Truetypewriter PolyglOTT: Your multilingual typewriter assistant

Sergey Beatoff

Tyapkin Sergey, fl 68,
pereulok Shovkunenko 14,
Odessa,
Ukraine 65009
Email: sergeybeatoff at gmail dot com

This article is not meant to be a description of any kind of scientific research. It is merely the history of how I came to like font design, what difficulties I encountered and how I found solutions. It is about creating my font Truetypewriter PolyglOTT. I describe my long way from the idea of creating the font to its different implementations, with version 3.76 as of October 2018. In the beginning, I mistakenly thought there were several fonts simulating true typewriter printing; however, after checking out, I noticed that nearly all of them included only the Latin part. Besides, they merely accentuated the defects and inaccuracies of typewriters' text trying to make the fonts look too much realistic. In the article, I am also discussing the creation of computer fonts in general.

The beginnings

My given name is Sergey, but many of my friends call me Sam. I live in Odessa, Ukraine and work as a musical editor for an FM radio station. Creating computer fonts is one of my hobbies. As a matter of fact, as long as I remember, I have been interested in font design. When I was about six, I used to draw not only cars and wars, as all boys of my age, but also some unusual beautified letter variants. However, I was not thinking about turning this into a profession. Nevertheless, at university I studied philology, which is not a science about fonts at large, but generally about language!

My first serious encounter with fonts happened in the mid 1990s. My wife was teaching English in an Odessa gymnasium. From time to time, she would have to make visuals and handouts. Once she needed to make transcription signs that would look printed, not hand written. Nowadays, in Unicode there are blocks of IPA Extensions and many fonts, including those which go together with different operating systems, have these blocks filled. However, at that time, not only did Unicode fonts

not exist, but the standard itself was not supported. So, after a fruitless search, I decided to create myself one font for her. In addition, I found among my floppy discs a program called Altsys Fontografer and tried to use it. Of course, I did not get everything at once, but the general principles were clear. So, I chose a standard Times New Roman font, saved it under a different name and designed the necessary symbols from those which looked like the ones my wife and I needed. Some of my creations appeared a little clumsy, but it was noticeable only when zooming in. Moreover, having a unique encoding, the font was, naturally, absolutely incompatible. Nonetheless, it proved handy and my wife used it for several years.

Serious work

Sometime later, I got interested in working with fonts seriously. For the next several years, in my free time, I studied both font creating software and how fonts are used. At the same time, there was going on a gradual transition to Unicode fonts and also OpenType came into play. Afterwards, I started creating my own fonts—not clumsy and crooked anymore, but proper, correct fonts without mistakes and in standard encodings.

At present, the most successful and complete of my projects is Truetypewriter PolyglOTT. It might be interesting to mention that the idea about this font came up accidentally. In the company where I was working, I once saw a designer scanning a typewritten text and pasting it in a project of his. I wondered why he did not use a computer font imitating a typewriter (there already existed such fonts). He replied that this was exactly his idea at first. Unfortunately, it appeared there was a lack of fonts supporting the Cyrillic alphabet—there were only a couple or so of such fonts, and those did not include all the necessary letters. Another problem was that the fonts imitated an old, dirty, wrecked-looking typewriter outcome, whereas he needed an ordinary typewritten text; not with the perfectly even contours as is characteristic of most monospace computer fonts, but also not with such dirty and crooked letters as were found in existing typewriter simulation fonts.

After that, I decided to design a font meeting the necessary demands. Understanding the time required for creating a new font, I realized I could not make the font in time for that project. However, I was thrilled with the idea.

First, I planned to limit my creation to the set of WGL4 glyphs that were standard for the fonts provided with the operating system. In order not to leave the project nameless, I assigned it the working title "Truetypewriter," which was a blend of the two words *TrueType* and *typewriter*. It was to be just a temporary name. The title was so obvious, that I thought to find a lot of fonts with the same name. I was going to exchange it for something better when I would come up with another name. Strangely enough, I have not found any fonts with the same name, or at least I have not come across any mentioning of such. As a result, I left the name as is: "Truetypewriter WGL4." After, I decided to expand the limits of WGL4, I came up with the heading

PolyglOTT—a blend of the word *polyglot* with the abbreviation OTT for Open TrueType, i.e., TrueType font with OpenType extensions. I have now added this to my other multilingual fonts with OpenType support.

