Working w

parameter

creating w

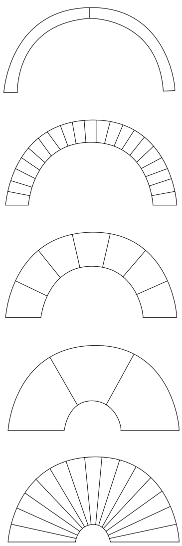
thinking ab

visual for

In this capacity, the designer is no longer making choices about a single, final object but creating a matrix encompassing an entire population of possible designs. This is a move from thinking about an object to thinking about a field of infinite options. This involves searching for and exploring a population of designs that meet certain requirements, behave in particular ways, or fit the desires of the designer.

These letters use parameters to determine the size of each circle. Working point by point, the program draws one circle at a time, with the radius cycling between small and large. Because the frequency of oscillation and circle size are both parameters, this permutation is only one of an infinite number of possible results.

Geno Pheno Sculpture "Fractal Dice No. 1", by Keith Tyson, 2005 A statement released for the exhibition of this sculpture declared, "An algorithm written by the artist determines what will be on view in this new exhibition." The subsequent rolls of a die determined the parameters of the sculpture, including the color, depth, and position of each element.



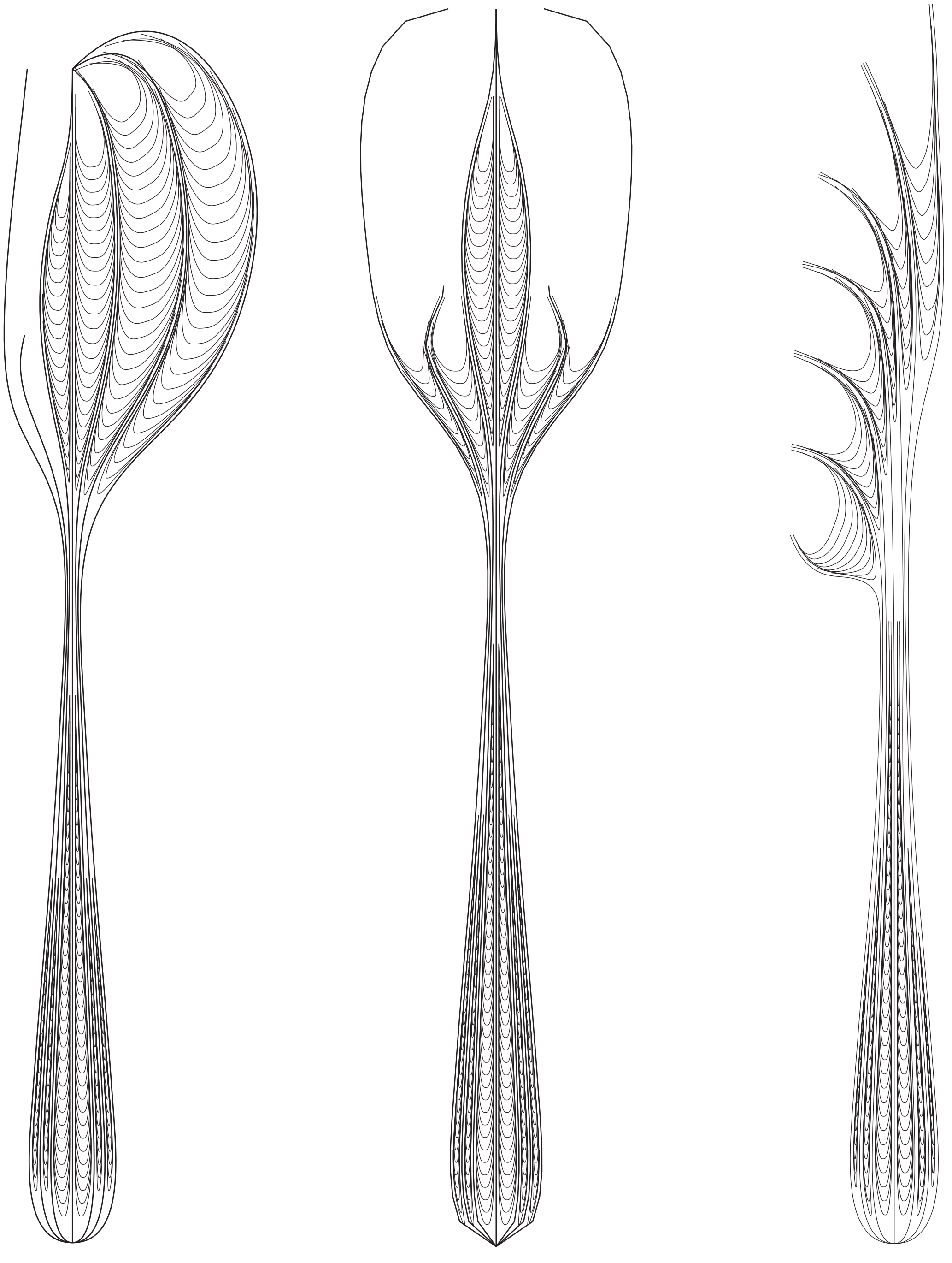
Defined broadly, a parameter is a value that has an effect on the output of a process. This could be something as straightforward as the amount of sugar in a recipe, or as complex as the activation threshold of a neuron in the brain. In the context of architecture and design, parameters describe, encode, and quantify the options and constraints at play in a system. A common constraint might be the budget available for a project, while a configuration option might control color, size, density, or material.

Identifying and describing the variable elements in a process—be it a section of code or the rules of a Dadaist poem—is called parameterization. This multistep process requires that the designer decide both what can change and the range of possible values for each parameter. For example, a designer can explore the effects of different color palettes on a logo design. In this case, the colors of the elements within the logo are the parameters, and the list of possible palettes defines the value range. Parameterization creates connections between the intention of the designer and the system he or she is describing.

