

The Font Engineering Toolkit

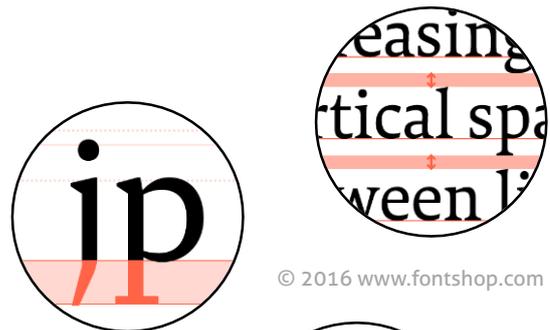
Live Font Creation in a Self-supporting Programming Environment

Bachelor Project Proposal 2016/2017
Software Architecture Group
Prof. Dr. Robert Hirschfeld

Software Tools Support Font Design

Font designers rely on tool support to create and modify fonts. These tools are not just mere image processors but highly customizable, domain-specific software that support designers to specify requirements, draw glyphs, and version intermediate results. Examples include Glyphs¹, FontLab², FontForge³, RoboFont⁴, or TruFont⁵.

When designers want to create new Unicode fonts, there are thousands of characters to cover. Hence, such huge fonts can hardly be designed and implemented glyph-by-glyph from scratch, but the whole process is supported by automation, derivation, reuse, and contextual visualization.



© 2016 www.fontshop.com

Towards Open Font Design

Many existing tools do not support a guided font engineering process. They tend to be monetarily expensive and entail a steep learning curve. Plugins and scripts are often free and complement the paid product extensively in small portions. There is also a strong focus on Latin writing systems. For example, there is typically only rudimentary support for Arabic, Japanese, Devanagari, or Burmese. Yet, the worlds demand for fonts that support such complex writing systems is bigger than ever.



There are several novel approaches to font design, which try to remedy certain aspects of the font creation process, such as Metapolator⁶, Glyphr Studio⁷, and Prototipo⁸. Yet, there is no open platform that supports the whole font production pipeline in a guided fashion.

¹ Glyphs: <https://glyphsapp.com>

² FontLab: <http://www.fontlab.com>

³ FontForge: <https://fontforge.github.io>

⁴ RoboFont: <http://doc.robofont.com>

⁵ TruFont: <http://trufont.github.io>

⁶ Metapolator: <http://metapolator.com>

⁷ Glyphr Studio: <http://glyphrstudio.com>

⁸ Prototipo: <https://www.prototipo.io>

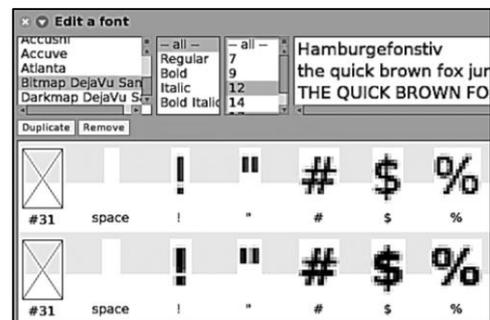
In this project, we want to address the following challenges:

- Support a guided, agile font engineering process with short feedback loops
- Support the creation of comprehensive Unicode fonts
- Support a community-driven approach to font engineering

The Font Engineering Toolkit

From a technical perspective, we want to employ the strengths of a live programming system to support live font engineering. In Squeak/Smalltalk⁹, programmers are in control of the full technology stack and are, in principle, able to use the font they currently design in their accompanying tools. Such a working mode fosters short feedback loops and supports programmers and font designers to discover design issues early in the process.

The Morphic framework in Squeak supports rapid prototyping of custom tools to help solve specific font design issues. If the programming interface is simple enough, font designers, which do not necessarily share a deep programming background, might be enabled to shape their own interactive tools according to their needs.



It is important that the designers do not have to re-invent common workflow patterns in font engineering for every new project. This is, unfortunately, the case in way too many projects at the moment. A tool's scripts and plugins help support smaller steps, but are quite decoupled from the underlying process.

The whole platform is to be integrated into the existing landscape of font engineering tools and their data exchange formats. For example, the *Unified Font Object* (UFO) format stores glyph data and font features. See FontShop's Glossary¹⁰ and the Typo-Wiki¹¹ for more information about the terminology.

Functional requirements will be discussed continuously with our project partner — maybe also with more font designers from the Berlin area — and may encompass:

- Build a Smalltalk object representation for glyphs and font features
- Advance, fix, or replace Squeak's font rendering system
- Design and implement several interactive tools that help font designers
 - Begin, pause, and resume font engineering sessions
 - Derive glyphs from existing glyphs
 - Render glyphs in a textual context to improve kerning and other properties
 - Design families of fonts to be used side-by-side, even a mixture of different writing systems
- Support well-known data exchange formats such as UFO and OpenType, even with incomplete font data

⁹ The Squeak/Smalltalk Programming System: <http://www.squeak.org>

¹⁰ FontShop Glossary: <https://www.fontshop.com/glossary>

¹¹ The Typo-Wiki: <http://www.typografie.info/3/wiki.html>

Implementation Details

The implementation of the project will involve programming with the Squeak/Smalltalk programming system and the Morphic framework. It will also involve learning about the Open Smalltalk Virtual Machine (VM) and its plugin architecture to employ lower-level functionality and foreign libraries in the C programming language. An agile, iterative process will be employed for software development. All results will be published under the MIT license¹².

Organization

A group of about six to eight (6–8) students may participate in the project. Organization and tasks will be determined by the project participants. The project will be carried out at the Hasso Plattner Institute in Potsdam. Project participants are expected to communicate with our partner via the GitHub issues/wiki, email, chat, or voice on a regular basis. Communication is likely to be conducted in English and German. In the winter term 2016/17, participants will work on exploring the domain and creating first prototypes. Main steps in design and implementation of a solution are to be executed in the summer term 2017. Expected results include a working software accompanied by appropriate documentation. There is an option to meet other font designers in Berlin to complement the requirements and collect real user feedback.

Partner & Contract

Lasse Fister
graphicore, Fürth
<http://graphicore.de>, lasse@graphicore.de

Prof. Dr. Robert Hirschfeld, Tobias Pape, Tim Felgentreff, Marcel Taeumel
Software Architecture Group, Hasso Plattner Institute, Potsdam
<http://www.hpi.uni-potsdam.de/swa>, hirschfeld@hpi.uni-potsdam.de

¹² The MIT License: <https://opensource.org/licenses/MIT>