ Л М Н О П Р С Т У У̌
Ф Х Ч Ц Ш Щ Я Ъ Ь Ы Љ Њ Ѕ Є З І Ї Ј Ћ Ю Ѹ Ұ

---

*

ht HUERTA TIPOGRÁFICA

Лора кириллица

каллиграфия и шрифт

классическая *гарнитура* в четырех начертаниях

Regular, *Italic*, **Bold**, & ***Bold Italic***

ритм и движение

Приёмы каллиграфии очень разнообразны. Могут использоваться завитки, лишние хвостики, крючки, попеременно твёрдые и мягкие штрихи, плавные и ломаные линии, угловатость, вписывание букв или слов в геометрические фигуры или определённое пространство, вычерчивание одних букв и оставление невычерченными других, расположение букв или слов в виде определённых фигур (арабская каллиграфия), рамки, вычерчивание букв двойным или тройным слоем, проведение параллельно уже проведённым линиям линий иного цвета и другие.

Raleway кириллица

Исследователи разработали систему лечения

Ученые из Великобритании создали препарат, который может омолодить иммунную систему пожилых людей, резко снизив смертность от гриппа Исследователи разработали систему лечения, которая значительно повышает способность бороться с гриппом и другими вирусами. По заявлениям специалистов Оксфордского университета, после инъекции нового химического вещества пожилые люди получат иммунную систему, как у 20-летних.

которая значительно повышает способность бороться

Ученые из Великобритании создали препарат, который может омолодить иммунную систему пожилых людей, резко снизив смертность от гриппа. Исследователи разработали систему лечения, которая значительно повышает способность бороться с гриппом и другими вирусами. По заявлениям специалистов Оксфордского университета, после инъекции нового химического вещества пожилые люди получат иммунную систему, как у

с гриппом и другими вирусами. По заявлениям специ-

Ученые из Великобритании создали препарат, который может омолодить иммунную систему пожилых людей, резко снизив смертность от гриппа. Исследователи разработали систему лечения, которая значительно повышает способность бороться с гриппом и другими вирусами. По заявлениям специалистов Оксфордского университета, после инъекции нового химического вещества пожилые люди получат иммунную систему, как у

алистов Оксфордского университета, после инъекции

Ученые из Великобритании создали препарат, который может омолодить иммунную систему пожилых людей, резко снизив смертность от гриппа. Исследователи разработали систему лечения, которая значительно повышает способность бороться с гриппом и другими вирусами. По заявлениям специалистов Оксфордского университета, после инъекции нового химического вещества пожилые люди получат иммунную систему, как у

нового химического вещества пожилые люди получат

Ученые из Великобритании создали препарат, который может омолодить иммунную систему пожилых людей, резко снизив смертность от гриппа. Исследователи разработали систему лечения, которая значительно повышает способность бороться с гриппом и другими вирусами. По заявлениям специалистов Оксфордского университета, после инъекции нового химического вещества пожилые люди получат иммунную

иммунную систему, как у 20-летних. Это позволит людям

Ученые из Великобритании создали препарат, который может омолодить иммунную систему пожилых людей, резко снизив смертность от гриппа. Исследователи разработали систему лечения, которая значительно повышает способность бороться с гриппом и другими вирусами. По заявлениям специалистов Оксфордского университета, после инъекции нового химического вещества пожилые люди получат

старшего возраста, как минимум, хорошо реагировать

Building Cyrillic fonts together

Alexei Vanyashin

Learn Cyrillic is a resource I created to teach type designers across the world to master Cyrillic. The idea came to mind when I realized that there is lack of information on Cyrillic type—unless you can read Russian. When I studied type design in Moscow we were taught by our instructors that only natives can design proper Cyrillic forms. I wanted to challenge that.

In 2011 I ran a Cyrillic workshop for alumni of the Royal Academy of Art in The Hague along with fellow Russian designers Irina Smirnova and Gayaneh Bagdasaryan. I realized that some of the students' works were remarkably accurate, and that anyone with proper training could design as good a Cyrillic script as a native. Shortly after, I launched learncyrillic.tumblr.com and published a quick guide on Cyrillic.

I received emails from designers around the world seeking advice. During my consultations, there was always a dilemma between choosing correct Cyrillic forms and fitting in with the spirit of the typeface. I listened carefully to what designers said and adjusted my methodology accordingly. I learned to be less involved, to stand back and focus on providing more references and hints rather than direct solutions. It's about guiding, not steering.

By summer 2013 I was consulting on seven Open Fonts with Cyrillic for the Google Fonts library. It was one of my best experiences. The benefit of working with a F/LOSS methodology was the back-and-forth collaboration, involving direct drawing of shapes by both parties. In general it took between three to eight iterations to agree on appropriate Cyrillic forms with the author.

When consulting on proprietary fonts I usually don't have this luxury and freedom of expression. Everything is tied to a deadline.

Here is a preview of fonts with Cyrillic support that are already published or will appear soon:

- Exo by Natanael Gama
- Lora Cyrillic by Olga Karpushina, Cyrillic extension by Alexei Vanyashin
- Arvo Cyrillic by Anton Koovit
- Kaffeesatz Cyrillic by Sol Matas
- Raleway Cyrillic by Pablo Impallari
- Oswald Cyrillic by Vernon Adams
- Merriweather Cyrillic by Eben Sorkin
- Bitter Cyrillic by Huerta Tipográfica

Another thing I realized was that a script has many dialects rather than one norm. A designer from Serbia has a different view on the shapes and proportions of letters compared to his Bulgarian and Russian colleagues—which one is correct? I believe there isn't one correct form. All the different styles are necessary to express national identity. Today, Russian type designers are very intolerant to the "international" Cyrillic style, perhaps in the same fashion as chauvinists when they hear a foreign accent. I feel this is wrong, and should be resolved by educating and providing an opportunity to fine-tune with the peculiarities of our script.
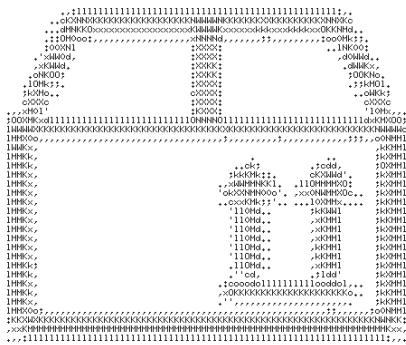
Forging foundries

ANA ISABEL CARVALHO AND RICARDO LAFUENTE

The Libre type design ecosystem has flourished in recent years. First, with hubs like Open Font Library, started in 2006, then followed by smaller, independent Libre foundries such as The League of Moveable Type, OSP Foundry, and the VTF Type Foundry. In 2010, Google Fonts came onto the scene, kickstarting the rise of Libre typography by sponsoring type designers around the world to produce more Libre fonts, in a collection that's growing bigger every day.

But there hasn't been much recent activity outside the existing hubs and foundries. We think this is caused by the difficulties of publishing a font. The act of packaging and publishing is an integral part of the type design process, but it's under-documented when compared against the act of designing a typeface. Packaging and publishing are not glamourous tasks, and we believe that the lack of a standard workflow for postprocessing and distributing a font is a major bottleneck in production.

After finishing the design of a font, as well as taking care of spacing and hinting, there are many small tasks required before a public release:
- choosing the output file formats
- picking a license
- setting and validating font metadata
- documenting the work (the README and FONTLOG files)
- ensuring compatibility with various systems
- creating a downloadable package file
- uploading everything to a server

If a designer is also running their own type foundry, there are additional tasks:
- creating new pages for the font
- updating the website text
- adding specimens and/or preview images
- adding classification or category
- uploading the package to other font distribution sites (if applicable)
- maintaining the website backend
- periodically ensuring every published version and metadata are up to date

The lack of standard ways to carry out these tasks makes packaging and distribution of fonts difficult and highly individual, with each designer and/or foundry devising their own preferred formats, methods and workflows for font postprocessing and publication. Raphaël Bastide addressed this problem with the Unified Font Repository[1], a standard proposal for Libre type font distribution.

Debian, one of the oldest and stablest Linux distributions, has provided us with clues for structuring a potential solution. Software for the Debian operating system can be found in a central repository. This repository is curated in a decentralized fashion by volunteers, organized in a clear hierarchy, who are in charge of reviewing software and packaging it for Debian. The Debian package model decouples the roles of the developer and the packager. The developer is responsible for coding the software, whereas the packager is in charge of reviewing and preparing the code to be distributed and used. Developers don't need to further burden themselves with the intricacies of packaging software for specific operating systems, and packagers are an important addition to the development flow by ensuring and certifying the integrity of software. This clear division creates a structured and standardized workflow for the release of software, minimizing the chance for errors and ensuring a smooth collaborative workflow.

We think Libre type design could adopt similar habits. Foundry-in-a-box is an effort at providing a common way to package and distribute typefaces. We propose a flexible format for font packages and a set of tools that take advantage of the format for validating, editing, and publishing fonts, including a foundry creator.

Before proposing such a system, though, we first need to define what a font package actually is. Our definition would be the collection of elements that are essential for releasing and distributing a font or font family: the font files, metadata, and extra information. It could be extended to include specimens, extended documentation, and other miscellaneous data.

Font packages enable a simple workflow for publishing fonts.

1. Design the typeface and save it as a UFO file.
2. Add the user-specified metadata in fontinfo.json (author info, date, license, etc).
3. [Optional] Add a README file with an introduction and description of the font file.
4. [Optional] Add or edit the FONTLOG with the recent additions.
5. Validate the package to ensure everything is ready to publish.
6. Publish the package to the remote Git repository.

Editing and updating existing packages is also straightforward.

1. Update the typeface's UFO file.
2. Update fontinfo.json, README, FONTLOG if necessary.
3. Validate the package to ensure everything is ready to publish.
4. Log the changes in a Git commit message.
5. Publish the package to the remote Git repository.

So far, we've developed a proof-of-concept command line application, *fib*[2], in order to deal with all of these steps. This forms the best base to later develop a desktop or browser-based application to create and edit font packages.