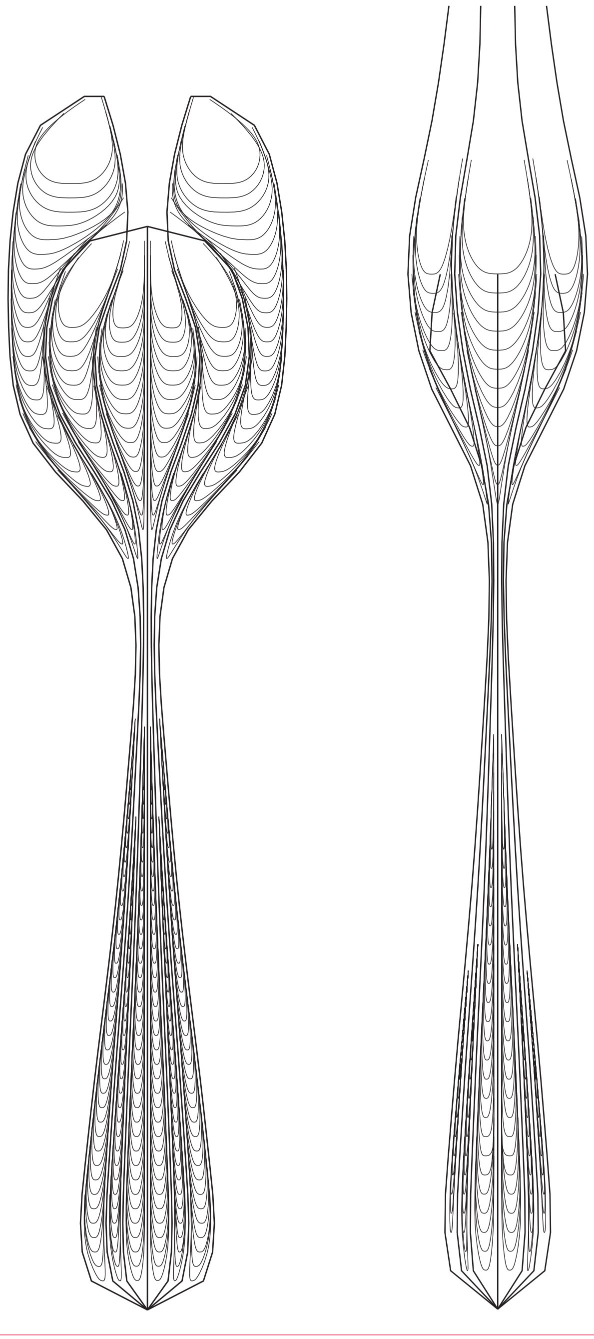
As a greater number of parameters are identified and incorporated into a process, the number of possible outcomes also increases. Imagine each parameter as defining an axis on a graph, and a parameterized system as defining a space populated by potential design states (resulting from a combination of specific values being assigned to each parameter). As a simple example, consider a rack of T-shirts. Each shirt on the rack is a different size and color, and we can say that the "large, green shirt" is the design state when the size parameter is large, and the color parameter is green. But it is just as easy to imagine a "large, red shirt," as being exactly the same as a "large, green shirt," just in a different color.

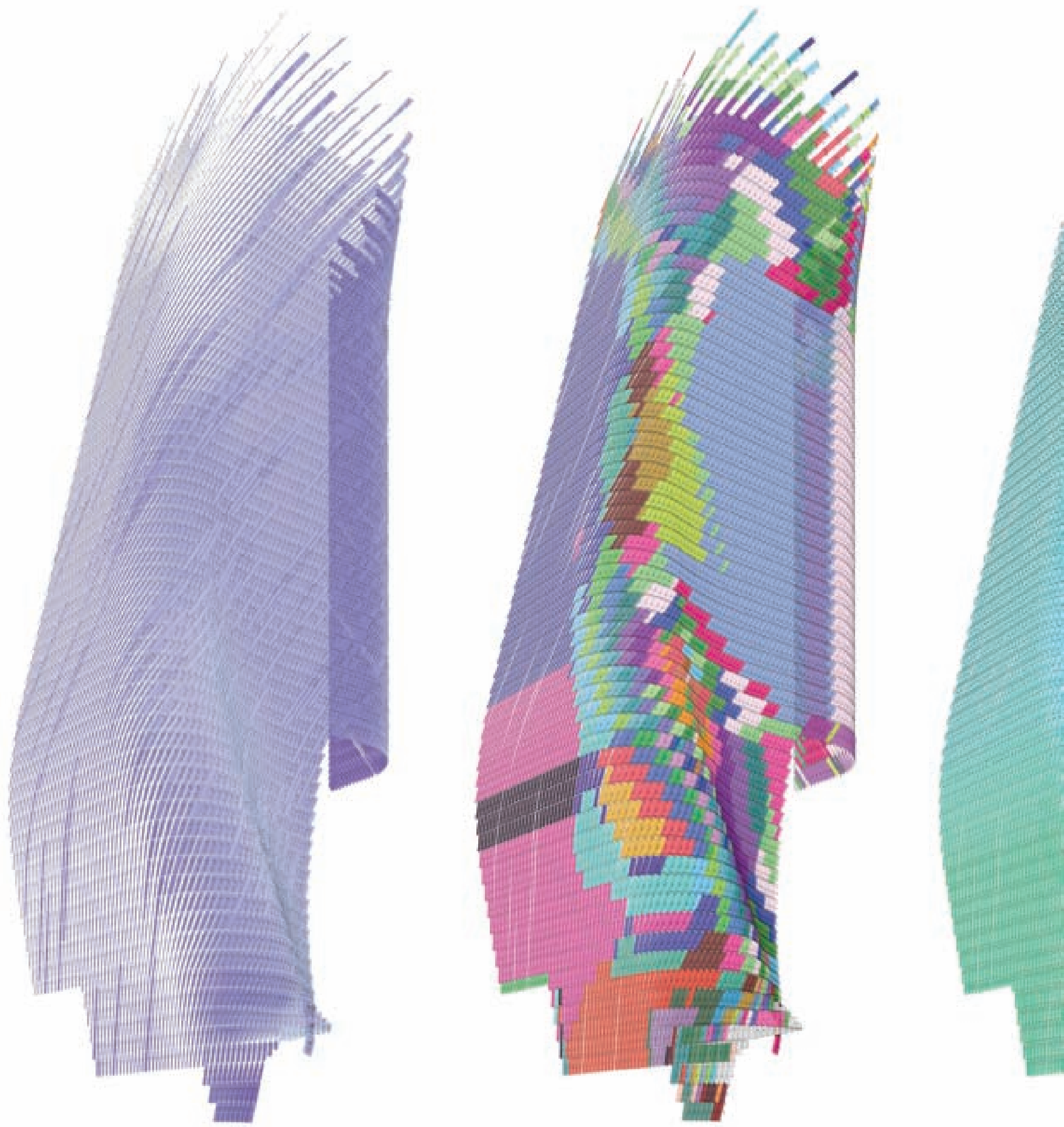
Thinking about parameters provides a bridge between repetition and transformation, as well as visualization and simulation. While transformation describes a parameter's effect on form, repetition offers a way to explore a field of possible designs for favorable variations. Both visualization and simulation require the use of parameters to define the system, and they describe how data or other inputs will influence the behaviors of that system.

Parameters
Two parameters—radius and number of segments—are varied to create a wide range of different forms. This is a minimal system, but it shows the power of parameters for exploring form.

