

2016-07-01

Generative Sans A Generative Typography Project Creating a Complex Future Narrative that Opens the Possibilities of Type with a Versatility to be used Across a Wide Variety of Mediums

Leon Butler

Technological University Dublin, leon.butler@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/designpart>



Part of the [Art and Design Commons](#)

Recommended Citation

Butler, L. (2016) Generative Sans A generative typography project creating a complex future narrative that opens the possibilities of type with a versatility to be used across a wide variety of mediums. *Iterations / Design Research and Practice Review, Issue 3, 2016*

This Article is brought to you for free and open access by the Design and Printing at ARROW@TU Dublin. It has been accepted for inclusion in Academic Articles by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@tudublin.ie, arrow.admin@tudublin.ie, brian.widdis@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)

ITERATIONS

Design Research & Practice Review

Issue 3 | July 2016

ISSN 2009-8243



9 772009 824000

07 >



**INSTITUTE
DESIGNERS
IRELAND**

Generative Sans

A generative typography project creating a complex future narrative that opens the possibilities of type with a versatility to be used across a wide variety of mediums.

Generative typography is a relatively new area of type design that disrupts the norms and presents a new evolution of the processes that have developed. It sheds some of the tools defined in the last 30 years and allows it to form an evolution of the process. Within the production of this typeface I wanted to build on what had gone before and define a new aesthetic with a complexity and future narrative that would open the possibilities of type with a versatility to be used as a display type in publishing and branding projects with the ability to generate custom ligatures joined strings of text for titles, headers and logo forms.

Through Metafont, a description language used to define vector characters developed by Donald Knuth [Knuth, 1979] and other generative processes, it has long been theorized that programs could create new forms of type. Indeed, Hermann Zapf the creator of 'Palatino', 'Optima' and 'Zapfino' stated in 1968 that *'electronics will soon force its claims upon letterforms, and let us hope it will liberate us from the dust of the past'* [Rugglcs, 1983].

During my residency at the School of Visual Art in New York, I researched the past and current developments in generative typography. While interesting strides had been made, especially in the work of the highly acclaimed author, Donald E. Knuth, whose unhappiness with the production of *'The Art of Computer Programming'*, listed by the American Scientist as one of the twelve best physical science monographs of the 20th century, led to his development of the Metafont system which allowed for the generation of a new mathematical typography; using it to create a new typeface for mathematics which he called *'AMS Euler'* for the American

Mathematical Society (AMS). While these developments took place in the early 1980's, I felt the current offerings fell into two distinct categories. The first is extremely experimental and expanded and bent the forms in an interesting and inventive manner, however, legibility was compromised in pursuit of this invention, for example, the *'Alphabet Synthesis Machine'* created by Golan Levin with Jonathan Feinberg and Cassidy Curtis in 2002; the characters formed were not intended to be read but rather to convey a concept (Levin, 2006).

The second more common generative type was too shackled to the geometry of the existing letter forms. Erik van Blokland and Just van Rossum's typeface *'Beowolf'* developed in 1990 [Van Blokland, 1990], which is programmed with a randomizing algorithm that generates different outlines every time the font is used in different output applications, is an example of this and while it was my goal to create a legible typeface, it was clear that I would have to disrupt this initial geometry if it was to achieve the aesthetic I desired.

Donald Ervin Knuth. 1979. TEX and METAFONT: New Directions in Typesetting. American Mathematical Society, Boston, MA, USA.

Lynn Rugglcs. 1983. Letterform Design Systems. Report No. S'1,IN-G-83-971. Department of Computer Science, Stanford University.

Levin, Golan, J. Feinberg, and C. Curtis. "The Alphabet Synthesis Machine." (2006).

Van Blokland, Erik, and Just van Rossum. "Random code—the Beowolf random font." The PostScript Journal 3.1 (1990): 8-11.

Ahn, Yeohyun, and Ge Jin. "Type+ code ii: A code driven typography." Proceedings of the IEEE VIS Arts Program (VISAP) (2013).