We're also developing a special mode of the *fib* tool for a use case that will be familiar to type designers: the ability to create and maintain a simple type foundry website, which we're already using for our own foundry project. The generator is based on a list of fonts to include, as well as miscellaneous content. The generated site is plain HTML, generating using the Jinja templating language.

All of this is a tentative first step in thinking about packaging and distribution as crucial steps in type design. To enrich the Libre type ecosystem, we need to take some of the frustration out of making typefaces available and maintaining foundries.

REFERENCES

1. Unified Font Repository *https://github.com/raphaelbastide/Unified-Font-Repository*
2. Fib command-line tool *https://gitorious.org/manufacturaind/fib*

Feeling design through type

GINGER COONS

A few times a year, I teach non-designers a little bit about the basics of design. Just a short workshop, a couple hours long. The people I'm teaching are young professionals, enrolled in a graduate program that aims to spit them out into the workplace with a set of up-to-date skills. The vast majority of them are good at research, writing, and even management, but visual literacy isn't in their standard skill set. More importantly, the idea of design as a functional activity, rather than a veneer of prettiness, is pretty foreign.

Though I teach a number of basic design principles—things like colour, balance, and pattern—a large portion of the session is taken up looking at type. Most people come in with some knowledge of type, on the popular level of "Helvetica good, Comic Sans bad." My task is to explain why different typefaces do different things, and help the participants in the workshop develop some kind of instinct for what to use, when.

We work with Libre type specimens, for the most part, though that's not readily apparent at the start. I bring a stack of pangrams with me, written in a broad collection of fonts, from Linux Libertine and OSP-DIN to freeware novelty fonts that look like they're dripping blood. We play some games with those specimens. In groups, I ask them to pick out which of their specimens look futuristic, or serious, or classic, or fun. They debate amongst themselves and have to come up with a consensus decision for each category. Inevitably, a serif or handwriting font ends up being the classic one.

But the ever-important question is "Why?" Why is one typeface more classic than another, what makes a drippy-looking font fun instead of horrifying? Why does monospace feel futuristic to a particular group? Most participants in the workshops have similar intuitive ideas about what the feelings behind particular fonts are and, by extension, what uses those fonts might be put to. By asking them to put those intuitions into words, I ask them to articulate, both to the other participants and to themselves, just what characteristics, assumptions or associations lend those fonts their particular feelings.

It seems a trivial thing, getting people to talk about how fonts make them feel, but it's a first step in a certain kind of visual literacy. Realizing that styles of type exist, that not all classic-looking serifs are Times New Roman (something I've often seen Linux Libertine mistaken for), and that a designed object—even something as simple as a poster for a bake sale—is an assemblage of visual elements of which type is a crucial part, all of these little things matter in instilling a basic understanding of how and why design works, and how its most basic principles can be put to use.
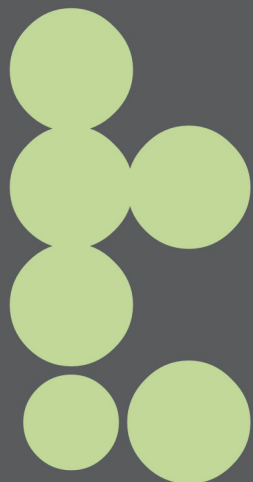
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase
Showcase

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

0 x A000

Regular	Hamburgevons
Dots	Hamburgevons
Monochrome	Hamburgevons
Pixelated	Hamburgevons
Squarish	Hamburgevons
Boxes	Hamburgevons
Regular Mono	Hamburgevons
Dots Mono	Hamburgevons
Monochrome Mono	Hamburgevons
Pixelated Mono	Hamburgevons
Squarish Mono	Hamburgevons

0xA000 font family

ØYVIND KOLÅS
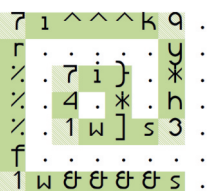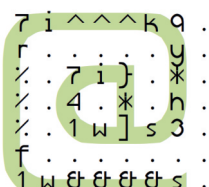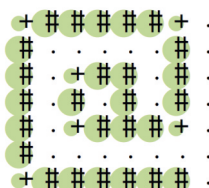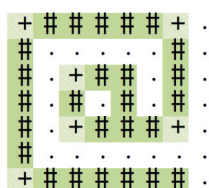
Design inspiration is often fuelled by constraints. In that spirit, 0xA000 started out as an experiment in trying to create a pixel font with greyscale squares for anti-aliasing. The initial ASCII set was made first as an image in GIMP and a custom C program to turn the image into a UFO file where each set pixel referenced a component.[1] For easier editing, the program was changed to output an XPM-inspired text file description of the font, where the character mapping for different greyscale pixel values can be controlled.

By coding/creating/designing a new set of components—puzzle pieces corresponding to the different characters used to design the glyphs—new visual variations for the family can be created. When doing manual anti-aliased drawing, known as pixeling, one imagines the curvature of the shape intersecting with the pixel's geometric area. The scope and power of expression of the font creation pipeline expand when adding puzzle pieces corresponding to these imagined shapes. It is possible to create semi-legible pixel fonts with a 3px-high lower-case grid, thus the 3x3 set for drawing a lower-case "o" was a starting point.

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

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

1. Many font file formats support components, reusable vector shapes. These are normally used during type design to share vector shapes, like the undecorated base glyphs "A," "E," and "O" for Ã¥Ã¢Ãªáº½Ã£ and Ã¶.

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

THE SCREENLESS OFFICE

WHAT'S WRONG WITH SCREENS?

YESTERDAY'S TOMORROWS
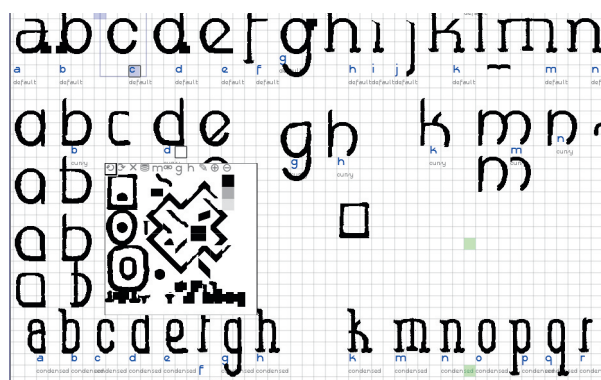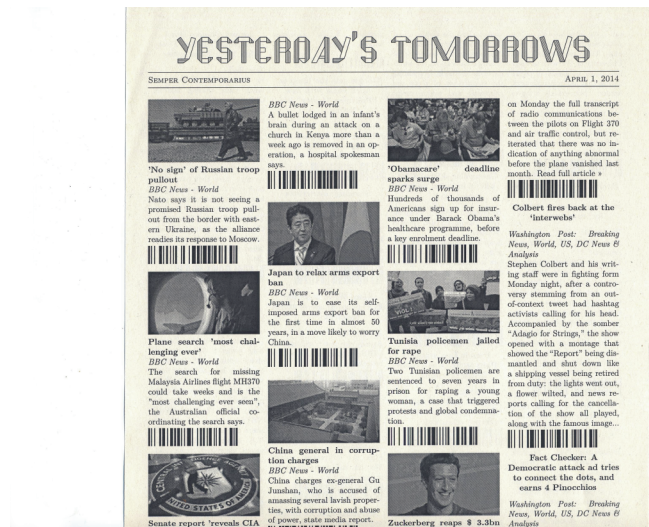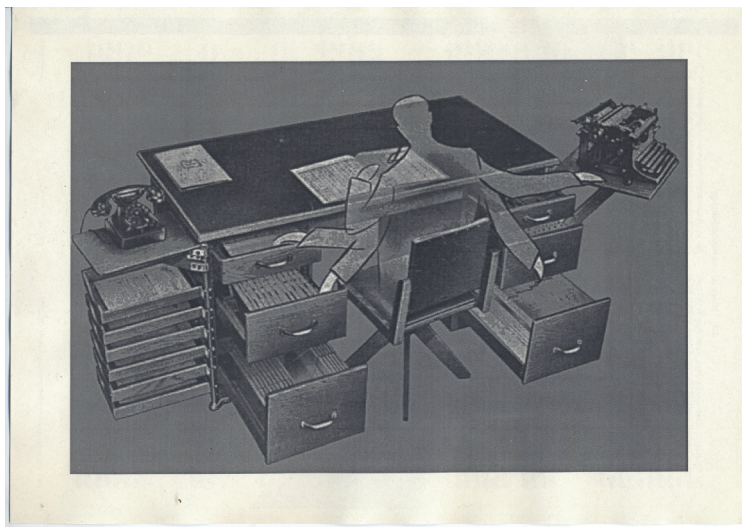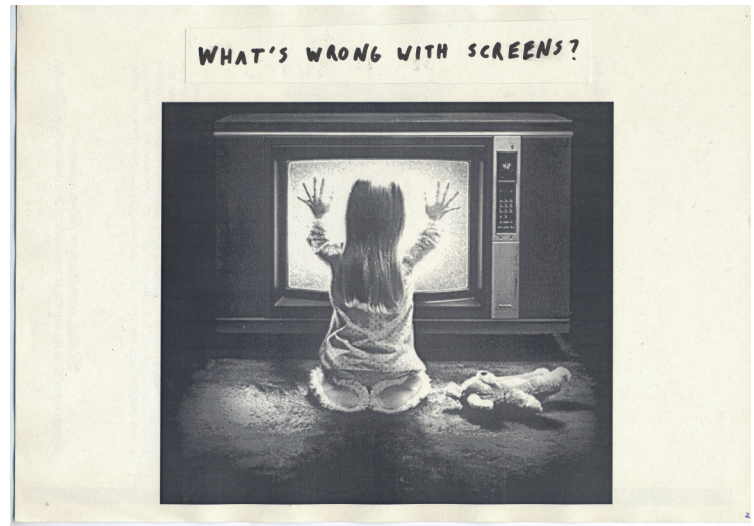
The Screenless Office

BRENDAN HOWELL

The Screenless Office is an artistic operating system for working with media, which eschews the use of a raster-based display. The goal of the project is not to produce techniques that can be justified in terms of speed and efficiency. Instead, the Office describes an alternative mode of everyday life with networked computation which might be calmer, more embodied, and personal. The system is constructed using Free/Libre/Open hard- and software components for print, databases, web-scraping, and tangible interaction.