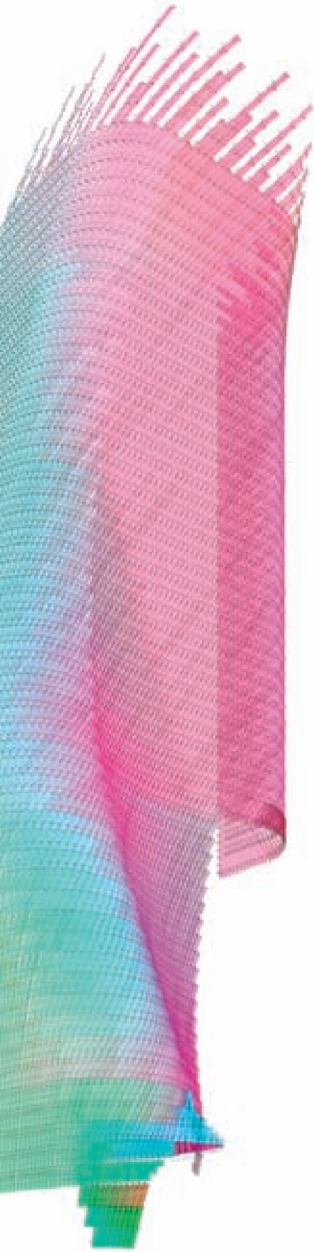


Flatware,
by Greg Lynn,
2005–present
This fifty-two-piece flatware set was designed by mutating, blending, and evolving a base form: a bundle of tines and a webbed handle. These specialized utensils, including obscure pieces such as a mustard spoon and bon bon server, belong to a larger family of forms, with each being a variation of the others.





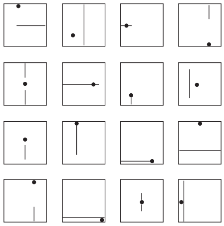
Phare Tower, by Morphosis, 2008
The Phare Tower is a design for a sixty-eight-story skyscraper in Paris. Software developed by Satoru Sugihara was used to iteratively develop, test, and refine the structure to address multiple parameters essential to the design. These images show how the software was used to design the following systems; shown from left: optimization for solar performance, panel-dimension optimization, and panel-angle analysis.

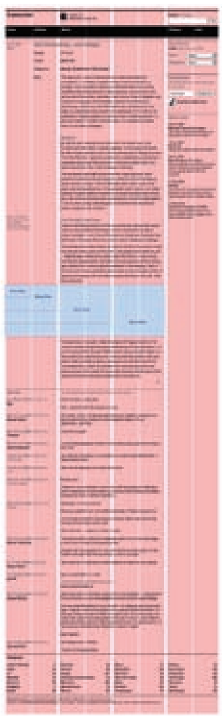


Painted and unpainted sheet aluminum, iron wire and copper rivets. 84.5 inches high (214.6 cm) Soloman R. Guggenheim Foundation, Peggy Guggenheim Collection, Venice, 1976. 76.2553.137. Photograph by David Heald © The Solomon R. Guggenheim Foundation, New York. Art © 2009 Calder Foundation, New York / Artists Rights Society (ARS), New York.

Arc of Petals,
by Alexander Calder,
1941
This mobile is composed of a well-balanced set of relationships between shapes. Some of these relationships are constant; others change in response to air currents.

COMPOSITION SYSTEMS





The desire to construct a system for composing images, rather than making a single image, has a long history in modern art. Marcel Duchamp's 3 Standard Stoppages from 1913–14 is an early and fascinating example. To create this series of objects, he dropped a string, measuring 1 meter, from a height of 1 meter to define a curve. Defined by gravity, this ephemeral curve and the twisting of the string as it fell was then cut out of wood and used as a template for other images. For example, the curves were used within his Large Glass to define the shapes of the bachelor figures. A contemporary of Duchamp, Jean Arp, produced collage works, such as Untitled (Collage with Squares Arranged According to the Laws of Chance), by scattering the elements onto a page. Perhaps the clearest, most iconic presoftware examples are the mobiles of Alexander Calder. In these sculptures, shapes relate to one another through fixed connections, under the weight of gravity, but they are so well balanced that the wind can move the elements to shift their positions. Umberto Eco wrote of these objects:

> Each of his works is a 'work in movement' whose movement combines with that of the viewer. Theoretically, work and viewer should never be able to confront each other twice in precisely the same way. Here there is no suggestion of movement: the movement is real, and the work of art is a field of open possibilities.[1]

Experimental writers Tristan Tzara and William S. Burroughs introduced innovative, unpredictable operations as methods of writing, and John Cage used randomness as a fundamental technique for musical compositions.

While it's clear that these early compositional systems relied heavily on chance, these artworks are important within the context of parameters in that each of their creators defined a set rules where some elements were selected by themselves and others resulted from events outside of their control. They invented systems from which an infinite number of unique works could (and did) emerge. This way of working is summarized well by Sol LeWitt's statement: "The idea becomes a machine that makes the art."[2]

More carefully determined systems include grids for composing pages in books, magazines, websites, and for posters. They allow each page to be unique, while still relating to every other page. For example, the Unigrid System, designed for the U.S. National Parks Service (NPS) in 1968 allows each park (Yellowstone, Yosemite, etc.) to have a unique brochure suited to its needs while also allowing the NPS to maintain a strong organizational identity. The Unigrid System is a flexible, open framework that allows individual designers to make their own decisions about layout while working within a larger system. Subtraction.com, which is the website and blog of Khoi Vinh (the art director of NYTimes.com) is a more contemporary example. A grid system applied to a website allows for hundreds, even thousands of pages to be generated based on a single structure. Vinh cites Josef Müller-Brockmann and Massimo Vignelli as influences, and he makes direct links between the history of the grid within print design and how it transfers to the web. Speaking about grids in an interview, he said:

> A grid system is not just a set of rules to follow…but it's also a set of rules to play off of—to break, even. Given the right grid—the right system of constraints—very good designers can create solutions that are both orderly and unexpected.[3]

[1] Umberto Eco, The Open Work (Cambridge, MA: Harvard University Press, 1989), 86.