I started with the Cyrillic part, first, because it was most familiar, and secondly because at that time typewriters were still in use, although not as much anymore. I scanned letters in different resolutions and made tracings, after which I saved the font and printed it in different type sizes to evaluate the result. Gradually, I started to understand defects in vector view characteristics of a typewritten text; which of them to retain in order to make the font look natural, which to ignore and which to eliminate so as not to overload the glyph contours with an excessive number of control points.

At the end of the 19th century and the beginning of the 20th century, the appearance of typewriter letters of different manufacturers differed appreciably. However, by the mid 20th century, the overwhelming majority of Cyrillic typewriters were manufactured in the USSR, the German Democratic Republic (GDR), Czechoslovakia and Bulgaria. Nearly all shapes of corresponding typebar letters did not differ, although the typewriters were produced by different plants and under different names (one such typewriter is shown on Figure 1). Thus, *socialist standardization* (socialist countries were characterized by attempting to standardize everything to the utmost) gradually led me to the idea not to blindly copy the typebar letter forms (letters on the type hammers, and more precisely, their imprints on paper) of this or that manufacturer, but to choose the most standard variants ignoring even serious differences if they were characteristic of only one typewriter manufacturer.

Moving on to the Latin part of the font, I understood that, even if I had not come with the idea described above for working with the Cyrillic part, I would be compelled to do so. However, while Cyrillic typewriters were readily available, giving the opportunity to type and print out necessary symbols and experiment with them, typewriters with Latin fonts were rare in the USSR—and consequently, later in the countries of the former USSR. Thus I had to be content with previously scanned documents. The lack of samples led me to the conclusion that I should not limit myself to the WGL4 set, but to make other letters from corresponding codepages, even if samples to refer to were non existent. Besides, I wanted to construct even those letters which were never used in typewriters. On the whole, the glyph structure is clear. One can construct them taking parts of different glyphs and compose them together.

It should be specified, this is not an easy task. You cannot simply copy the samples presented on the Unicode site [1] and build glyphs according "to their image and similarity." First, on the Unicode site, you can find only one variant for every table cell, whereas in many languages the same letter can be written differently, though often with some similarity. Apart from that, many font designers make errors when they use some external similarity they had noticed. For instance, several times I came across fonts where the Cyrillic small letter м ('em': `U+043C`) was written as an upside-down small Latin 'w.' Yes, the obtained letter looks recognizable, especially in context.

Figure 1: Yatran PEK-305-05 (Ятрань ПЭК-305-05) electric typewriter. It was one of the widely used typewriters in the USSR from the 1970s until the 1990s. During that period, up to 140,000 Yatrans were produced per year at the Drukmash plant in Kirovohrad (or Kirovograd, now Kropyvnytskyi), Ukraine.

However, in majority of Cyrillic fonts it should look like a smaller Cyrillic capital letter M ('Em': U+041C). So, in every specific case I tried my best to learn the history of every letter in order to understand the way of thinking of the designers who made initial variants of typebar letters.

The Greek part appeared to be all Greek to me, indeed! By the time I was getting the first font version ready for publication I had found only one image of a Greek typewritten text on the Internet, in which not all the letters of the Greek alphabet were present. Using as a sample the font Courier New and some other analogous fonts, I tried to construct the letters which were illegible or absent on the image that I had found. This time *the system failed*: in spite of the fact that the letters were quite recognizable and readable, some of them—as it was clarified later—looked different on typewriters. I, so to speak, did not manage to fully understand the way of thinking of the master-producer of the typebar letters; I could not step into his shoes. That is the reason why in the current version, the Greek part was considerably remade thanks to a friend who provided samples printed on an Olivetti Lettera 35 with a Greek keyboard (see sample on Figure 2). That friend also explained to me how Greek typewriters were used: e.g., the combinations for accented vowels, etc. Having this experience in mind, what I know now, for sure, contrasts with what is believed; Courier New does not resemble a typewriter font!

Apart from this, since the first publication, there have been some minor corrections in Hebrew and Georgian. The corrections are included in the current version as well.

