More than Words: Fonts as Generative Art

Prof. Erik Demaine **Martin Demaine**
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology, Cambridge, MA, USA
https://erikdemaine.org/fonts/
e-mail: demaine@mit.edu

Abstract

Artistic fonts offer a way to transform any text into an art print, interactive software, or animation. Thus we can view artistic fonts as a form of generative art: whereas most generative art is determined by an initial random seed, with fonts the "seed" is arbitrary text. An advantage of this approach is that the resulting print can encode one or more messages. With puzzle fonts, where the letters are encoded but not explicit in the glyphs, this message can be hidden within the resulting artwork.

We demonstrate this approach to generative art using a series of mathematical and puzzle fonts we have designed over the years, which are available as free interactive web apps from https://erikdemaine.org/fonts.

Figure 1 and Figure 2 show two examples of art prints designed with two of our fonts: one font that reveals itself when folding translucent paper in half and another based on glass cane from the art of glass blowing.

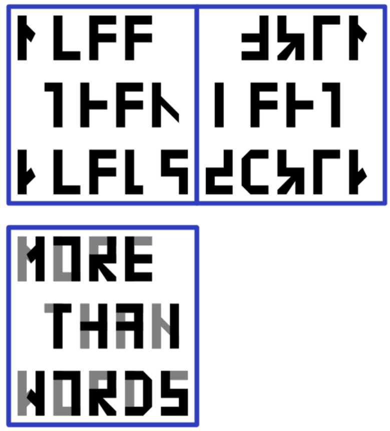

Figure 1: One-fold silhouette font applied to the text "MORE THAN WORDS". Top: Art print hiding the text. Bottom: Message revealed from folding in half.

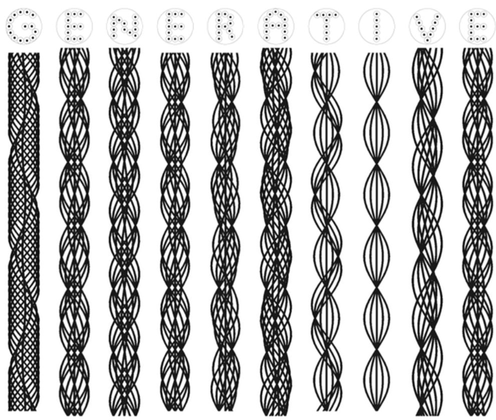

Figure 2: Glass cane font applied to the text "GENERATIVE". Top: Cross-section which reveals the letters. Bottom: Side view resulting from rotating cross-section at constant speed.

1. Introduction

Abstractly, ***generative art*** can be seen as an algorithm for transforming a random seed into artwork. Similarly, a ***font*** can be seen as an algorithm for transforming text into a print. If we focus on ***artistic fonts*** where each glyph is itself an artwork, text becomes an algorithmic way to compose these artworks into a larger artwork. In this way, artistic fonts can be seen as a kind of generative art, where the "seed" is a piece of text.

We can take this idea one step further by combining a font with an algorithm for generating the text, of which there are many; see, e.g., [1], [2]. Together, these two algorithms transform a random seed into a print in a "doubly generative" process: first generating text, and then generating the rendering of that text in the font.

In this paper, we focus on the latter stage: *artistic ways to render text*. We give several examples of generative art produced using a series of mathematical and puzzle fonts we've been developing over the years [3]. These fonts are freely available as interactive web apps, where the user can enter text and the software dynamically generates the corresponding rendering. We encourage the reader to experiment with making their own art using these fonts.

By using a ***puzzle font*** where reading the text requires solving a puzzle, or other obscuring fonts where the text is difficult to read, we can generate artwork that does not directly reveal text (unlike the traditional view of fonts), while still hiding that text implicitly within the artwork for an additional layer of meaning. We show several designs following this approach.

2. Voronoi Diagrams

Our Voronoi fonts [4] enable the creation of planar tilings, both regular and irregular, which in turn can be transformed into stained glass or floor/wall tile designs. Figure 3 through Figure 12 show some new designs we constructed based on this approach; the reader is encouraged to experiment themselves.