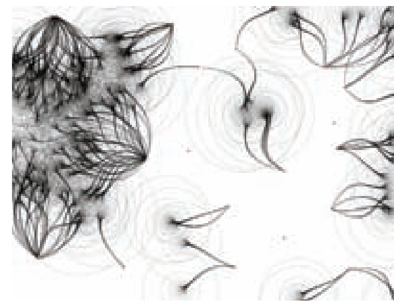
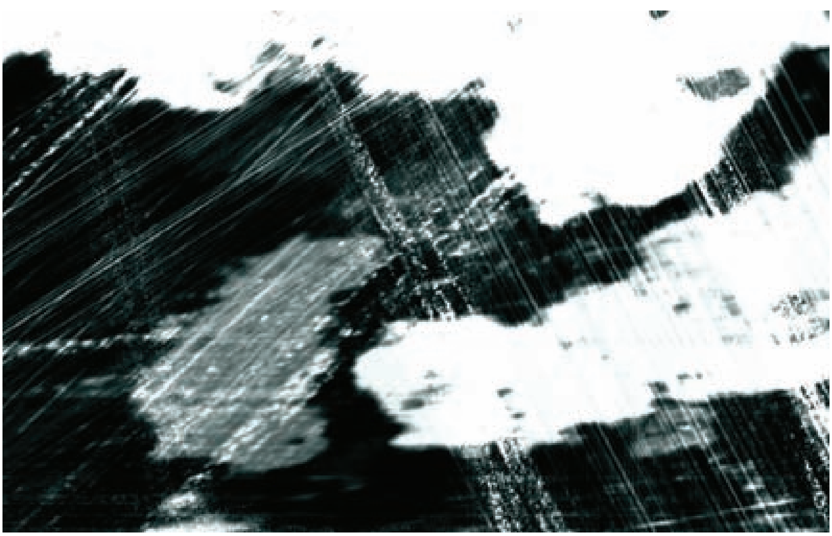
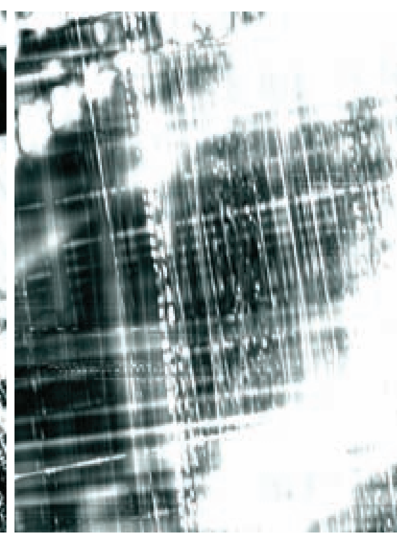
[2] Alexander Alberro and Blake Stimson, eds., Conceptual Art: A Critical Anthology (Cambridge, MA: MIT Press, 2000), 14.

[3] Interview with Khoi Vinh: http://www.thegridsystem.org/2009/articles/interview-with-khoi-vinh/.

Subtraction.com, by Khoi Vinh, 2000–09
There are many different layouts on this website, but Vinh uses an eight-column grid to give the same structure to each variation.

Variation Example, by Emil Ruder, 1967
Emil Ruder's book Typographie shows how a highly constrained set of elements can be used to generate a wide range of compositions. Using only one line and one dot, he diagrams thirty-six unique compositions. He makes a point to write that his variations are "only a fraction of the almost unlimited possibilities."



proximityOfNeeds,
by Lia, 2008
Lia provides controls at the bottom of the screen, so viewers can change the parameters of her software and therefore change how the forms grow. She controls these parameters live in her audiovisual performances.

Mortals Electric,
by Telcosystems, 2008
This sophisticated, live audiovisual performance has more parameters than can be manually controlled. It balances direct control with an automated context-aware system to produce a stream of sonic and visual events that move beyond the expectations of its creators.

VARIABLES

In any system or set of rules, there exists the potential for variation. Though the primary variation of form present in a Calder mobile comes from the unpredictable interaction of natural forces, it is still possible to get an even wider field of possibilities by changing other parameters in the system. These include the lengths of the rods, weights of the objects, and positions of the connections. A compositional system built out of 1-foot (0.3-meter) rods will look and behave very differently than a system made of 1-meter (3.3-feet) rods.

When the value of a parameter can change, we call this a variable. Variables can be distinguished from constants, whose values cannot change, such as the force of gravity; or constraints, which are fixed in response to the requirements of the project, such as cost or available materials, and provide boundaries that define the edges of a design space. The creation of a mobile, for example, may be constrained by the size of the room it will hang in and the need for it to be light and strong enough to hang from the ceiling. Though all three of these parameter types will effect the range of possible forms, the variables can be considered as the primary axes of variation. The artist changes the variables, either by hand or with code, in search of interesting outcomes.

Sometimes the variable's value will only makes sense within a certain range. Consider the tuning knob on a radio; only frequencies within the range shown on the dial are valid. It is conceivable that a radio could use values outside of this range, but the results will be unexpected, and certainly won't sound like radio. Defining the range of values is one way that designers can assert their aesthetic sensibilities in a parameterized system. Perhaps not all values will look good or create interesting results. Much like the dial on the radio, the range can be refined to produce a narrower, but more pleasing field of variations.

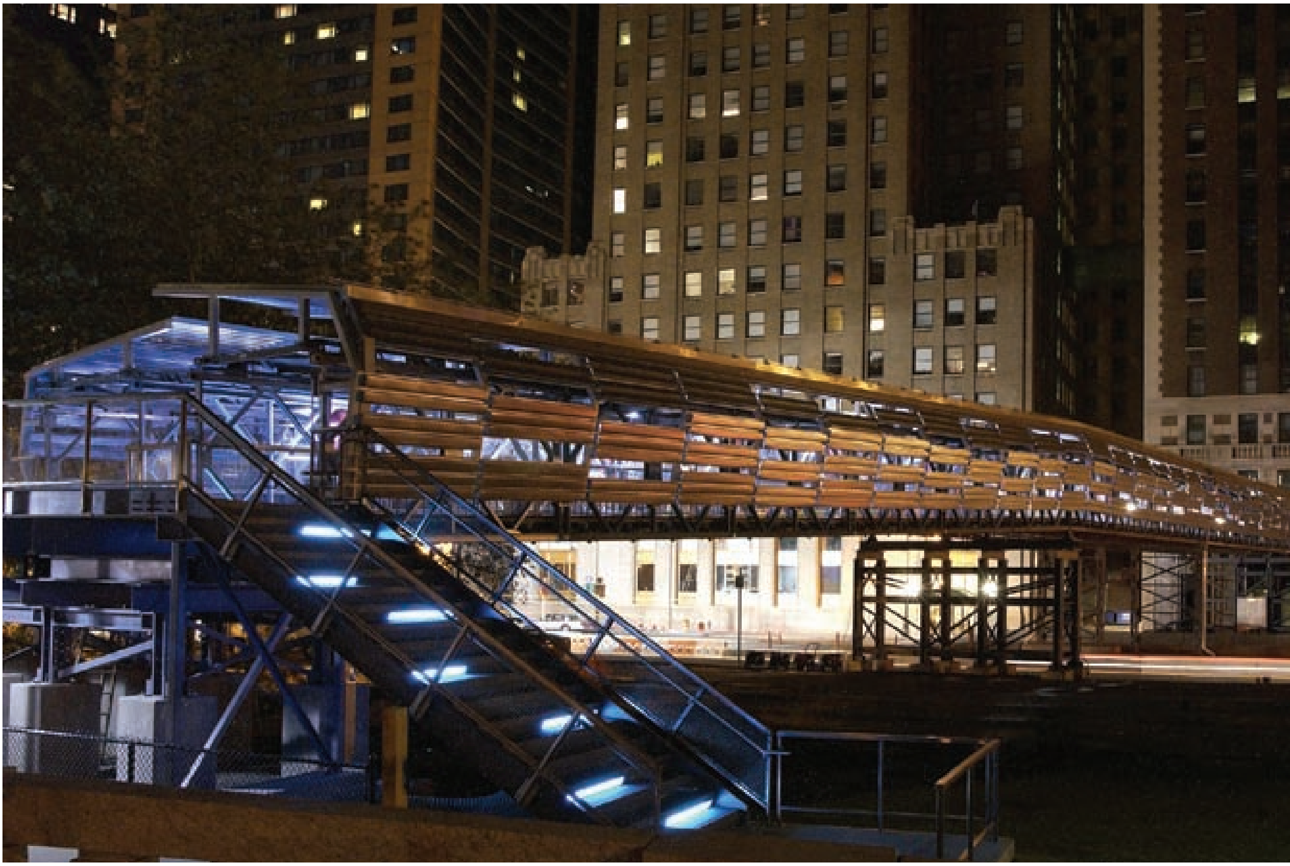
As with the compositional systems in use prior to the invention of the personal computer, randomness is a useful tool for finding interesting variations in a parameterized system. Random values can be used to emulate unpredictable qualities of our physical reality and to generate unexpected compositions. Though not as random as a toss of the dice, code provides a more flexible way to create random values. Sequences of random numbers can be generated in such a way that each number in the sequence differs only slightly from the last; this technique aids in the simulation of natural effects like wind, waves, and rock formations.[4] Although using random numbers to find interesting variations is not an efficient way to discover every possible form, it does provide a way to explore an exhaustively large parameter space in order to get an idea of possible outcomes. Even in a small system that uses only three parameters, each of which has a value from zero to 100, there are a million possibilities—far more than one can explore methodically.