Some of the experimental components include a printed daily newspaper, a receipt printer that spits out tweets and fortune cookies, a document camera, a hybrid note card database and a "smart" Rolodex contact organizer. The core *Office Manager* and prototype *Bureau* modules will be released soon. Anyone experienced with Python and templating systems should find it relatively easy to extend, hack or customize their own Screenless Office.

http://wintermute.org/brendan/?e=261

A–B
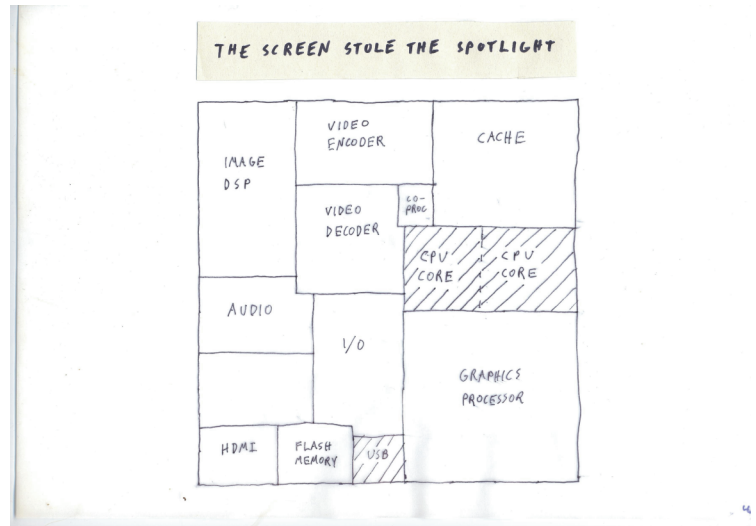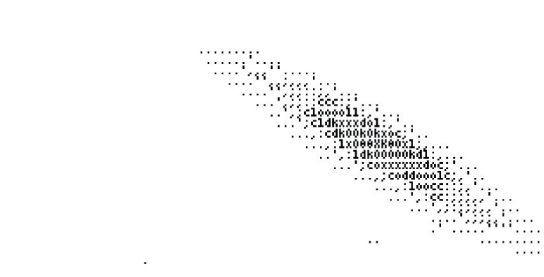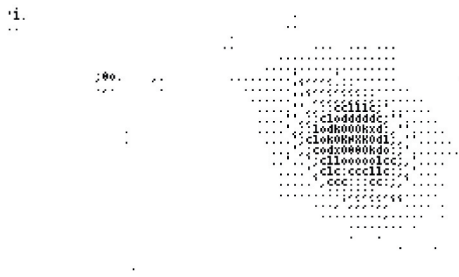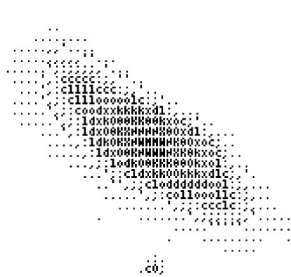
http://github.com/uplaod
Tags: revival, original creation, text, display
License: SIL Open Font License
Notable typefaces: Futura Renner, Lil Grotesk, Young Serif
Date created: –

Citype

http://www.citype.net
Tags: thematic, cities, display, revival
License: Creative Commons—Attribution (CC BY 3.0)
Notable typefaces: Amsterdam, Bergen
Date created: -

Cyreal

http://www.cyreal.org
Tags: text, cyrillic
License: SIL Open Font License (some, but not all fonts on Cyreal are Libre)
Notable typefaces: Alice, Lora
Date created: 2010

Huerta Tipográfica
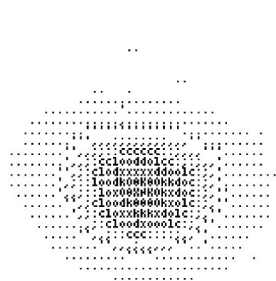
http://www.huertatipografica.com
Tags: text, Latin typography
License: SIL Open Font License (some, but not all fonts on Huerta Tipográfica are Libre)
Notable typefaces: Alegreya, Alegreya Sans, Andada
Date created: 2009

Janusz Marian Nowacki

http://jmn.pl/en
Tags: text, LaTeX
License: SIL Open Font License
Notable typefaces: Antykwa Półtawskiego
Date created: 2007

Kontrapunkt

http://www.kontrapunkt.com/type
Tags: type agency, display
License: Creative Commons Public license
Notable typefaces: Kontrapunkt
Date created: 1985 (2004 for the Libre type library)

OSP Foundry
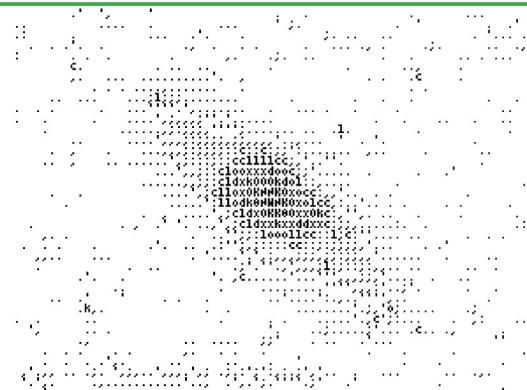
http://ospublish.constantvzw.org/foundry
Tags: revival, text, display, derivative
License: SIL Open Font License
Notable typefaces: Libertinage, NotCourierSans, OSP-DIN, Reglo
Date created: 2010

OXSHARK FONTWORKS

HTTP://OXSHARK.ORG
TAGS: REVIVAL, DISPLAY, DERIVATIVE
LICENSE: SIL OPEN FONT LICENSE
NOTABLE TYPEFACES: FACHADA, SERRERIA
DATE CREATED: 2013

PRACTICE FOUNDRY

HTTP://PRACTICEFOUNDRY.COM
TAGS: TEXT, DISPLAY, FUNCTIONAL
LICENSE: SIL OPEN FONT LICENSE
NOTABLE TYPEFACES: COLLATOR, FABRICA, NARRATOR
DATE CREATED: 2011

Glukfontsz
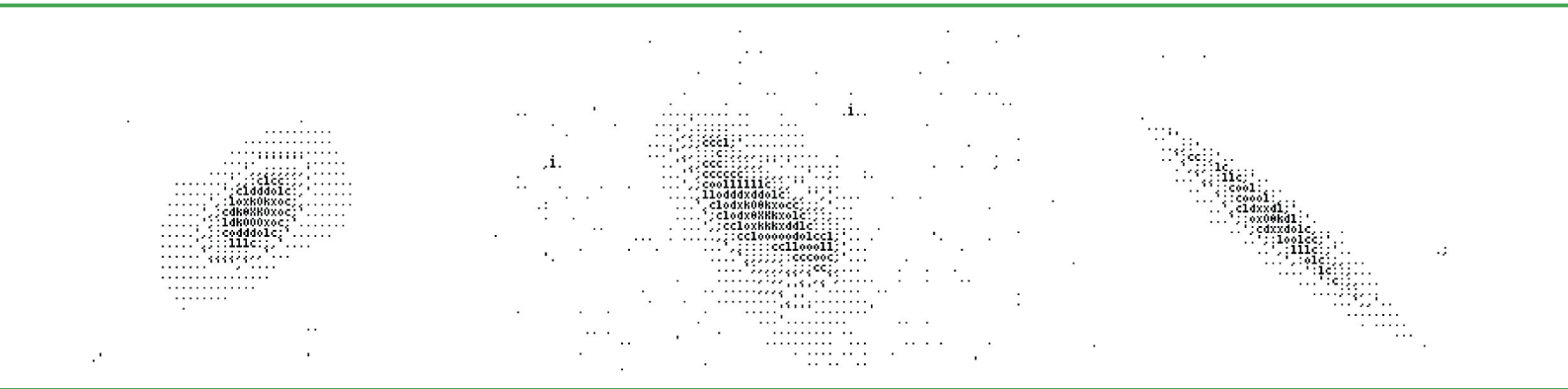
http://www.glukfonts.pl
Tags: expressive, script, text, display
License: SIL Open Font License
Notable typefaces: Foglihten, Resagokr
Date created: 2010

Google Fonts

http://www.google.com/fonts
Tags: font warehouse, text, display
License: SIL Open Font License, Apache
Notable typefaces: Noto, Open Sans, Playfair
Display, Roboto
Date created: 2010

Greek Fonts Society

http://www.greekfontsociety.gr
Tags: greek, LaTeX
License: SIL Open Font License
Notable typefaces: GFS Bodoni, GFS Didot
Classic
Date created: 2005

THE LEAGUE OF MOVEABLE TYPE
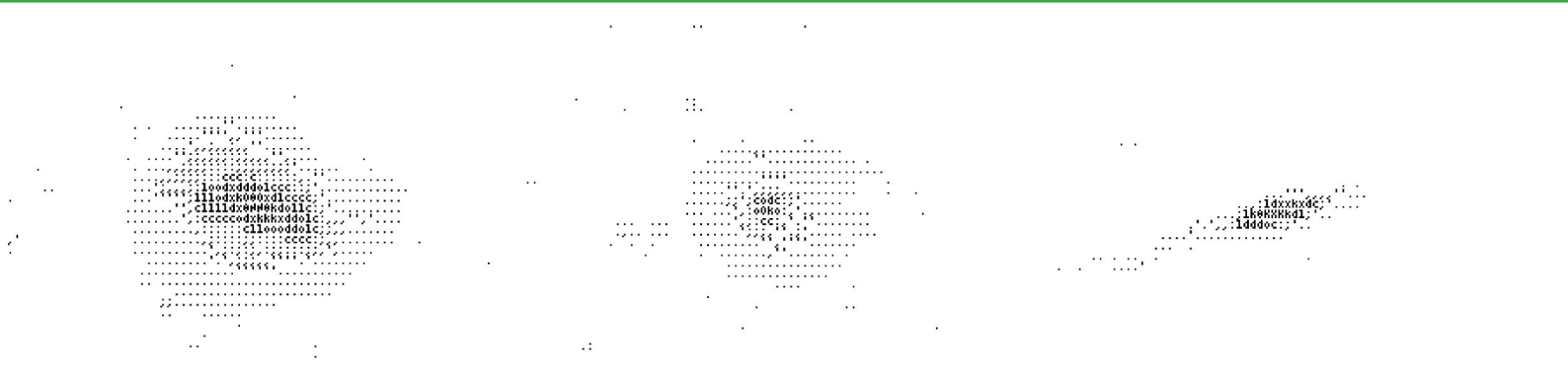
HTTPS://THELEAGUEOFMOVEABLETYPE.COM
TAGS: TEXT, DISPLAY, REVIVAL
LICENSE: SIL OPEN FONT LICENSE
NOTABLE TYPEFACES: LEAGUE GOTHIC, OSTRICH SANS
DATE CREATED: 2009