Sullivan, Louis H. "The tall office building artistically considered." Lippincott's Magazine 57.3 (1896): 406.

Rappo, François, ed. Typeface as program: applied research and development in typography. ECAL/University of art and design, 2010.

To create this altered form I wrote a processing sketch which would hack an open source sans serif typeface replacing the vector data associated with the typeface and replace it with a randomized form of eclipses between 0 and 30 pixels (float $r = \text{random}(0, 30)$). This generated forms that could be manipulated and regenerated to produce a form that felt it would deliver the best available geometry while still retaining a strong sense of legibility. From here a vector PDF file was created and brought in to Adobe Illustrator for refinement. At this stage the ellipses were all individually available to edit and through a process of trial and error I found different alterations pushing and pulling these elements to achieve the most desirable geometry for use in Fontlab and the subsequent generative sketch. The next step in the process was to use the pathfinder to consolidate the form and create a closed vector path, scaled to a suitable size for use in the Fontlab application. Copying the forms from Adobe Illustrator they were placed them into Fontlab and any open paths were closed until the character was complete. This allowed me to export a beta version of the typeface which I labelled 'CircleType' and placed it in the data folder of the generative sketch.

The generative sketch in processing was designed to randomize the connection of a set number of points available in a defined distance. 'Geomerative' library by Ricard Marxer [Ahn, 2013] allowed for the breakdown of the typeforms previously generated and exported in true type format into a series of paths, handles and point that the sketch would use to construct the form. First, passing single letters strings through the sketch a number of times to create variations of the form through a series of randomized connections, further refinements were made to the original letter forms in Fontlab and/or Adobe Illustrator depending on the severity of the changes needed. Once the alphabet was complete, a layout of the letterform was placed in order to establish balance, evenness of form and legibility. Further refinements and alterations were suggested, discussed and implemented with over fifty versions of each form being created for consideration of weight, form, legibility and suitability to the other letterforms in the family. The forms included in the final sample reflect the overall nature of the



Just Van Rossum and ErikVanBlokland:FFBeowolf 1990



Circle Type

typeface but these forms are ever evolving and should never be considered as a concrete form, instead existing in a plastic state, both malleable and reactive. The American architect, Louis Sullivan, for whom Frank Lloyd Wright served as assistant for many years said that *'Form follows function'*. In this case it is the actual 'function' that creates the form (Sullivan, 1896).

In the creation of longer strings of text, characters would form connecting ligatures. This provided an interesting extra element to the generative nature of the text but also meant alterations to the spacing of the characters in CirleType. By working on the spacings in Fontlab and generating new variations of the typeface I was able to control the extent the characters would interact with each other and how the ligatures would form and it produced excellent results.

The book *'ECAL: Typeface as Program'* by Peter Bilak, Jürg Lehni and Erik Spiekermann starts with the simple question *'is there such a thing as a computer program capable of taking over the routine tasks of letter design'* [Rappo, 2010]? While my project does not completely replace the routine tasks of letter design it arrives at a new form and aesthetic that is legible, progressive, utilises the power of the computer to create something that would not be conceivable using traditional digital techniques while still remaining in the control of the designer. Typography is based on the premise of embedded experiences *'with every typeface possessing a narrative and its own voice and I feel I have created a typeface with a unique voice and one that defines a new narrative.'*

GENERATIVE





Generative SANS

A font specimen for 'Generative SANS'. The text is displayed on a white sheet of paper held by two black binder clips at the top. The font is a sans-serif typeface where each character is constructed from a complex, interconnected network of thin black lines, creating a mesh-like or wireframe appearance. The characters are arranged in four rows: the first row contains the word 'GENERATIVE', the second row contains 'SANS', the third row contains the uppercase alphabet 'A B C D E F G H I J K L M N O P Q R S T U V W X Y Z', and the fourth row contains the digits '0 1 2 3 4 5 6 7 8 9'. Each row is separated by a thin horizontal line.

SANS

A B C D E F G H
I J K L M N O P Q
R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9