[4] Ken Perlin greatly impacted the computer graphics world with his invention of Perlin Noise in 1985. This method for generating textures is widely used in computer graphics to create visual effects like smoke, fire, clouds, and organic motion.

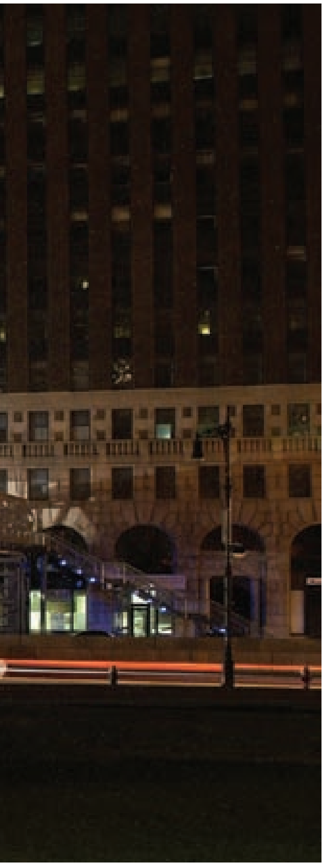
Two Space, by Larry Cuba, 1979
Cuba combined a set of nine tiles with twelve symmetrical pattern arrangements to create a mesmerizing animation. The white-on-black scheme creates optical illusions, figure-ground reversals, and after-image effects. The parametric organization of the software allows for any of the tiles to be combined with any of the patterning schemes.

Sweep,
by Erik Natzke, 2008
Natzke creates his images with custom drawing software that allows him to turn parameters on and off while he draws on-screen.

fractured landscape, by Jean-Pierre Hébert, 2004
The shape of the lines in this image were generated entirely from code, and drawn onto paper by a mechanical plotter. The physical quality of this dense drawing is impossible to reproduce in a book. The rich lines of the pen's marks and the high-quality paper have an essence that cannot be duplicated using modern printing processes.



Rector Street Bridge #1, by SHoP Architects, 2002
SHoP Architects was asked to design and construct a bridge in the shortest possible time. They used parametric techniques to meet these requirements and to create an interesting system for augmenting a prefabricated structure.



CONTROL

Parameters are often used to create a system that generates optimal variation within a given set of constraints. These constraints can be semifixed, meaning that they provide a boundary for the field of variations, but the constraints themselves can be changed when necessary. The most common example of this type of constraint is the cost of producing a variation. For example, a CNC-milling machine is a cost-effective digital fabrication technique, but most machines are not capable of producing shapes with undercut areas. The CNC bit sits on a computer-controlled arm that descends onto the material, so it can only remove the material on top. A parametric design system might take this into consideration by eliminating any design model that requires this type of cut. The constraint is semifixed, because it is always possible to use a different, more costly fabrication technology if necessary, or if the budget allows.

When the architecture firm SHoP Architects was commissioned by the Federal Emergency Management Agency (FEMA) to design a bridge in Lower Manhattan to restore a vital pedestrian connection that was lost after September 11, they used a combination of readymade structures and parametric design techniques to quickly create an interesting, cost-effective form. The architects wanted the exterior cladding of the bridge to let light in during the day and allow it to shine through at night, but also to discourage sightseers from stopping on the walkway. These programmatic constraints were combined with time and budgetary concerns, as well as the fixed geometry of the readymade bridge structure. Taking all of these elements into consideration, they were able to create a pattern using only a small number of unique pieces that gave the feel of a much more complex system.

In contrast to using random numbers to explore a field of possible designs, parametric systems can be controlled to determine the final form and to meet specific needs. In something akin to turning the knobs of an old television set to tune the picture, in this model, the designer first creates the system and defines the parameters, then inputs and adjusts specific settings to control and optimize the final output. Taking this one step further, it is possible to quantify and encode the designer's preferences or certain outcomes into the system. Less strict than a constraint, this allows flexibility in the range of possible variations, while encouraging the system to generate solutions that the designer likes.