Omnibus Type

http://www.omnibus-type.com
Tags: text, display
License: SIL Open Font License
Notable typefaces: Archivo, Asap, Chivo
Date created: 2011

OPEN FONT LIBRARY

HTTP://OPENFONTLIBRARY.ORG/
TAGS: REPOSITORY, TEXT, DISPLAY, REVIVAL, DERIVATIVE
LICENSE: SIL OPEN FONT LICENSE AND OTHER
PERMISSIVE LICENSES
NOTABLE TYPEFACES: AVERIA, GNU TYPEWRITER, PECITA
DATE CREATED: CREATED IN 2006, RELOADED IN 2011

SIL

http://scripts.sil.org
Tags: notreallyafoundrybutaninitiative, multiscript,
unicode
License: SIL Open Font License
Notable typefaces: Doulos, Gentium
Date created: 2003 (2005 for the Open Font License)

Velvetyne

http://www.velvetyne.fr
Tags: cyberpunk, experimental, display
License: mostly SIL Open Font License
Notable typefaces: BizMeud, Fengardo Neue, Grotesk, Rupture
Date created: 2010

Libre type
foundry Atlas

FRANK ADEBIAYE

Pirate Party identity and PPPoster

SAMUEL RIVERS-MOORE

PIRATE PARTY IDENTITY

The Pirate Party is a young political party founded in Sweden in 2006, gently seeping all over the world. Post-digital culture is part of its DNA and its program mainly focuses on human rights and fundamental freedoms issues, as well as on contemporary copyright problematics.
Surfing on a general exasperation towards conventional political parties, it presents itself as an alternative to Left/Right divisions, as well as a way to "upgrade" democracy for a more horizontal political system. The Pirate Party acts as a bridge between, on the one hand, some more or less radical and structured political movements (Anonymous, WikiLeaks, GNU, etc.) and on the other hand, an institutional way of influencing the democratic process. In most countries, its visual identity is quite conventional and its communication is sometimes clumsy, as is the case in France.

Since the Pirate Party depicts itself as something different, my project aims to experiment with unconventional methods to design its visual identity. I also aim to imagine some envelopes that fit its ideas in a faithful way, rather than in a simplistic or watered down one.
This project is a pirate project in the sense that it hasn't been commissioned by the PP, nor has it been conceived to be used by the party, to be functional or cosmetic. It simply tries to show what could be the visual identity of a political party if this party had an unconventional, self-critical approach to communication, sincerely saying "this is what we are."

The importance of being recognizable in the visual landscape doesn't mean shapes have to stay rigid. Inspired by water, flags and CAPTCHAs, I developed an undulating logo using Pure Data. Its shape is always changing, influenced by random deformations and twists. It plays with the oxymoron expressed by the party's name, opposing the rigidity of institutional structures to the more dynamic, moving and undefined aspects of activism.
By the same token, the fonts used are, on the one hand, a custom version of Terminal Grotesque, a rough and geometric pixel font drawn by Raphaël Bastide; and on the other hand, the more classical Linux Libertine designed by Philip H. Poll and already used for Wikipedia's logo. Both are F/LOSS fonts.

Parti Pirate

Parti Pirate

Parti Pirate

Parti pirate

Parti Pirate

Parti Pirate

Parti pirate

PirateParty   PirateParty   PirateParty
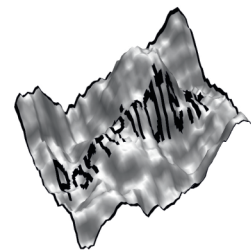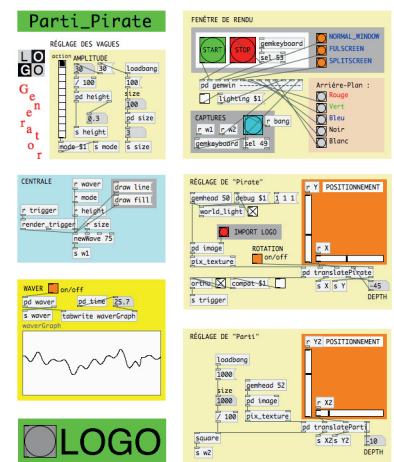
Just loved "Punk Pirate Show as Broadcast On Radio Sutch 2.6.13" by punkpirateshow

Liberté égalité fraternité, les lois ancestrales de la piraterie

@TheMetalinux – le 27.06.13 à 13:33
Parti Pirate

#party #powder #pirate

http://t.co/IKyW4vjuy ✿HU... for SUMMER✿ http://t.co/LRnSMXYObd #boebot #DiggRT #TheEtsyGuy #boebot

@katsorova – le 24.06.13 à 21:48
Parti Pirate

KEIN PFUSCH®: E Grillo? – See on Scoop.it – Pirati
Pirate Party Italy's insight: L'IMU

@partipirata – le 27.06.13 à 14:43
Parti Pirate

RT @PartiPirate: Le parti pirate recrute ses candidats pour les élections de 2014 #municipales #europeennes http://t.co/Xp2uhtxnDZ

@Elus20 – le 27.06.13 à 10:14
Parti Pirate

WTF? #ppposter

@Ohax – le 27.06.13 à 13:07
Parti Pirate

Je pirate donc je suis #ppposter

@kadc0m – le 27.06.13 à 13:09
Parti Pirate

PPPOSTER

PPPoster is a small proto-app coded with Processing. It allows the design of simple and strong text-based posters in a very short time without having layout skills. The app works like a minimal text editor. It follows only one strict and direct formal rule: each line of text automatically adjust itself to fit the width of the poster. This strong constraint is an invitation to play. The original idea was to experiment around a tool that could be used by anybody. I pushed the idea a bit further with a program using the same principles to automatically generate posters using tweets containing keywords linked to the Pirate Party.

Automatically converting tweets into official and branded posters is a way to invite the Pirate Party to cede control and to give some spontaneity to its own political communication. It is a way to make the party accept and assume its own criticisms. It is also a way to make the party say "anybody can speak in our name" and "these criticisms are part of us." This project does not pretend to be realistic. Instead, it is very experimental and prospective and aims only to be an inspiration for people who would like to use visual communication in an unconventional manner.

KRITA

Painting for digital artists

http://krita.org

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.

Fontlinge

It organizes your fonts, creates a database, and offers automated specimens. It's a little archaic to install, but that's part of the fun.
http://sourceforge.net/projects/fontlinge

Font Manager

True to its name, Font Manager manages your fonts, and does so visually. It offers preview texts and lets you sort based on a variety of criteria.
https://code.google.com/p/font-manager

Fontmatrix

One of the longest-reigning Linux font managers, Fontmatrix is still the go-to choice for many designers looking to find that perfect font in an expansive collection.
http://fontmatrix.be

Hershey Fonts

A set of single-stroke fonts, designed for drawing with pen plotters or etching text on printed circuit boards. There is an Inkscape extension available to render them properly in SVG documents.
http://emergent.unpythonic.net/software/hershey
http://www.evilmadscientist.com/2011/hershey-text-an-inkscape-extension-for-engraving-fonts

Dadadodo

A probability-based creative text scrambler which generates quasi-surrealist text collages.
http://www.jwz.org/dadadodo

Figlet

A venerable command-line tool to create ASCII-art renderings of text strings. Hundreds of available ASCII fonts make Figlet text a genre of its own.
http://www.figlet.org

Toilet

A successor of Figlet, with support for colour text and ANSI characters.
http://caca.zoy.org/wiki/toilet

Best of type collections

We all love a good collection of type specimens. There's something pleasing about looking through pages of carefully selected typefaces, admiring the sweep of a lower case "y" or imagining where you could possibly use that absurd but compelling novelty font. That joy of discovery is the common thread tying together this "Best of."
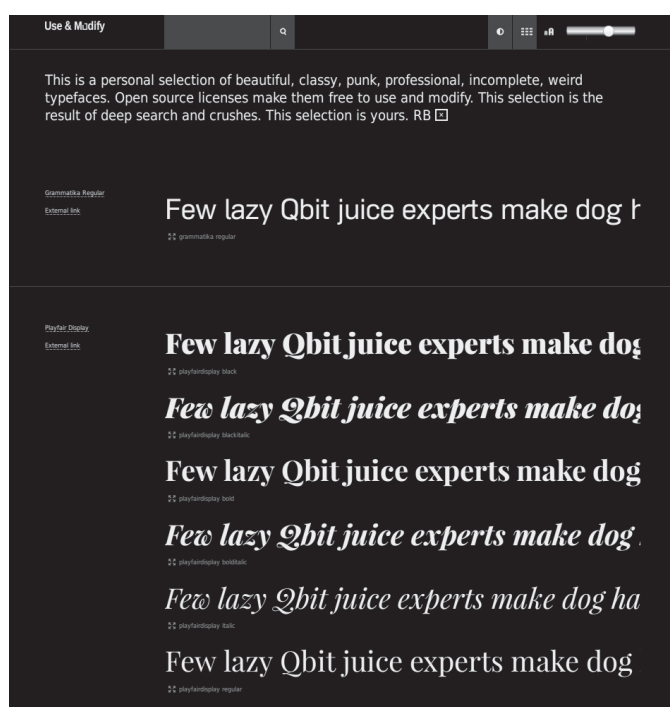
Use/Modify

Raphaël Bastide's growing showcase, *Use/Modify*, is a particularly well-designed repository of "beautiful, classy, punk, professional, incomplete, weird typefaces" from many sources. The common thread—that they allow their unrestricted use and modification—is the principle that defines the project's apt title.

http://usemodify.com

Delubrum

Stephen G. Hartke has maintained a comprehensive list of existing Open fonts which used to live in Geocities, but which now have a home at delubrum.org. While not as designer-oriented as the previous examples, this resource covers several typefaces from the F/LOSS world, along with detailed notes on authorship, license, glyph coverage, styles and TeX support.

http://delubrum.org

HAND-PICKED TALES *from*
ÆSOP'S FABLES
with HAND-PICKED TYPE *from*
GOOGLE FONTS

There are over 650 **Google Fonts** available for free. But, pairing typefaces isn't easy and many of those fonts don't work for typical websites. Part of the 25x52 initiative, this collaborative, ongoing project offers inspiration for using Google's font library.