The Voronoi fonts are based on Voronoi diagrams, which go back to Descartes in 1644. Imagine growing several circles simultaneously and at the same speed from several different center points called *seeds*. The ***Voronoi diagram*** consists of the straight edges where these circles first meet each other. These ***Voronoi edges*** divide the plane into one ***Voronoi cell*** per site, containing all the points that are closer to that site than to all other sites. Some of these cells (near the boundary) extend to infinity.

We constructed two Voronoi typefaces (families of fonts); refer to Figure 3. The normal ***Voronoi typeface*** arranges sites so that the *Voronoi diagram* — specifically, the Voronoi edges that do not extend to infinity — draw the letter. The ***inverse typeface*** arranges the *sites* in the shape of the letter. Thus, in the Voronoi typeface, the sites do not look like the letter, while in the inverse typeface, the Voronoi diagram does not look like the letter.

The latter property of the inverse typeface is particularly interesting because we can obscure the input text by showing just the Voronoi diagram and not the sites. The result is a tiling generated from text, but which no longer looks like the text. In fact, because we designed the Voronoi typeface as a separate Voronoi diagram for each letter, with only the finite edges forming the letter, we can also obtain obscured tilings from the Voronoi

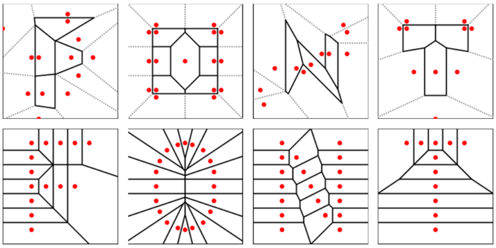

Figure 3: Voronoi typeface (top) and inverse typeface (bottom) for "FONT". Sites are drawn as red circles, and Voronoi edges are drawn as black lines. In the Voronoi typeface, edges extending to infinity are drawn dotted.

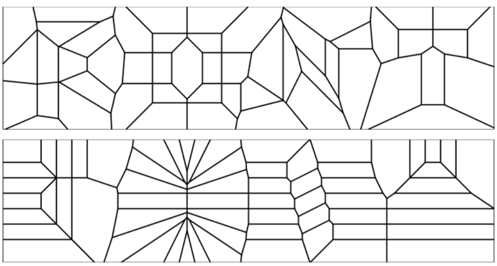

Figure 4: Voronoi tilings produced from Voronoi typeface (top) and inverse typeface (bottom) for "FONT" from Figure 3.

typeface, by combining the sites from all the letters into a single Voronoi diagram.

Figure 4 shows the results of this process applied to Figure 3. The text can be seen in the top tiling upon looking closely, while the text is thoroughly obscured in the bottom tiling.

If we use a regular pattern of letters, we obtain regular tilings. Figure 5 and Figure 6 show two such designs. Alternatively, if we use an irregular pattern of letters, we can make irregular tilings. Figure 7 shows a grid-aligned design, while Figure 8 shows a staggered design.

More complicated text can produce all sorts of irregular tilings. Figure 9 shows self-describing artwork, where the

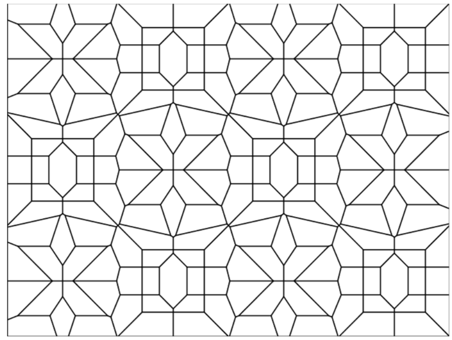

Figure 5: Tiling from an alternating pattern of "X" and "O", using Voronoi typeface.

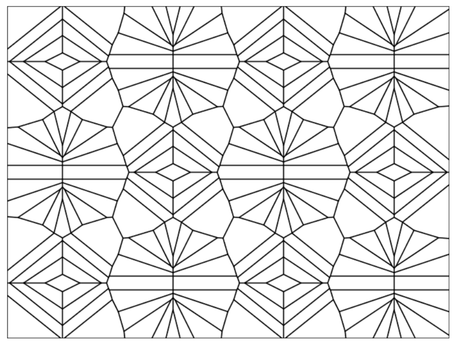

Figure 6: Tiling from an alternating pattern of "X" and "O", using inverse typeface.