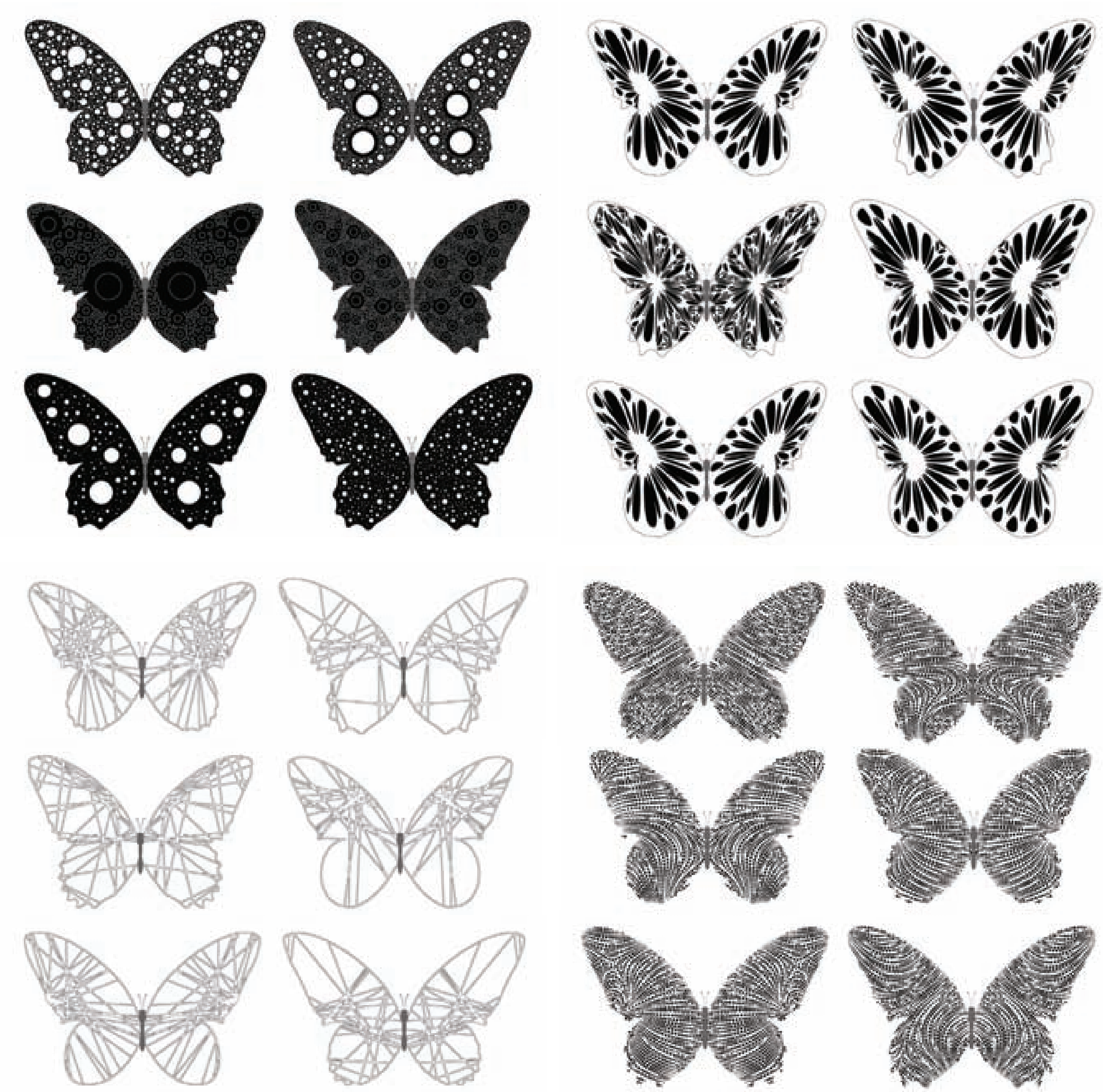
Though parametric control is often understood in terms of numbers and variables, complex and unpredictable forms can be achieved by linking the parameters of multiple elements together. For example, an architect designing a staircase might want to connect the height of the staircase to the height of the ceiling, so that if the ceiling height changes, the staircase will adapt to the new values. The space of possible staircases is coupled to the properties of a single variation of the ceiling. This type of controlled coupling allows the designer to experiment and explore a field of possibilities, while simultaneously controlling related forms. In this way, parameters can be used to control things like proportion and scale without giving up the freedom to try different permutations.

SCUMAK no. 2,
by Roxy Paine, 2001
Paine's machine resembles industrial production equipment, but it produces idiosyncratic blobs of polyethylene, rather than mass-market consumables. A software program controls the process: folding layers of material to subvert the idea of mass production and produce a series of unique, appealing objects.