All passages are from the **Project Gutenberg** transcript of Æsop's Fables. All photographic images are from **Unsplash.com**.

If you'd like to contribute to this project, please **submit a pull request on Github**.

Need help with Github? Want to learn more about web typography? **Sign up for our tutorial events.**

PLAYFAIR DISPLAY, FAUNA ONE

THE GNAT
& THE BULL

A Gnat alighted on one of the horns of a Bull, and remained sitting there for a considerable time. When it had rested sufficiently and was about to fly away, it said to the Bull, "Do you mind if I go now?" The Bull merely raised his eyes and remarked, without interest, "It's all one to me; I didn't notice when you came, and I shan't know when you go away."

We may often be of more consequence in our own eyes than in the eyes of our neighbours.

THE TWO BAGS

FUGAZ ONE, OLEO SCRIPT, MONDA REGULAR

Every man carries Two Bags about with him, one in front and one behind, and both are packed full of faults. The Bag in front contains his neighbours' faults, the one behind his own.

Hand-Picked Tales from Aesop's Fables with Hand-Picked Type from Google Fonts

Phoebe E's project, *Hand-Picked Tales from Aesop's Fables with Hand-Picked Type from Google Fonts*, is a one-page feature of several well-considered combinations of typefaces. An important concern when typesetting in print or on the web, different fonts are tastefully combined into an eclectic showcase of distinct and elegant layouts.

http://femmebot.github.io/google-type

Goodgle Fonts

Spawning from Frank Adebiaye's desire to provide a selection of good samples from Google Fonts, Forthcome's *Goodgle Fonts* is an early effort to single out those "good" fonts to be found in Google's typeface repository.

http://www.forthcome.fr/work/goodgle_fonts

CURTIS LESS TYPEFACE

Fork me on GitHub

Curtis is the name given by Dave DeSandro to a family of geometric sans-serif fonts. Other incarnations exist as Fontstructions: Curtis Heavy and Curtis Pixel 14.

The **Curtis LESS Typeface**, which is based on the Curtis CSS font, takes form in CSS compiled from LESS for added parametric options. All shapes are rendered in the browser, using a combination of background color, border width, border radius, and a heavily reliance on absolute/relative positioning.

Each character is wrapped in a <span> and then depending on the complexity of that character, more empty <span> elements are added to the markup to render each shape. Here's the markup for R:

```
<span class="css_char r">
  R
  <span class="inside split vert"></span>
  <span class="outside split vert"></span>
  <span class="stroke"></span>
  <span class="fill"></span>
</span>
```

The Curtis CSS typeface is an original project by Dave DeSandro and was used in the article Why Art?.

The Curtis LESS typeface is an adaptation of Dave's work by Julien Deswaef using LESS with the intention of having more flexible font sizes and colors while keeping all the original ideas from the Dave's design.

This design pattern is released under a Creative Commons Attribution License. You are free to share and remix this work.

Font size
○ 28px ○ 56px ○ 140px ● 280px

Display options
● Standard ○ Inspect

Custom values
Font size (in px): 280
Background: none
Positive: #000
Negative: #ccc

ABCDEFGHI
JKLMNOPQ
RSTUVWXY
Z?!..
QUICK
BROWNFOX
JUMPS OVER
THE LAZY
DOG.

Before webfonts

JULIEN DESWAEF

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 themself, 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 sIFR[3] 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 font[4] 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 sto change the weight and proportions at will. To display it in the browser, I found some obscure Javascript library created by Walter Zorn[5] 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.

After this little personal project, I kept my eyes open for unusual font drawing experiments in the browser, especially the ones involving code. I was impressed by Dave Desandro's Curtis CSS typeface[6] which is a pure HTML/CSS font using a combination of background colour, border width, border radius and a lot of <span> tags to draw a set of very pleasing fat characters. This is especially remarkable because it doesn't rely on Javascript, making it a very good candidate for the "displays-everywhere" award.

Another experiment to notice is Claudio Cañete's 1k Notebook[7] project to implement a full parametric font in <canvas>—in just under 1 kB. He first used a 3×7 grid of nodes to draw a custom, line-based font. But since this wasn't really using much of the 1024 bytes available, he pushed further by porting a script typeface from the Hershey font collection[8] and tweaked it to draw the glyphs in a continuous single trace. The encoding and decoding of the font is where Claudio's work happens, to stay under that limit he set himself. And this is where the beauty lies.

With the latest HTML specifications, pretty much any typeface can be displayed in a web page, and, don't get me wrong, this is such a relief. But creativity gets its strength from the constraints it faces. There's still room for inventiveness with font rendering in the browser. Just set yourself the limits of your canvas.

FOOTNOTES:

1. FF-Chartwell *https://www.fontfont.com/how-to-use-ff-chartwell*
2. Microsoft core fonts
http://en.wikipedia.org/wiki/Core_fonts_for_the_Web
3. Scalable Inman Flash Replacement
http://en.wikipedia.org/wiki/Scalable_Inman_Flash_Replacement
4. xuvfont *http://xuv.be/static/xuvfont/typo.html*
5. Vector Graphics library by Walter Zorn
http://www.walterzorn.de/en/jsgraphics/jsgraphics_e.htm
6. Curtis CSS typeface
https://github.com/desandro/curtis-css-typeface
7. 1k Notebook *http://www.claudiocc.com/the-1k-notebook-part-i*
8. Hershey fonts *https://en.wikipedia.org/wiki/Hershey_font*

The League of Movable Type—an interview

Micah Rich runs The League of Movable Type, a foundry specializing in F/LOSS type design. Micah was interviewed by Manufactura Independente, discussing the intricacies of running a foundry and producing Libre fonts.

This interview has been shortened and edited for clarity.

Manufactura Independente: How did it all start for you? I mean, we usually all have this kind of start: some people we know, most people, the designers in Free Software that we know start from a traditional background, and then get into Free Culture somehow or using Free Software. And we would really like to know: how did the whole motivation for starting a Free type foundry start? What's your story?
Micah Rich: OK, so I started with a partner, Caroline Hadilaksono, and we had gone to college together and taken a few traditional graphic design classes and started working together. After school, we started working together professionally as well. And I didn't really know much about programming—I had only attended webdesign and all that but I spent a lot of my senior year learning Ruby and Ruby on Rails, and a lot of Free and Open Source software that contributed to me being able to learn how to program and make my senior thesis. When we graduated and started our company, we were learning more about new technologies with the Internet, and browsers were starting to be able to use more than what used to be just web-safe fonts.

The capability was there, but none of the type foundries that we saw were really okay with using their fonts on the web. They basically considered it stealing. I remember I was looking on a type forum, Typophile, and there a student had posted a question in one of the forums saying "Does any-body here, any of the great typographers and type designers, know of any Open Source font that I can use for free in a student project?" And there was a giant backlash from all the professional type designers down there screaming "How dare you? How could you even ask a thing like that? You're ruining our livelihood!" I saw that and I thought "That's crazy. What?" I'm coming from the world of programming where you learn so much by dissecting people's Open Source code, and you use that code to build the things that you need. I thought "We need to take that concept and apply it to type design." Caroline had a font that she had made in school and we knew a couple

of other people that had done that as well. Then, you know, we could just put it out there and find people that supported that ideal.

So then you have a foundry. How did you get in touch with the licensing of typefaces—namely, beginning to use the Open Font License?
That's a good question and I'm not entirely sure. I remember when we first started we were already fans of the Creative Commons licenses. We had sort of found that they didn't entirely fit the medium of typography. I think this is partly because we believe, at least in the US, that type is considered software and Creative Commons doesn't always relate well to software, I think.

You mentioned that you were doing this with other people. And we suppose that they were okay with this direction for Free and Open licenses. But did you find the opposite? In general, how do you feel is the reaction to the proposal to release a font under the Open Font License, with regards to people you've worked with?
While a lot of outside people are afraid of the idea behind the Open Font License, everyone that I worked with has loved the principle. The first couple of people that we contacted that we didn't know but we wanted to have fonts they had made in The League were already giving away their fonts. They basically had the same mentality of "Free is good for everybody" and "This is a fun thing I made that I want everyone to have." I think we came at it when we were talking to those people with the intention of educating as well, and I think everybody was on board with that idea. So everybody we actually worked with already had that mentality and that's why we approached them.

We worked with a couple of companies outside of the sphere of Open Source type as well. At one point, we were talking to a highly regarded professional type foundry, and were talking about possibly being able to commission a font; in other words, pay them to make the font and release

Knewave

Raleway

League Gothic

Junction

OSTRICH SANS

Chunk

BLACKOUT

League Spartan

it as Open Source; they totally loved that idea. This was a few years ago and it didn't end up happening for reasons on our end, but they were very much into it.

Especially considering we were talking about being able to fund the ideal of making type for a living and give back to the community. Since then, there have been a couple of commissions where different companies have come to us saying "Hey, we love the fonts that you have. We would like to expand them." And we always say "Well, that comes with a clause that this is going to go back to the Open Source community." They've all been totally cool with that too. So I think it's interesting that the Open Source mentality, once somebody says "this is important to us," can prove its value to other people. Most people who are not initially supportive end up jumping on board.