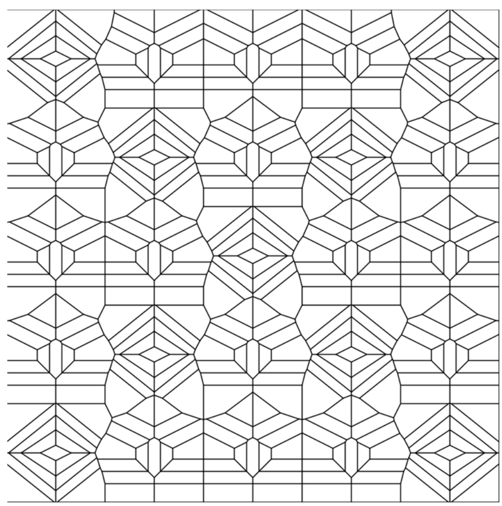

Figure 7: Voronoi tiling produced from inverse typeface for an X pattern of "X"s in a grid of "A"s.

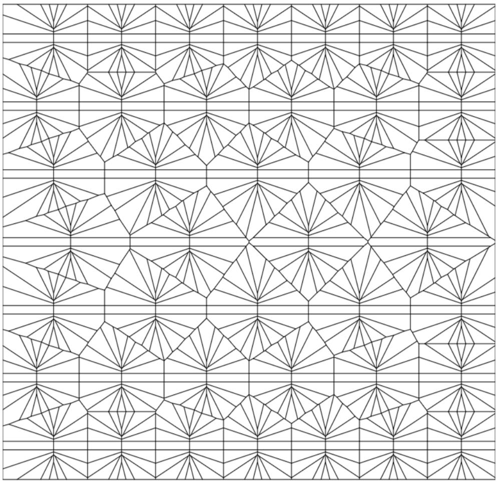

Figure 8: Voronoi tiling produced from inverse typeface from lines of "O"s with varying spacing.

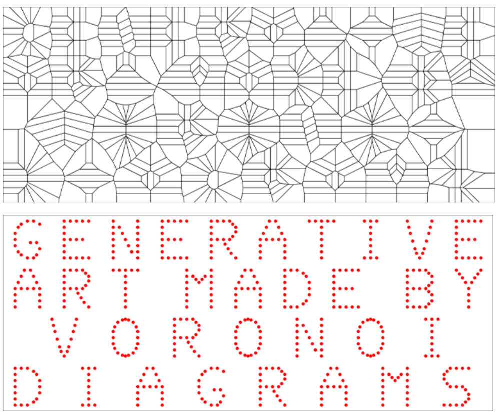

Figure 9: Voronoi tiling produced from inverse typeface (top) and the associated sites (bottom).

underlying text describes the artform itself.

We can also color the tilings. Figure 10 shows a Voronoi typeface tiling for the lyrics of a childhood song, with a random coloring based on the HSL color space and yellow hue ($60°$). To reveal the secret message, we change the hue of certain Voronoi cells to orange ($30°$).

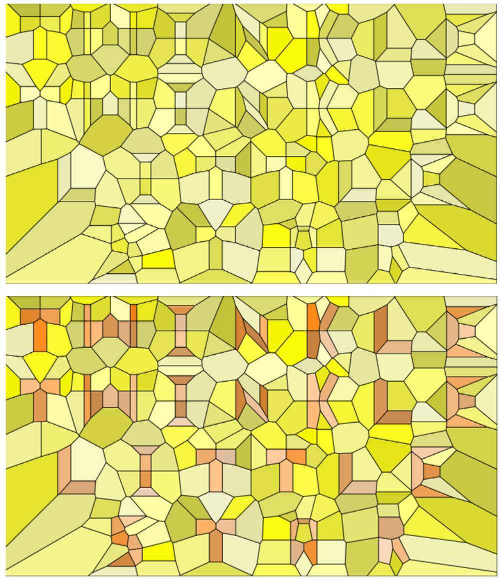

Figure 10: Two colored Voronoi tilings produced from Voronoi typeface for "TWINKLE TWINKLE LITTLE STAR".