αβγδεζηθικλμνξοπρσcςτυφχψω ϙϻϝϳϛ
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣC ΤΥΦΧΨΩ ϘϺFJϚ

Ἔφη γὰρ αὐτοὺς τοὺς Ῥωμαίους αἰτίους εἶναι τοῦ μὴ πειθαρχεῖν αὐτοῖς τοὺς Ἕλληνας, ἀλλὰ παρακούειν καὶ τῶν γραφομένων καὶ τῶν παραγγελλομένων. δυεῖν γὰρ οὐσῶν αἱρέσεων κατὰ τὸ παρὸν ἐν πάσαις ταῖς δημοκρατικαῖς πολιτείαις, καὶ τῶν μὲν φασκόντων δεῖν ἀκολουθεῖν τοῖς γραφομένοις ὑπὸ Ῥωμαίων καὶ μήτε νόμον μήτε στήλην μήτ' ἄλλο μηθὲν προυργιαίτερον νομίζειν τῆς Ῥωμαίων προαιρέσεως, τῶν δὲ τοὺς νόμους προφερομένων καὶ τοὺς ὅρκους καὶ στήλας καὶ παρακαλούντων τὰ πλήθη μὴ ῥᾳδίως ταῦτα παραβαίνειν, ἀχαϊκωτέραν εἶναι παρὰ πολὺ ταύτην τὴν ὑπόθεσιν καὶ νικητικωτέραν ἐν τοῖς πολλοῖς. ἐξ οὗ τοῖς μὲν αἱρουμένοις τὰ Ῥωμαίων ἀδοξίαν συνεξακολουθεῖν παρὰ τοῖς ὄχλοις καὶ διαβολήν, τοῖς δ' ἀντιπράττουσιν τἀναντία.

Figure 2: A sample of the Truetyperwriter PolyglOTT font. Greek alphabet set at 14 pt with 18 pt leading. Text from Polybius' *Histories* (24.9) set at 10 pt with 14 pt leading.

Into the world

When the first public version, which was version 2.37 but not 1.00 (I enumerate every interim version not to get confused) was ready, I decided to make it accessible for everyone. The previous working versions were shared only with friends. I studied different sites dedicated to fonts and eventually chose FontSpace [2], where I uploaded my font. I singled out FontSpace due to the thoughtful organization, accurate layout and categorization of font material.

It should be pointed out, my font very quickly appeared among the leaders according to the number of downloads, which was a pleasant surprise. This was despite the fact that there are numerous fonts imitating typewriters; Truetypewriter PolyglOTT is but one of the many. It was quite pleasant to get emails with thanks, but no less pleasant (and even more useful) were the messages with critical comments. I was glad that it seemed I had not made any serious errors. I corrected discovered inaccuracies without delay, in a short time uploading a corrected version. Some of the comments I took into consideration and corrected the indicated inaccuracies, putting aside some others.

When I say "finished the work on the version," I exaggerate to some extent. During the work, new ideas kept popping up and I would keep brushing and adding. There is such a joke in Russian: "Repairs in a flat cannot be finished, repairs only can be stopped." This is also true about fonts: you set some limitations and arrange the

Figure 3: Composite images with Cherokee text (left) and Thai (right) text set in Truetypewriter PolyglOTT.

rest to match them. Some ideas that keep arriving are left for further realization in the next versions.

Besides polishing and correcting errors and lapses, many new features are added, including significantly increasing the number of supported languages (Figure 3). You can use Truetypewriter PolyglOTT now for Cherokee, Tifinagh, Lisu and Deseret; Glagolitic also (not the original, ancient so-called "rounded," which became obsolete even in ancient times but "angular" that was used until the mid 20th century (according to some data, it is still used for church services in Croatia and some other countries). Apart from this, in terms of experimenting, I added base support of some Indic scripts: Devanagari, Bengali, Thai, Lao and Khmer—I plan to add full support in further versions. Today, modern word processors include OpenType support, which really makes it possible to work on complex scripts. Whereas in 2012, when I published the first version of my font, full OpenType support hardly existed except in Adobe InDesign and Adobe Photoshop. I hope to hear feedback from native speakers of these languages which will make it clear which tasks prioritize.