Right now I'm going to pick on an expression that you just used and that you use in your manifesto as well and I'm quoting "Maybe there's nothing wrong with giving things away sometimes." Actually the two of us [Manufactura Independente] have been talking recently about this whole idea of "giving away" and how the term usually suggests that you're relinquishing control on one hand, but actually being subtracted of the thing that you have. You give it away and don't have it anymore. We were wondering if the term "sharing" wouldn't be closer to what we are actually doing with digital files.

How much do you find that you give away when you release a font? Since it's not just an art work but also a tool, an object. How do you feel about the act of giving away or sharing those objects that you make?

I think in the end what happens is you say "You are welcome to use this thing" and people come back and are so grateful and happy about it that it's a huge personal reward. And at the same time I think it's a professional reward too. Tyler Finck is one of the people who has contributed the most to The League with his type designs. When we first met him, it was something he was doing in his back room and not showing to that many people. Now, a good chunk of his living comes from making type. And he continues to give it away for free and share it. And I think that because he did that, he can now do it even more.

Many things have changed since you started. Google Fonts, for instance, was something that could not have been thought about six years ago and right now it's something that's unavoidable in many type discussions. Open fonts don't seem to be a niche thing any more, and are broadcast billions of times a month to people's browsers. What's your view on what's going on right now?

We have a good relationship with TypeKit, which was the first company to start trying to distribute fonts on the web. They have a really good outlook on Open Source as well as on the commercial side. Then Google Fonts came along and made that sort of mass-digestible. And even though I would agree that using web fonts has grown significantly to the

point where pretty much everybody does it, at this point I still think that Open Source typography has not entirely caught on. And that's the mission I continually keep pushing forward. I think people see Google Fonts as a place to get free fonts to use. There's a whole mass of people that want to learn more about how type is made, about some realities and history of type. There's a lot more that goes into web fonts than just using them.

What's your experience with the reach of your fonts, what other people do with them? Not just using them in their work but changing them, modifying them, republishing them... You've been active for so long, certainly you have a few examples or a view on how well things might or might not work.
Yeah, I think that's an awesome question. It's so rewarding to see that some person that I had no idea existed made our font ten or twenty times better than it was and says "Hey, here you go. This is an update that you should totally

> I'm coming from the world of programming where you learn so much by dissecting people's Open Source code, and you use that code to build the things that you need.
> I thought "We need to take that concept and apply it to type design."

give to everybody else." There have been a handful of awesome experiences like that. And I've seen a lot of that too, not necessarily just with The League fonts but other Open Source fonts that I know of. But I think the reason that contributors' collaborative nature is still sort of under the radar is because there's a bunch of weird tools and a bunch of unconnected workflows. Almost everybody has a different way of working on things and there's no real common way that everybody has agreed is the most efficient way to contribute back. I mean, that's something I want to try to help improve somehow with The League but I think that's why it's lacking today, because there [are] so many different ways that everybody does things that it's difficult for people to really collaborate.

Because usually that's mentioned as one of the strengths of Free and Open Source software, right? This idea of diversity is usually taught as one of the great things about Open Source—the freedom and diversity. But then, you mention that at some point, we could use a more definitive, stable solution.

How do you think that can work?
It's not so much that I think that there should be one tool and one specific workflow, but solutions like the UFO file format are some things that I actively try to support and encourage because a) it's an Open format and anybody can figure out how it works; and b) it's portable and easy to update. When I say weird tools, I mean things like FontLab—I don't use FontLab but I know a couple of people that don't use anything else. As far as I know at this point, you have to install some strange plugins that only export to UFO instead of saving as UFO, and there are other programs that save as UFO natively but have different headers in the file than other programs that save as UFO. And having certain standards, I think there can be many specific workflows, with certain standards in file formats and ways that we can work on the same thing without screwing each other up or rewriting each other's work. UFO is a good example but on top of that I have been trying to convert the people that I know to using a legitimate version control system like Git.

Git is one of those things that is slowly permeating from software to other industries that realise "Hey, this is actually a really nice system if we use certain formats and agree on file names that we will use, we can work on stuff together without screwing up each other's work." I think the type designers that I have met are not quite there yet. The tools that everybody likes to use aren't quite on board with that idea all the time either; some of them help with a workflow like that and some of them hinder it. And just the fact that it looks like a big jumbled mess seems to me to be the problem.

It's not so much that everybody needs one tool and one specific workflow or anything that precise. I think it's just the mentality of collaboration doesn't exist as uniformly as it could.

You operate a foundry and a type foundry itself is much more than just putting some files on a web server. Each one has its own needs, its own way of working for which we're sure you set up, again, your own personal workflow. That is something that the two of us have been wondering recently, tools for foundries.

Can you picture tools for making and maintaining type foundries? Does it make sense to you or is it something that's up to each curator?
It is definitely up to each curator and I think that is always going to be the case. But you seem to be suggesting that there could be a tool that help us come up with a system and I completely agree. Actually, that was something I talked with one of the guys at Google, last year. I kind of gave my two cents on the idea of creating, at the very least, some sort of platform to double check the technical quality of fonts. There is a type of application

Sniglet

Sorts Mill Goudy

Goudy Bookletter 1911

Orbitron

Prociono

League Script

Linden Hill

Fanwood

It's so rewarding to see that some person that I had no idea existed made our font ten or twenty times better than it was and says "Hey, here you go. This is an update that you should totally give to everybody else."

in software programming that is called "continuous integration"—people call it CI for short. You have an application where you write some code, you push the code to this application and it runs a set of tasks to make sure that your code doesn't have typos or other problems and that it will actually run. Not so much the creative aspect of the code but the technical pieces. And that was what I talked to Google about too. They started working on it, I'm not sure if they still are. I think tools like that are sort of in line to what we are talking about. They don't necessarily say "You have to follow this workflow" but "Here are ways and here's an application to make some of those processes, that are really complicated and overly difficult, a lot easier if you could just get into doing it." You know what I mean?

I would love to see tools like that. [...] There's lots of other things that I end up having to do (laughs) but I think that the industry is realizing this and that the culture is right for having tools like that, to help us all to come up with efficient ways to get rid of the junk that we have to do and focus more on the creativity and the collaboration of making type together.

LIBRE GRAPHICS MAGAZINE ISSUE 2.4

Capture

Data capture sounds like a thoroughly dispassionate topic. We collect information from peripherals attached to computers, turning keystrokes into characters, turning clicks into actions, collecting video, audio and images of varying quality and fidelity. Capture in this sense is a young word, devised in the latter half of the twentieth century. For the four hundred years previous, the word suggested something with far higher stakes, something more passionate and visceral. To capture was to seize, to take, like the capture of a criminal or of a treasure trove. Computation has rendered capture routine and safe.
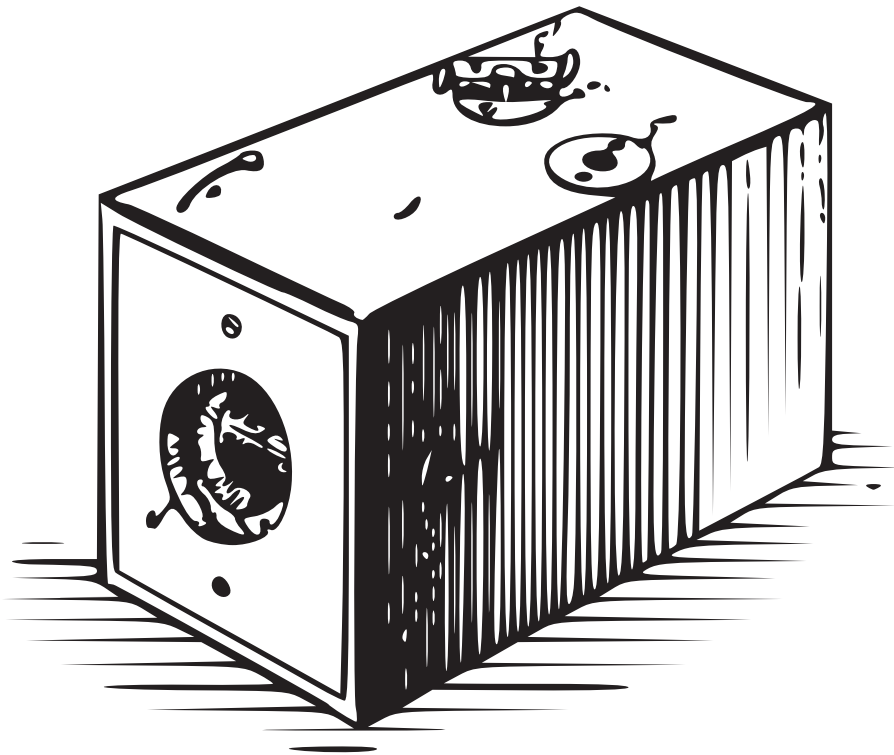
But capture is neither simply an act of forcible collection nor of technical routine. The sense of capture we would like to approach in this issue is gentler, more evocative. Issue 2.4 of *Libre Graphics* magazine, the last in volume 2, looks at capture as the act of encompassing, emulating and encapsulating difficult things, subtle qualities. Routinely, we capture with keyboards, mice, cameras, audio recorders, scanners, browsing histories, keyloggers. We might capture a fleeting expression in a photo, or a personal history in an audio recording. Our methods of data capture, though they may seem commonplace at first glance, offer opportunities to catch moments.

We're looking for work, both visual and textual, exploring the concept of capture, as it relates to or is done with F/LOSS art and design. All kinds of capture, metaphorical or literal, are welcome. Whether it's a treatise on the politics of photo capture in public places,

a series of photos taken using novel F/LOSS methods, documentation of a homebrew 3D scanner, any riff on the idea of capture is invited. We encourage submissions for articles, showcases, interviews and anything else you might suggest. Proposals for submissions (no need to send us the completed work right away) can be sent to *submissions@libregraphicsmag.com.* The deadline for submissions is **May 11, 2015**.

Capture is the fourth and final issue in volume two of *Libre Graphics* magazine. *Libre Graphics* magazine is a print publication devoted to showcasing and promoting work created with Free/Libre and Open Source Software. We accept work about or including artistic practices which integrate Free, Libre and Open software, standards, culture, methods and licenses.