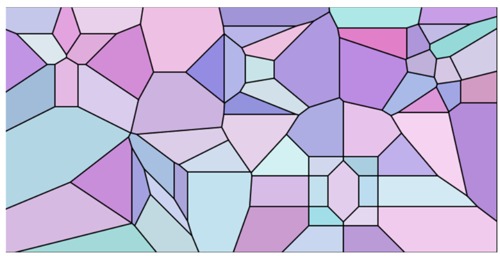

Figure 11: Colored Voronoi tiling produced from Voronoi typeface for "YES / NO".

Figure 11 shows a colored tiling with a visible but obscured two-word message about binary conflict. Figure 12 shows a colored tiling with a completely obscured self-referential message.

3. Maze Folding

Origami design algorithms enable the creation of intricate line drawings, which describe where to fold a piece of paper to achieve a desired form. One such algorithm is for ***maze folding*** [5]: given a

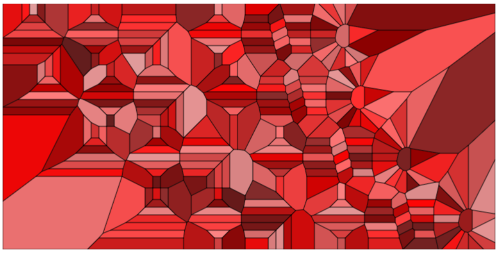

Figure 12: Colored Voronoi tiling produced from inverse typeface for several shifts of "TILING".

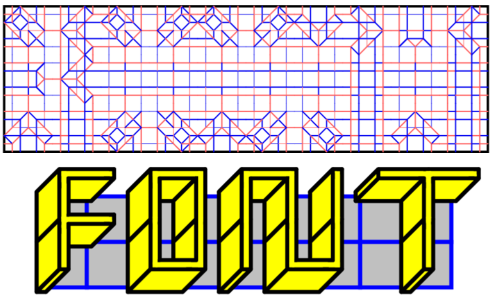

Figure 13: Crease pattern (top) that folds into the 3D shape of "FONT" (bottom), computed by the maze folding algorithm.

"maze" of horizontal and vertical same-height walls arranged in a grid perpendicular to a floor, the algorithm gives an efficient folding of a rectangular piece of paper into that maze. Our origami maze font [6] designs a "maze" that looks like each letter of the alphabet, and these mazes can be combined into a larger maze which then can be plugged into the origami design algorithm.

In its simplest form, we can use this font to design line-art crease patterns that hide a secret message, which could be decoded by folding the piece of paper. Figure 13 shows a simple example. Figure 14 shows a more complex design with self-describing text.

To better illustrate the crease patterns as an artform, Figure 15 shows a design without an image of the corresponding 3D

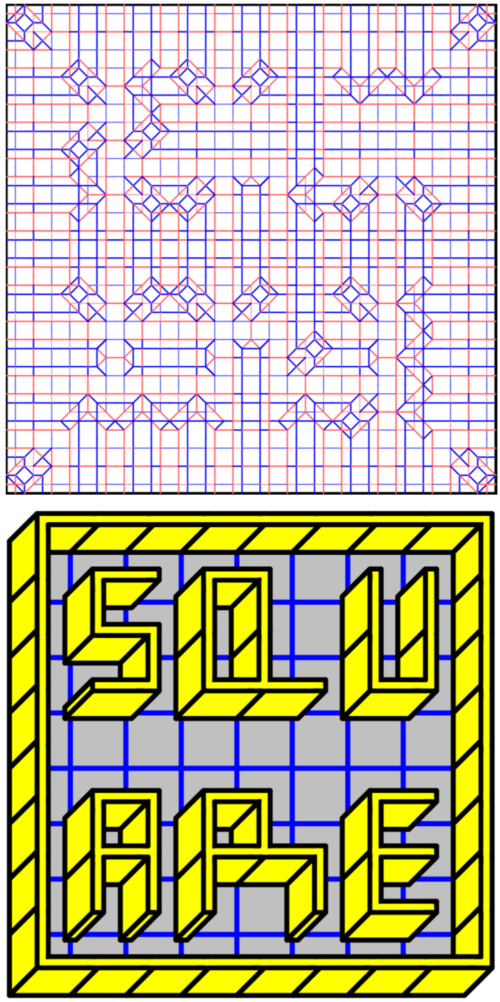

Figure 14: Crease pattern (top) that folds into the 3D shape of a square-framed "SQUARE" (bottom), computed by the maze folding algorithm.

folding. Even by themselves, this form of line art is captivating to study, prompting your mind to figure out what it folds into.