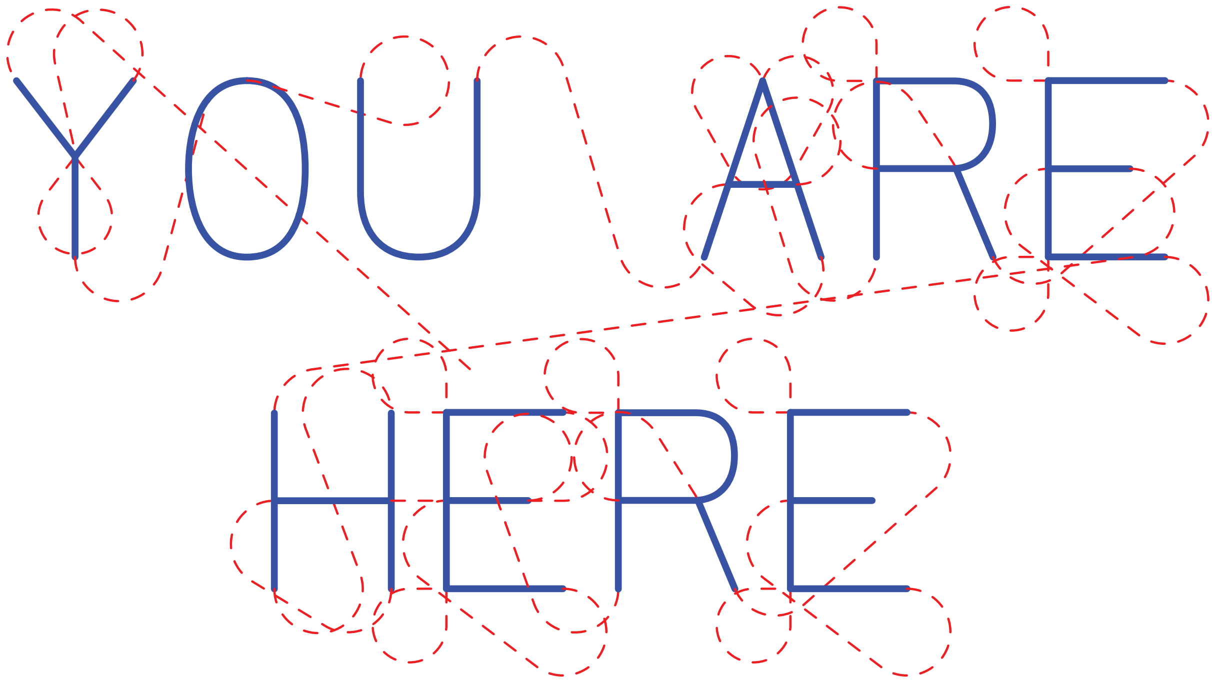
PARAMETER TECHNIQUE
ONE-OF-ONE

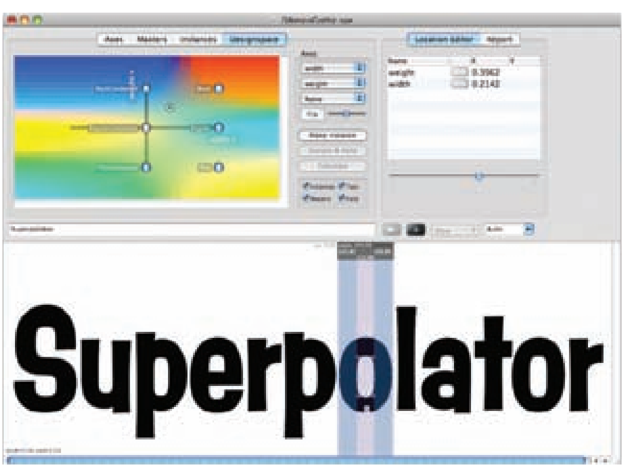
Using parameterized algorithms to generate form involves an exploratory process of searching through a field of designs to find interesting variations. As a result of this process, each variation is likely to appear related to but different from other possible versions. Depending on the number of parameters and the range of possible values, these resemblances can be subtle or extreme. When used in conjunction with repetition, these techniques allow artists and designers to straddle the boundary between creating serial editions and one-of-a-kind pieces. By presenting multiple versions, one-of-one works operate as singular pieces, but at the same time provide a window into the complexity of the system used in their creation.

P1011-F1, P1011-H1, P1011-I1, by Manfred Mohr, 2004 Mohr explains that his software "selects a subset of cubes from a repertoire of 42,240 cubes inherent to the 11-d hyper-cube." Each image offers a view of the geometry through a window, as the structure rotates in eleven dimensions.



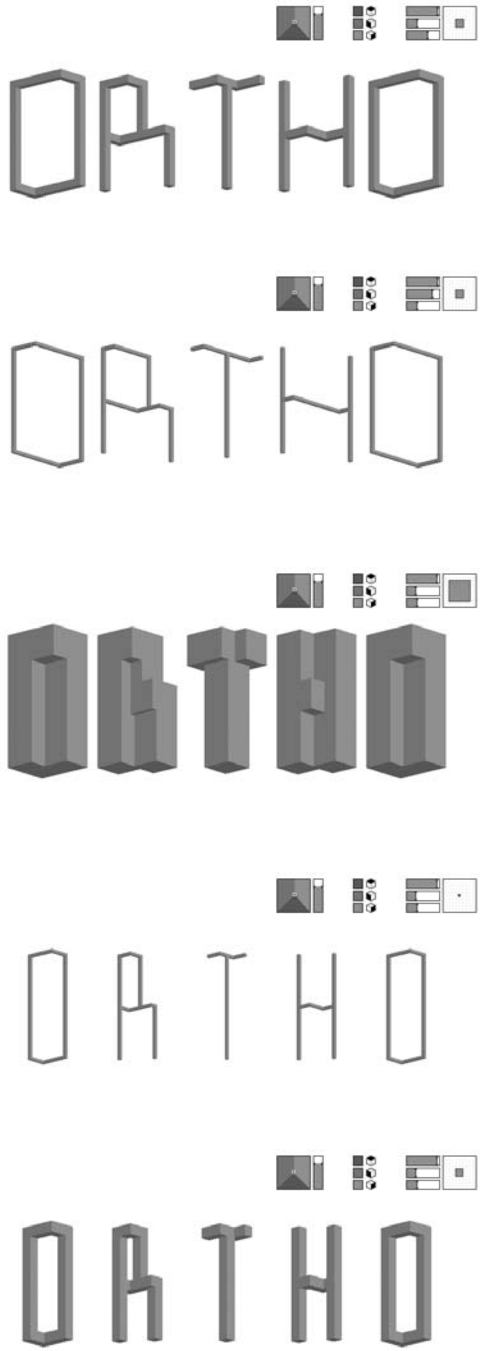
Biomimetic Butterflies, by The Barbarian Group, 2007
These wing patterns were created using custom algorithms, each parameterized to generate variety while preserving family resemblances.





Scriptographer, by Jürg Lehni, 2001–present
This plug-in is used to write scripts for Adobe Illustrator. In this example, the Scriptographer program was used to calculate the motion paths (shown in red) for Hektor, a computer-controlled spray-paint device.

Superpolator, by LettError, 2007
This software interpolates typographic glyphs between multiple axes. For example, a simple font could be generated using weight and width axes. A more intricate setup could have an axis for x-height, ascenders, descenders, weight, and contrast. Superpolator provides an interface for easily navigating these axes and for viewing the changes through animation.

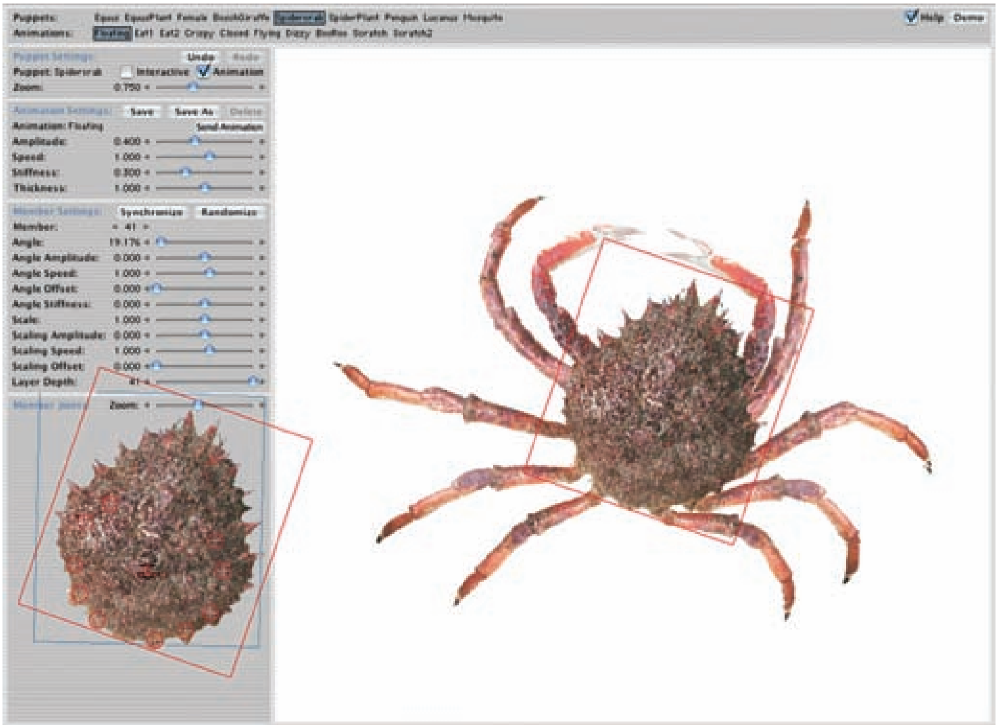


PARAMETER TECHNIQUE
VARIABLE FONTS

Since their invention, systems of writing have always been implicitly parameterized; the introduction of software has increased these possibilities. In ancient cuneiform writing, each symbol was made from a pattern of wedges impressed in clay. Changing the parameters, such as the quantity or size of the spaces between wedges, defined which character was written. This parameterization persisted to some degree in contemporary typefaces, but the Univers type family designed by Adrian Frutiger in 1954 was a landmark in this respect. Univers is a system of twenty-one related fonts, designed around the parameters of width, weight, and slant. The fonts range from Univers 39 Thin Ultra Condensed to Univers 83 Heavy Extended, with Univers 55 Roman as the base font.