"Phylogenetic Tree
of Type History" V3.1,
by Philipp H. Poll, 2011
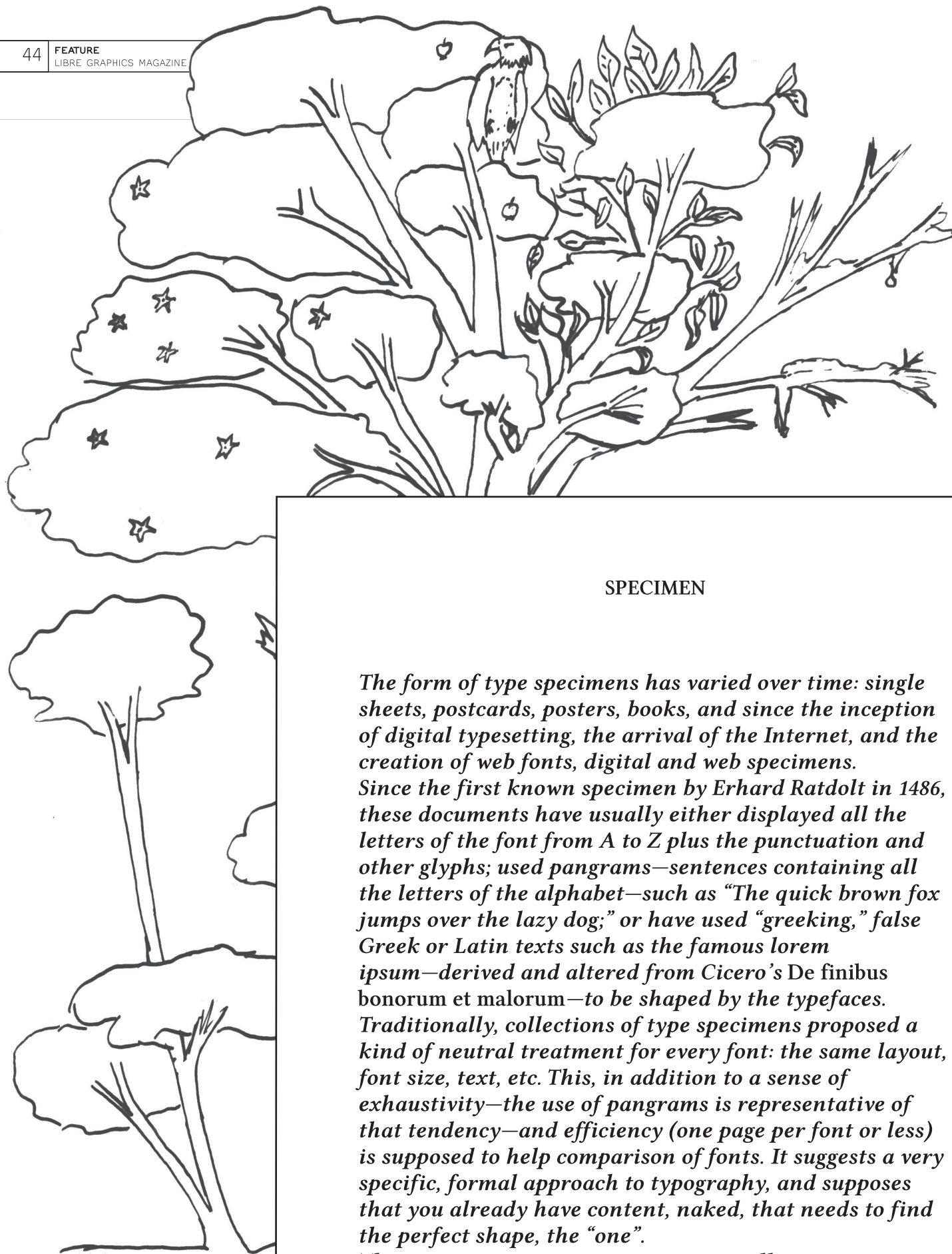
SPECIMEN

The form of type specimens has varied over time: single sheets, postcards, posters, books, and since the inception of digital typesetting, the arrival of the Internet, and the creation of web fonts, digital and web specimens.

Since the first known specimen by Erhard Ratdolt in 1486, these documents have usually either displayed all the letters of the font from A to Z plus the punctuation and other glyphs; used pangrams—sentences containing all the letters of the alphabet—such as "The quick brown fox jumps over the lazy dog;" or have used "greeking," false Greek or Latin texts such as the famous lorem ipsum—derived and altered from Cicero's De finibus bonorum et malorum—*to be shaped by the typefaces. Traditionally, collections of type specimens proposed a kind of neutral treatment for every font: the same layout, font size, text, etc. This, in addition to a sense of exhaustivity—the use of pangrams is representative of that tendency—and efficiency (one page per font or less) is supposed to help comparison of fonts. It suggests a very specific, formal approach to typography, and supposes that you already have content, naked, that needs to find the perfect shape, the "one".*

The reason texts in specimens are usually uninteresting or even unreadable is because they aren't supposed to be read, but rather looked at. They are models, wearing the type until they are replaced by the "real content." Like the contestants in a beauty contest, all the typefaces are forced into the same swimming suits before it is announced: "let the most beautiful win!"

This article was laid out by the author, Loraine Furter.

Today, since designers mostly use digitized and digital fonts, distributed through websites, specimens are usually found online. The tendency of contemporary web platforms to separate form from content (in which the content is stored in a database and pushed through a template) aligns perfectly with this aforementioned way of showing typefaces. We find this approach almost without exception in the foundries' websites and even more in the web font directories, where it is unfortunately rare to find something about the (hi)story of the font: why and how it was made, what were the references for the work, etc.
The informative text on a font is usually after the following categories:
Family (the name of the typeface)
Category (Serif, Sans Serif, …)
Designer
License
Full Language Support
Description / Information

I've considered making a cut-up out of different description texts from font directories to give a sense of that kind of literature, and try to construct the most unspecific text ever written. But in fact these texts don't need to be edited to speak by themselves:

"*** is the perfect font for body text and headlines on a website. Its modern style, suited with past characteristics of great typefaces, make it highly readable in any context. The full-circle curves on many characters make *** a great font to blend seamlessly with other fonts while still maintaining it's uniqueness.
Whether to be used for body text or headlines on a web page, *** is the right font for any project."

The search engines of big font directories sort through the names of the fonts, their physical characteristics, whether they have serifs or not, width, thickness, popularity, but never through their "stories".
And that isn't the only kind of literature that gets left out.

A digital font is made of programmatic instructions, visually interpreted and rendered by the interfaces used to create, read and print them. This dimension usually remains in the backstage of the computer.
Funnily, an exception is when it comes to legal issues. Legally, the creative status of a font is not so clear, and in a way this is not really the main point of interest of this text. But it is interesting to note that in US law, fonts are considered as utilitarian objects and are thus exempt from copyright restrictions. And as computer programs are under copyright, many foundries consider their fonts as such in order to be subject to the same legal protections.
This is taken very seriously by digital type foundries. One just has to look at their license agreements: in these sections, suddenly, the word font is associated with the word software in every sentence. Words that disappear once one gets back to other sections. Despite the legal importance of code, and even if it is today part of every font "body", little attention is paid to the code behind fonts.

In the context of type, using the term software means considering the fonts as a set of programmatic instructions. Basically, coordinates and bézier curve formulas.
Like this :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<glyph name="A" format="1">
    <advance width="720"/>
    <unicode hex="0041"/>
    <outline>
      <contour>
        <point x="383" y="229" type="move" name="top"/>
      </contour>
      <contour>
        <point x="715" y="0" type="line"/>
        <point x="715" y="13" type="line"/>
        <point x="703" y="13" type="line" smooth="yes"/>
        <point x="645" y="13"/>
        <point x="609" y="67"/>
        <point x="588" y="121" type="curve" smooth="yes"/>
        <point x="369" y="677" type="line"/>
        <point x="366" y="677" type="line"/>
        <point x="128" y="150" type="line" smooth="yes"/>
        <point x="110" y="111"/>
        <point x="69" y="13"/>
        <point x="13" y="13" type="curve" smooth="yes"/>
        <point x="5" y="13" type="line"/>
        <point x="5" y="0" type="line"/>
        <point x="226" y="0" type="line"/>
        <point x="226" y="13" type="line"/>
        <point x="218" y="13" type="line" smooth="yes"/>
        <point x="159" y="13"/>
        <point x="141" y="56"/>
        <point x="141" y="100" type="curve" smooth="yes"/>
        <point x="141" y="127"/>
        <point x="149" y="154"/>
        <point x="157" y="173" type="curve" smooth="yes"/>
        <point x="197" y="263" type="line"/>
        <point x="430" y="263" type="line"/>
        <point x="442" y="233" type="line" smooth="yes"/>
        <point x="469" y="164"/>
        <point x="488" y="111"/>
        <point x="488" y="74" type="curve" smooth="yes"/>
        <point x="488" y="34"/>
        <point x="465" y="13"/>
        <point x="407" y="13" type="curve" smooth="yes"/>
        <point x="395" y="13" type="line"/>
        <point x="395" y="0" type="line"/>
      </contour>
      <contour>
        <point x="322" y="538" type="line"/>
        <point x="422" y="283" type="line"/>
        <point x="206" y="283" type="line"/>
      </contour>
    </outline>
</glyph>
```
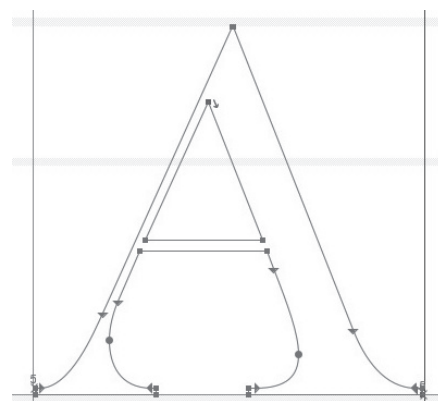
In case you didn't recognize it, this was the code of the "A.glif" file from the Open Baskerville font.
Let's not be afraid of it.
If you read it, this code describes the drawing of a classical digital font: built as outlines. The points are coordinates for the contour of the letter.

Technically, the code of every typeface is public, as they can all be opened in font editing software and reveal the position of their points and curves. But unlike software, the interest and originality of a font usually still remains the same: its shape. Even though there is a growing interest in programmatic fonts, fonts today are hardly ever designed by writing code, they are instead drawn in visual interfaces, and in the end it is their visual form that is read and used. Rather than viewing these approaches separate from fonts, wouldn't it be nice to

consider all these aspects–that is the visual aspects, as well as the code and the history of a font–as a whole?
The F/LOSS approach seems to offer a good framework for such a proposition.

In his essay *Take Care*, art critic Anthony Huberman speaks about "thankful" behaviour for institutions and curators, inviting them to perform their job in the key of the "I Care" (borrowed to Jan Vervoert's essay Exhaustion and Exuberance): paying homage to who or what they work with, taking the risk of performing both the "I Know" and "I Don't".

F/LOSS approach makes me feel something similar. There, people are invited to perform in the key of the "I Share," sharing one's work and making it possible for others to appropriate it, enrich it, enabling a dialogue. And more importantly sharing one's sources, in the sense of recipes and instructions (code), but also sources in the sense of references and inscription in a (hi)story.
It's about influence, as a positive dynamic of circulation and transmission.
F/LOSS approaches emphasize the evolution, the history and narrative of projects, with frameworks and tools that facilitate the documentation of a project and show the lineage between projects.
Soon, with versioning systems such as Git, we will be able to visualize the evolution of a typeface, from 0.0 to 1.0 and beyond.
No type design comes completely out of nowhere. In fact, it's more the concept of originality or uniqueness that comes out of nowhere. Copy, re-interpretation of typefaces has always been part of the history of typography[1], and still today old metal typefaces need to be translated into digital formats to be used on new mediums.

The lines you've just read are composed in Linux Libertine, the body text font of *Libre Graphics* magazine, with Linux Biolinum [Fig.1] and `Prop Courier Sans`.
The multilingual Linux L^{iber}_{tine} was created in 2003 by Philipp H. Poll, and is inspired by 19th century book typefaces, such as Janson and Palatino, in turn inspired by Renaissance models. Its name is a very playful reference to the different meanings of "Open" and "Free". Not monogamist, it brings another metaphor to this whole specimen story, the family tree and its wild branches: bastards and other milkman's sons. [Fig.2]

It is very nice to be embrace the *voyeur* posture and observe the multiple relations between F/LOSS fonts.