Challenges

A separate paragraph is what I need to dedicate to the Arabic script.

There have been many typewriters created for the Arabic language and I have samples of the printed text. Initially, I did not want to add Arabic, as such a printed text looks greatly different comparing to the text printed in Indo-European languages. Moreover, all typewriters for European languages are monospaced, whereas typewriters for Arabic have three different letter widths. Thus, a mere copying of typewritten Arabic letters would have destroyed the idea and concept of my font. At

المادة 1.
يولد جميع الناس أحراراً متساوين في الكرامة والحقوق، وقد وهبوا عقلاً وضميراً
وعليهم أن يعامل بعضهم بعضاً بروح الإخاء.

المادة 2.
لكل إنسان حق التمتع بكافة الحقوق والحريات الواردة في هذا الإعلان، دون أي تمييز،
كالتمييز بسبب العنصر أو اللون أو الجنس أو اللغة أو الدين أو الرأي السياسي أو
أي رأي آخر، أو الأصل الوطني أو الاجتماعي أو الثروة أو الميلاد أو أي وضع آخر.

Figure 4: Part of the Arabic version of the *Universal Declaration of Human Rights*, typeset with Truetypewriter PolyglOTT at 10 pt with 14 pt leading. (Original text from: `www.ohchr.org`.)

first glance at printed pages in different languages it should not be very noticeable that the text in one language cardinally differs from the others.

Fortunately, I was advised by one of the FontSpace users to look at the works of Nasri N. Khattar, who reformed the Arabic written language in the mid twentieth century. Nasri N. Khattar was awarded with the Nobel Prize for his work. It is even claimed that IBM produced typewriters based on his designs [3].

Anyway, I approached the Arabic society in my city and showed them samples of Khattar's work. Surprisingly, they criticized these samples and said they are hard to read. However, they approved of my attempt to create an Arabic font looking somehow similar to a European font. As a result, I tried to create a compromise variant retaining general letter proportions (monospacing in the first place) resulting with letter shapes which are not as dramatically different from classical standard Arabic as Khattar's (see sample on Figure 4). Once again, I hope to get some helpful feedback from users.

The increased number of symbols in the font, resulted in its size enlarging. On one hand, computational capabilities are growing and the font size is not as crucial as it was ten or even twenty to thirty years ago. However, I did not want it to be a "heavy load" and "sluggish." Thus, in the new version I attempted to carefully review the contours and get rid of excessive control points. Besides, I tried to use composite glyphs where possible. It is a usual practice, though. But I also tried to use composites even where they are not conventionally used. Fortunately, the font is *de facto* monospaced, i.e., nearly all glyphs have the same width, but the monospace feature is not the font's characteristic, which spares the necessity of kerning. A big number of control points saves the necessity of hinting. This allows for cutting down on some font resources.

Besides all the above, after the publication of the first version, I noticed one unpleasant thing. Everything seems to indicate that there is still no single standard about the place of diacritic marks. In different text editors, diacritic marks take different places when marks have zero advanced width (and marks should have zero

advanced width). So, in the new version I fully rejected conventional zero advanced width and used hair-spaced advanced width (one point) instead. It has not affected the font appearance (at least it is hardly noticeable for a human eye), but then all marks take their places, precisely, regardless of the word processor. Otherwise, the only possible decision would have been the use of OpenType features. First off, it would have made the tables heavier; secondly, there are still some applications without OpenType support (their number is going down, though).

I hope the new version will become more popular among users than the previous. I am going to continue my work and finally get to including all well-spread writing types except the hieroglyphical, as the latter have too many differences in appearance and structure. It is worth noting that learning about font systems of various languages is not only interesting as is, but enriches knowledge about world culture, history and other spheres, which I would not have touched upon without getting into font design!