For more sophisticated hidden messages, we can add coloring to the unfolded crease pattern that, when folded, forms an image along with the 3D text. Figure 16 shows our first use of this idea to make text with false shadows, to represent

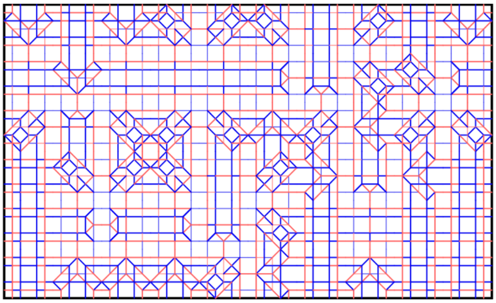

Figure 15: Crease pattern that folds into 3D maze spelling "WORD ART".

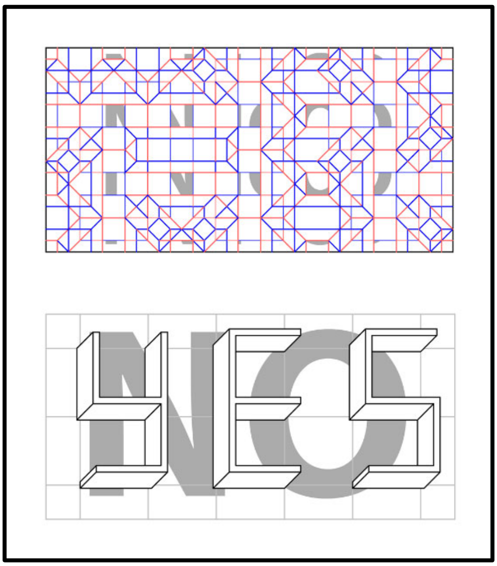

Figure 16: "Yes/No" (2011) by E. Demaine, M. Demaine, and S. Stengle. The shaded crease pattern (top) folds into a 3D "YES" with a false shadow of "NO" (bottom).

binary conflict. See [7] for additional examples. Although we have not yet fully automated this process into software, the same algorithm can be applied manually to any text or maze paired with a background image.

4. One-Fold Silhouettes

In our one-fold silhouette font [8], we designed a way to split letters into pairs of

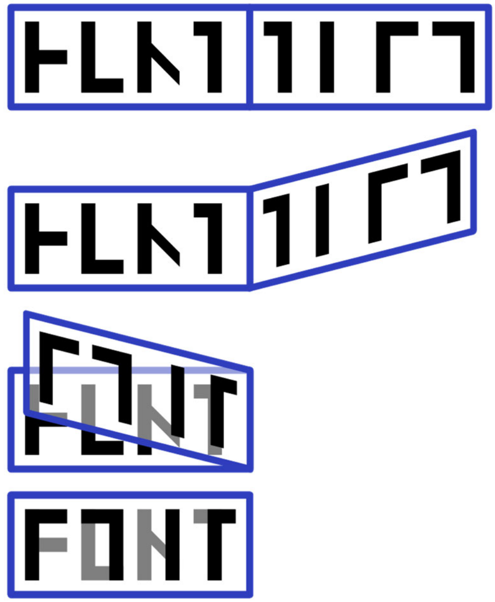

Figure 17: One-fold silhouette font applied to "FONT", ranging from completely unfolded (top) to completely folded (bottom).

shapes that, when overlayed (unioned), form the letters. The individual shapes are designed to not look like letters, or to look like different letters than what they are supposed to be. This gives a minimally obscured font, where two images simply need to be overlayed to reveal the message.

A simple physical mechanism to implement such an overlay is a single fold of translucent material. This mechanism reflects half of the shapes, so in the font we reflect the right half of the shapes so that these inversions cancel out. Figure 17 shows a simple example of the folding process, based on the animation provided by the web app [8].