```
Friendly fork of the Not Courier Sans
by OSP, Prop Courier Sans is "not here
to be correct".
To create this proportional version
of the Not Courier Sans, Manufactura
Independente developed a tool² called
transpacing, which transplants the
spacing information from a font to
another. With adaptations and regular
tweaks, this font evolves hands in
hands with Libre Graphics magazine.
Less polite and without softening
the edges, OSP took a cutting from
Nimbus Mono to grow Not Courier Sans
during the Libre Graphics Meeting in
2008, in Wroclaw.
```

Proposing another kind of tree, the colophon of the book Transparence Camouflage Opacité by Samuel Rivers-Moore highlights the lineage between the font he created for the book, Arcadia, and its "sources". Its graphical interpretation mixes the family tree and an indentation referring to programming languages.

Typographies : Linux Libertine (texte)
Philipp H. Poll, 2003
Free Open Font License

Arcadia (titres, notes et légendes)
Samuel Rivers-Moore, 2012
Free Open Font License

↦ Fork de la typographie *Limousine*
  (Open Source Publishing, Free OFL, 2008)
↦ ↦ Fork de la typographie *Free Sans*
    (Free Software Foundation, GNU General Public License, 2003)
↦ ↦ ↦ Inspirée de la typographie *Nimbus Sans*
      Dessinée en 1985 par Max Miedinger (URW++, 1999)
↦ ↦ ↦ ↦ Inspirée de la typographie *Helvetica*
        Dessinée en 1957 par Max Miedinger (Haas / Linotype)

And there again we find Libertine cooking up with OSP in a Limousine!

This article is a call for more attention to the (hi)stories in (F/LOSS) fonts projects.

[Fig.1]

Linux Biolinum E13D
Specimen of the Linux Biolinum, a humanist sans serif (in the step family of Optima), from the species of the *organo-grotesque.* Its cousin Linux Biolinum Keyboard is not really readable in sentences.

[Fig.2]

In 2008, OSP did a remix of the Linux Libertine, "Libertinage," with variations for every letter of the alphabet and corresponding font Libertinage-a, Libertinage-b, through Libertinage-z. (This paragraph uses a different Libertinage font variant for every word.)

NOTES

1. On this specific question, read the article "Appropriation and Type – Before and Today" by Ricardo Lafuente, 2008.
2. Wouldn't it be nice to do a specimen for a font program?

SPECIMEN

This article is a special version of the "about" of a web project called *SPECIMEN* – a blog on F/LOSS fonts: http://specimen.meteor.com. *SPECIMEN* is an attempt to propose a different approach to (choosing) typefaces, another way of considering specimens. Far from pretending to be neutral or exhaustive, it rather assumes its partial aspect. Each font thus has its own specific treatment, with its own content and shape, all contributing to tell its story: the (hi)story of the font is re-placed at the center of the specimen.
This project is inspired by the *Open Source Font Specimens*, by Greyscale Press. Greyscale Press released the specimen book *L'Ève future – Spécimens* de fontes libres, collection of font specimens for usage in print, with F/LOSS fonts. Created during a workshop, this specimen book uses as sample text the whole science-fiction novel *L'Ève future*, by Auguste de Villiers de L'Isle-Adam, published in 1886 and thus in the public domain.
It's a great project that made me want to go further in the questioning of the still rigid production and use of specimens today.

REFERENCES

On appropriation:
Ricardo Lafuente, "Appropriation and Type—Before and Today", 2008.
http://ospublish.constantvzw.org/blog/typo/appropriation-and-type-before-and-today
"You Need to Copy to Understand, interview with Harrisson", OSP-Blog, 2006.
http://ospublish.constantvzw.org/blog/typo/you-need-to-copy-to-understand
http://i.liketightpants.net/and/no-one-starts-from-scratch-type-design-and-the-logic-of-the-fork
On specimen:
Nick Sherman, "Le design des spécimens typographiques", The Shelf n°1, 2012.
L'Ève future – Spécimens de fontes libres, Greyscale Press, 2013.

Resources/Glossary

Apache license
A permissive license, normally used for software, developed by the Apache Software Foundation.

Ascii
An early standard used for the encoding of characters. Notable for its small character set, which can be contrasted with Unicode.

Bézier curve
A type of mathematical curve used extensively in vector graphics.

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

C
A popular, early programming language, still widely used, especially in high performance applications.

Cache
A temporary, local copy of a file or files, often used to speed up loading times for websites.

CAPTCHA
System for differentiating humans from computers, through the ability to recognize sets of random, distorted characters. Commonly used for SPAM prevention.

Cms
Stands for Content Management System. Software installed on a server in order to provide a simple framework for editing web pages. WordPress and Drupal are examples of content management systems.

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

Database
Read-write venue for storing information, applied broadly in computation.

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

FontForge
A F/LOSS font editor for GNU/Linux, Mac OS X and Microsoft Windows.

Free
As in freedom, or often, that which is or is of Free Software.

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.

Front end/back end
A distinction in web development between code which runs on the computer of the person viewing a given site (front end) and on the server which hosts the site (back end).

Geomerative
Library for Processing, with capacity for vector graphics and typography.

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

Git
A popular version control system, originally created to manage development of the Linux kernel.

Gnu/Linux
A group of operating systems which are built on the Linux kernel and components from the GNU project, among others, which are widely distributed and freely modifiable.

Java
A programming language and platform developed by Sun Microsystems, intended for nearly universal compatibility with a variety of devices.

JavaScript
A scripting language commonly used on websites.

Kdenlive
F/LOSS video editor, available for GNU/Linux, Mac OS X, FreeBSD.

Library
An encapsulated set of functions intended to be used in the development of larger programs. Often used to make common tasks easier to implement.

Libre
A less ambiguous adaptation of the word Free. Implies liberty of use, modification and distribution.

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

Open source
See F/LOSS.

Open standards
A standard which is available for viewing and implementation by any party, often at no monetary cost.

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.

Programming language
An artificial language with a restricted syntax, used as an intermediary between computers and human programmers.

Proprietary
A piece of software or other work which does not make available its source, which is not allowed or intended to be modified or redistributed without permission.

Public domain
The legal status of a creative work for which the copyright (or other rights restriction) has expired. A work in the public domain can be used by anyone, for any purpose, without restriction. Licenses such as the Creative Commons CC0 license emulate public domain.

Pure Data
A visual programming environment designed for the production of interactive multimedia and audio works. Available for GNU/Linux, Max OS X, Microsoft Windows, iOS and Android.

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

Raster
A method of images which makes use of pixels of colour. Treating an image like a grid of squares, pixels are laid out to form shapes.

Ruby
A modern scripting language commonly used in web development.

Ruby on Rails
A popular framework, written in Ruby, used to make web applications.

Scalable Vector Graphics (SVG)
A standard for vector graphics, developed by the W3C.

Server
A computer hosting data which is accessed remotely.

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.

Synfig
Timeline-based animation software supporting both vectors and rasters. Available for GNU/Linux, Mac OS X and Windows.

Template
In software or content management terms, a pre-made set of styles which can be applied on-demand to content.

TeX
A code-based typesetting system.

TrueType
A common vector-based font format.

UFO (Unified Font Object)
A cross-platform font format, based on XML. Intended to be human-readable.

Unicode
A standard used for the encoding of characters. The term is often used to refer to the set of characters defined by the standard.

Version control
A means of managing changes (and allowing reversion) to a commonly held body of work, most often a software project.

Web font
A font housed on a server which can be dynamically called up for use on a web page.

Web-scraping
The use of automated tools such as scripts to extract text and other information from web pages with minimal human intervention.

XPM (XPixMap)
An image file format, which uses ASCII characters and a defined palette to describe a raster image.