Making and giving for free

Truetypewriter PolyglOTT is a free font, and I am not going to make it commercial. It is my belief that multilingual fonts should be accessed free, especially those for a plain text. Paid-for should be decorative accident fonts having unique designer solutions. Moreover, I believe, the current prices for commercial fonts are too high considering the simplicity of downloading them; fonts are not protected anyways (serial number and so on). The price of a font (or a font family) should not be equal to the price of a font editor. (Font editors or some word processors cost as much as three or four fonts.) Not all companies, not to speak of individuals, can afford such expenses. They have to choose among several affordable free fonts, which results in textual monotony. The current situation with fonts resembles the "copyright wars" in music recordings and in pop music in general. I believe, instead of lifting up prices and provoking the pirate spread of fonts, it would be sensible to organize the system of downloading any font the user likes free unless for a commercial use. If the user really likes the font, they will make reasonable donations.

Anyway, I am an amateur. There is a possibility of minor flaws in my font, which a professional would not have made. That being said, I have come across several errors and mistakes in the fonts designed by respected companies. Those errors are upon their designers' conscience.

My main working tool is still FontForge [4], and I again want to thank George Williams for such a powerful free software. I have tried numerous font editors, from Softy to FontLab. Frankly speaking, I have not yet found an ideal editor with all the necessary features. At least FontForge is free, notwithstanding some shortcomings such as a not very convenient interface. It is worth noting, that nearly all font editors are customized for working with Type1 fonts, whereas I prefer to work with

TrueType exclusively. I consider the TrueType platform more prospective. However, comparing these two platforms is beyond this article's topic.

Also I have been using a set of different font utilities from Microsoft Typography [5]. Unfortunately, most of them were updated long ago and some of them have become obsolete. For example, when Font Validator [6] checks the newest fonts shows many errors, because these are standard updates not known (!) to Font Validator. Also, I often use DTL OTMaster Light [7] to analyze fonts. By the look of things, this software is good as a font editor as well, but I did not have an opportunity to use a commercial version, I have only used the Light one. Presently, the High-Logic FontCreator [8] has got me quite interested. At first, this software lost with respect to its resources in comparison to other software, nowadays it has developed into a powerful editor with a very convenient and well-thought out interface. I am thinking about buying and using it. Until recently, one significant drawback was the lack of manual hinting (although several latest versions included autohinting). Today, at last, Microsoft opened free access to their Visual TrueType (VTT) [9]—earlier, you needed to overcome several obstacles in order to get it. So, if necessary, hinting can be made separately.

You can download the latest Truetypewriter PolyglOTT version here: `https://www.fontspace.com/sergey-beatoff-aka-sam-t/truetypewriter-polyglott`. If you have any additional question or criticism, feel free to contact me *via* email.

Acknowledgements

I would like to express my special gratitude to Hans and Kateryna Wolf and Miliena Golovina for their linguistic help and their general support.

References

[1] Unicode, Inc., "Unicode 11.0 Character Code Charts." Last updated: June 12, 2018. URL: `http://www.unicode.org/charts/`. (Accessed on Oct. 31, 2018.)

[2] font.io, "FontSpace." URL: `https://www.fontspace.com`. (Accessed on Oct. 31, 2018.)

[3] Y. Khoury Nammour, "Fighting illiteracy with typography." *Works That Work*, no. 6, 2015. URL: `https://worksthatwork.com/6/unified-arabic`. (Accessed on Oct. 31, 2018.)

[4] G. Williams, "FontForge." URL: `http://fontforge.github.io/en-US/`. (Accessed on Oct. 31, 2018.)

[5] M. Jacobs, N. Hart, and M. Wojciakowski, "Font tools list." Oct. 19, 2017. URL: https://docs.microsoft.com/en-us/typography/tools/. (Accessed on Oct. 31, 2018.)

[6] Microsoft, "Font Validator." URL: https://github.com/Microsoft/Font-Validator. (Accessed on Oct. 31, 2018.)

[7] Dutch Type Library, "DTL font tools: OTMaster." URL: https://www.fontmaster.nl/index.php/otmaster/. (Accessed on Oct. 31, 2018.)

[8] High-Logic B.V., "FontCreator 11.5." URL: https://www.high-logic.com/font-editor/fontcreator. (Accessed on Oct. 31, 2018.)

[9] M. Jacobs, P. Linnerud, and R. McKaughan, "Microsoft Visual TrueType (VTT)." Oct. 19, 2017. URL: https://docs.microsoft.com/en-us/typography/tools/vtt/. (Accessed on Oct. 31, 2018.)