While we originally envisioned physical objects that allow the viewer to fold and reveal the message, printing the unfolded image as an art print leads to interesting

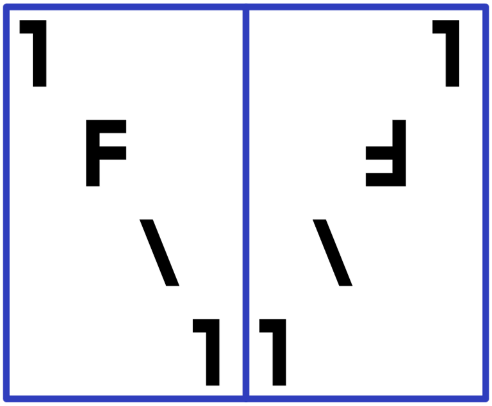

Figure 18: One-fold silhouette font applied to “TEXT” arranged in a diagonal.

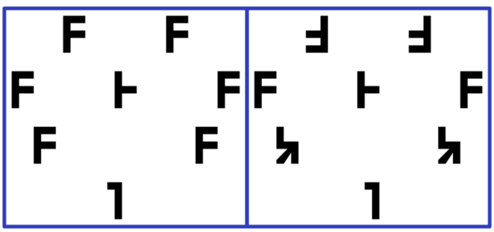

Figure 19: One-fold silhouette font applied to “HEART” arranged in a heart.

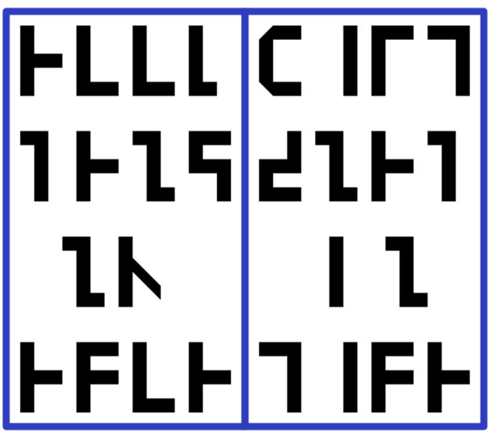

Figure 20: One-fold silhouette font applied to the self-referential text “FOLD THIS IN HALF”.

puzzles for the viewer. Figure 1 above shows one such design, with the solution revealed. Figure 18 shows a more minimal design, with no solution, inviting the viewer to think about what they are seeing. Figure 19 shows a design where the shape of the design gives a hint about the hidden message. Figure 20 shows a self-referential design, where the hidden text describes how to read the hidden text.

5. Glass Cane

Our glass cane font [9] is based on an ancient technique in glassblowing called “caneworking” or “zanfirico”, where rods of colored glass are assembled into a cylinder (or cube) of otherwise clear glass, and then the assembly is pulled and twisted repeatedly. Each rod of glass color thus becomes a helix, and the helices interweave to form complex patterns.

The Virtual Glass software [10] simulates these patterns to enable glassblowers to experiment with different designs before actually making them. For the glass cane font, Virtual Glass enables drawing perfect diagrams of the twisted form of the cane, more geometrically precise than what could be made by a human glassblower.

Figure 2 shows one design with the glass cane font, which includes the readable cross-sections at the top. Figure 21 shows another design with some self-referential text. The twisted cane patterns alone are fun to study, and with effort, the cross-sections can be deciphered from the side view. By omitting this “solution”, we can make prints that require more thought about what message is being shown.

Figure 22 shows a design with an alternative layout for the canes, following a circular spiral pattern.

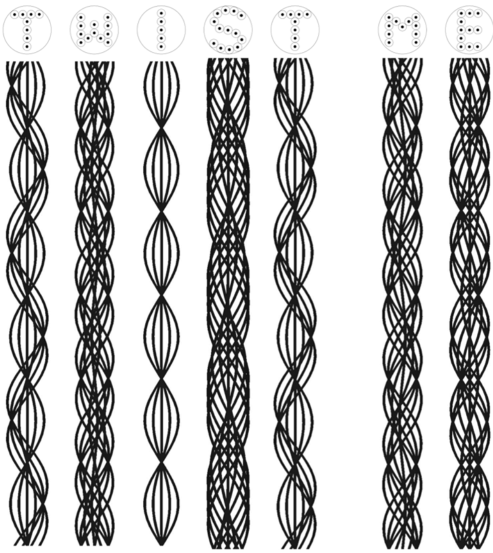

Figure 21: Glass cane font encoding of "TWIST ME", including readable cross-section view (top) and side view (bottom).

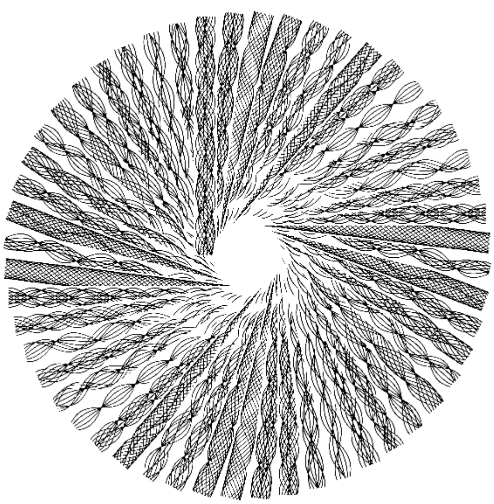

Figure 22: Circular arrangement of canes "A" through "Z", repeated twice, in the glass cane font.

As glassblowers, we have made glass cane from several of the letters in this font. We look forward to incorporating text into our glass pieces by combining multiple letter canes together in a vessel.

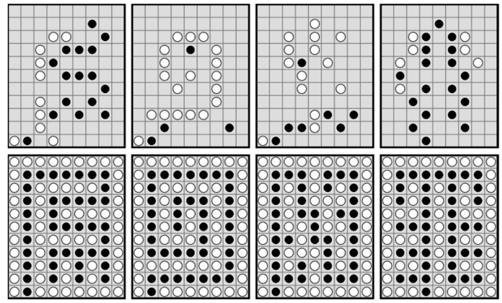

Figure 23: The unsolved (top) and solved (bottom) Yin-Yang puzzles for "FONT".

6. Yin-Yang Puzzles

One of our most recent fonts is based on pencil-and-paper puzzles. Specifically, ***Yin-Yang puzzles*** [11] take place on a square grid of cells, with some cells filled with a black or white circle. To solve the puzzle, the goal is to fill in the remaining cells with black or white circles so that the black circles are connected together, the white circles are connected together (by horizontal and vertical adjacencies), and there are no 2×2 squares of cells that have four circles of the same color.

Our Yin-Yang font [12] is a series of puzzles, one per letter of the alphabet, where the solution's black circles form the letter. Figure 23 shows an example with solution. We designed the solutions by hand to look like letters and satisfy the rules of Yin-Yang. Then we generated corresponding puzzles algorithmically by checking clues in random order for redundancy (according to brute-force search) and thus removability, and then hand-selected the most interesting and challenging puzzle for each letter. See [11] for details.

In many cases, fairly few circles suffice to uniquely determine the solution, and these patterns of circles make nice prints by themselves. The viewer can think through how to solve the puzzle in order

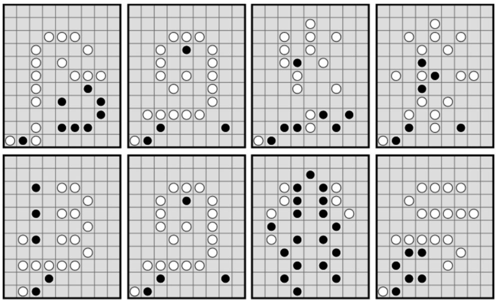

Figure 24: Unsolved Yin-Yang font for the self-referential text "CON-X DOTS".

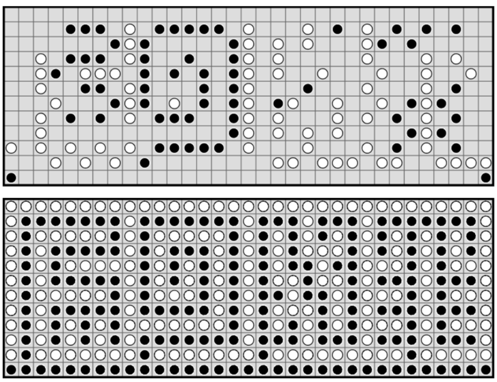

Figure 25: A single Yin-Yang puzzle (top) and its "FONT" solution (bottom) based on Figure 23. For an interactive puzzle, visit https://erikdemaine.org/fonts/yinyang/font.html

to reveal the text.

Figure 24 shows an example with self-referential text, where the solved text roughly explains what its own solution does: connects dots (where "connects" is abbreviated to "CON-X").

With more computational effort, we can construct larger Yin-Yang puzzles that contain entire words or phrases. The solutions can be obtained by carefully joining together individual letters from the font to still satisfy the constraints of Yin-Yang, while constructing minimal puzzles with the desired unique solutions takes longer (about a minute).

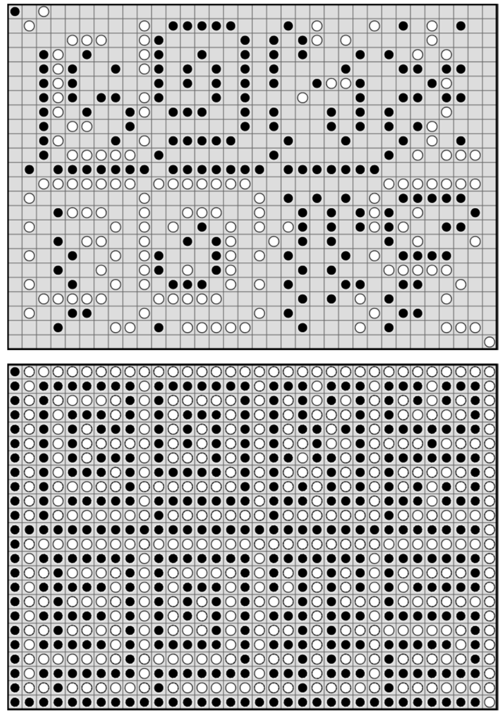

Figure 26: A single Yin-Yang puzzle (top) and its "CON-X DOTS" solution (bottom) based on Figure 24. For an puzzle version, visit https://erikdemaine.org/fonts/yinyang/conx_dots.html

Figure 25 and Figure 26 show single Yin-Yang puzzles and their solutions based on Figure 23 and Figure 24, respectively. Interestingly, these larger puzzles seem to need more clues to force unique solutions, resulting in medium-density designs. If desired, we could add extra dots to the puzzle to attain a desired higher density and readability.

7. Conclusion

We have shown several generative art designs that can be produced with algorithmically generated artistic fonts. We believe that this work only scratches the surface of what is possible within this genre. The reader is encouraged to try their hand at their own creations, either with our font software [3] or their own creations.

References

[1] B. T. Franklin, "Lost Rituals: Generating Text Using Behavioral Data Objects," in *23rd Generative Art Conference*, 2020.

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. https://paperswithcode.com/paper/language-models-are-unsupervised-multitask

[3] E. Demaine and M. Demaine, "Mathematical and Puzzle Fonts/Typefaces," 2003-2021. https://erikdemaine.org/fonts/

[4] E. Demaine and M. Demaine, "Voronoi Font," 2017. https://erikdemaine.org/fonts/voronoi/

[5] E. D. Demaine, M. L. Demaine and J. Ku, "Folding Any Orthogonal Maze," in *Origami^5: Proceedings of the 5th International Conference on Origami in Science, Mathematics and Education*, A K Peters, 2010, pp. 449-454.

[6] E. Demaine, M. Demaine and J. Ku, "Origami Maze Font," 2010. https://erikdemaine.org/fonts/maze/

[7] E. D. Demaine and M. L. Demaine, "Adventures in Maze Folding Art," *Journal of Information Processing,* vol. 28, pp. 745-749, 2020.

[8] E. Demaine and M. Demaine, "One-Fold Silhouette Font," 2018. https://erikdemaine.org/fonts/silhouette/

[9] E. Demaine and M. Demaine, "Glass Cane Font," 2014. https://erikdemaine.org/fonts/cane/

[10] "VirtualGlass". http://virtualglass.org/

[11] E. D. Demaine, J. Lynch, M. Rudoy and Y. Uno, "Yin-Yang Puzzles are NP-complete," in *Proceedings of the 33rd Canadian Conference in Computational Geometry*, 2021.

[12] E. Demaine and M. Demaine, "Yin-Yang Font," 2021. https://erikdemaine.org/fonts/yinyang/