Programming legend Donald E. Knuth's Metafont language from 1979 is a logical extension of Frutiger's plan. As the first fully parameterized software typeface, Metafont was capable of generating letters of every width and weight, because each was defined by a geometric equation. This idea was further commercialized by Adobe, with their Multiple Master (MM) technology. Like Metafont, an MM font such as Myriad or Minion could generate any width or weight according to the parameters set up by their designers. It remains an open question as to whether this kind of typographic flexibility is needed or even feasible, given that Adobe stopped producing MM faces in favor of the OpenType format. Nevertheless, this typographic quest was continued by LettError's Superpolator.

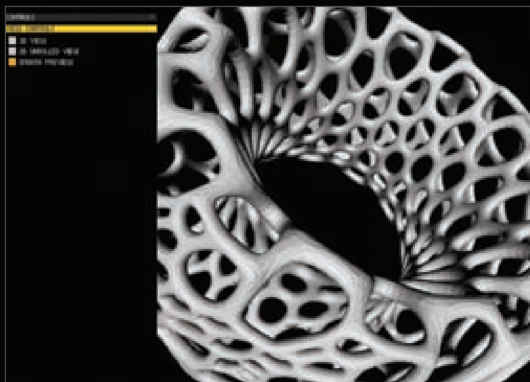
Ortho-Type,
by Enrico Bravi, Mikkel Crone Koser, and Paolo Palma, 2004
This orthographic-projection typeface system provides control over the 3-D view, height, width, depth, thickness, and color.



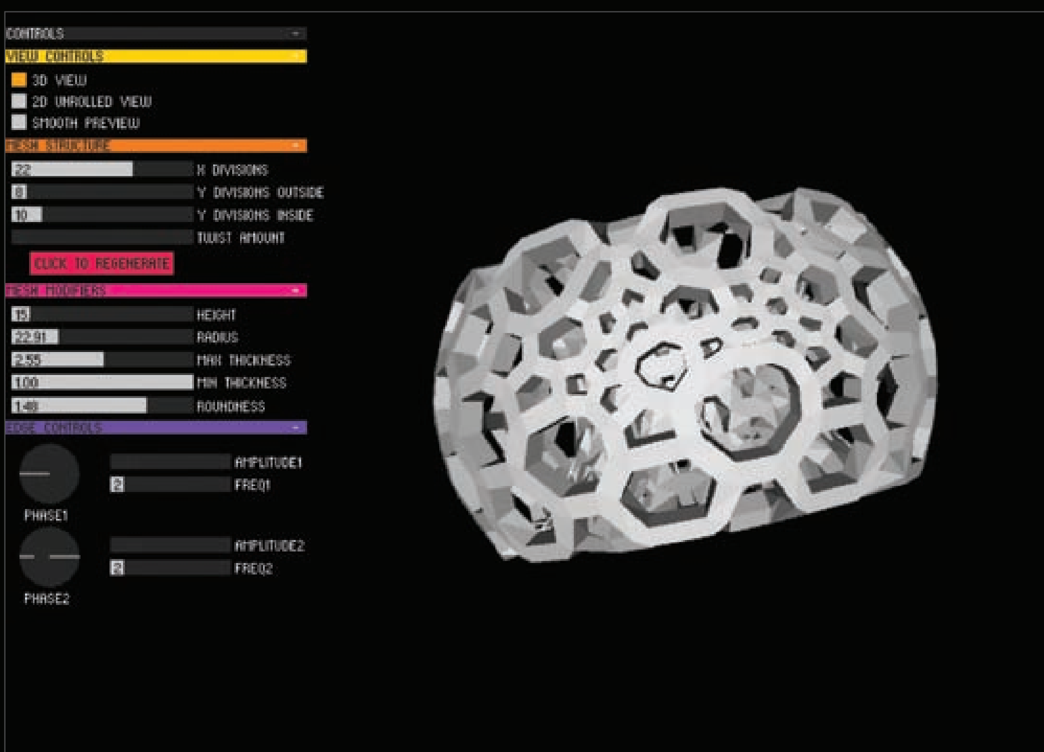
PuppetTool,
by LeCielEstBleu, 2002
The console on the left of this software tool controls a menagerie of screen-based puppets. It offers the abiity to reconfigure and distort the motion and connections between the limbs.

Oasis,
by Yunsil Heo and Hyunwoo Bang, 2008
The behaviors of the aquatic creatures in Oasis have over thirty parameters that can be tuned to change their behaviors. As part of the design process, this console allows users to modify the procedural animation while it runs.

PARAMETER TECHNIQUE
CONSOLE



A console is a set of controls for electronic or mechanical equipment, such as cars, airplanes, radios, and milling machines. The structure of the console has migrated to software interfaces. For example, many computer games have a dense set of controls at the bottom of the screen, allowing players to monitor and control the status of the game. The values controlled through a software console are linked to variables within the program; they allow the state of the program to change without rewriting the code. Users can manipulate the program, even if they don't know how to program; and programmers can view changes immediately without stopping the program, changing variables in the code, and restarting. Consoles are used as interfaces for a wide range of tasks, including for designing buildings, flying a simulated airplane, or building avatars for online communities.

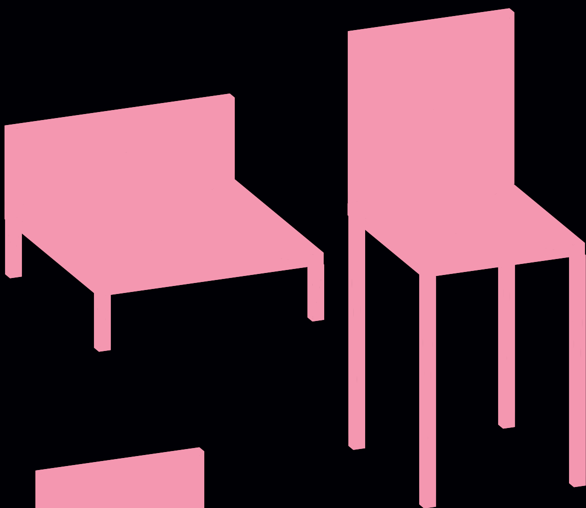


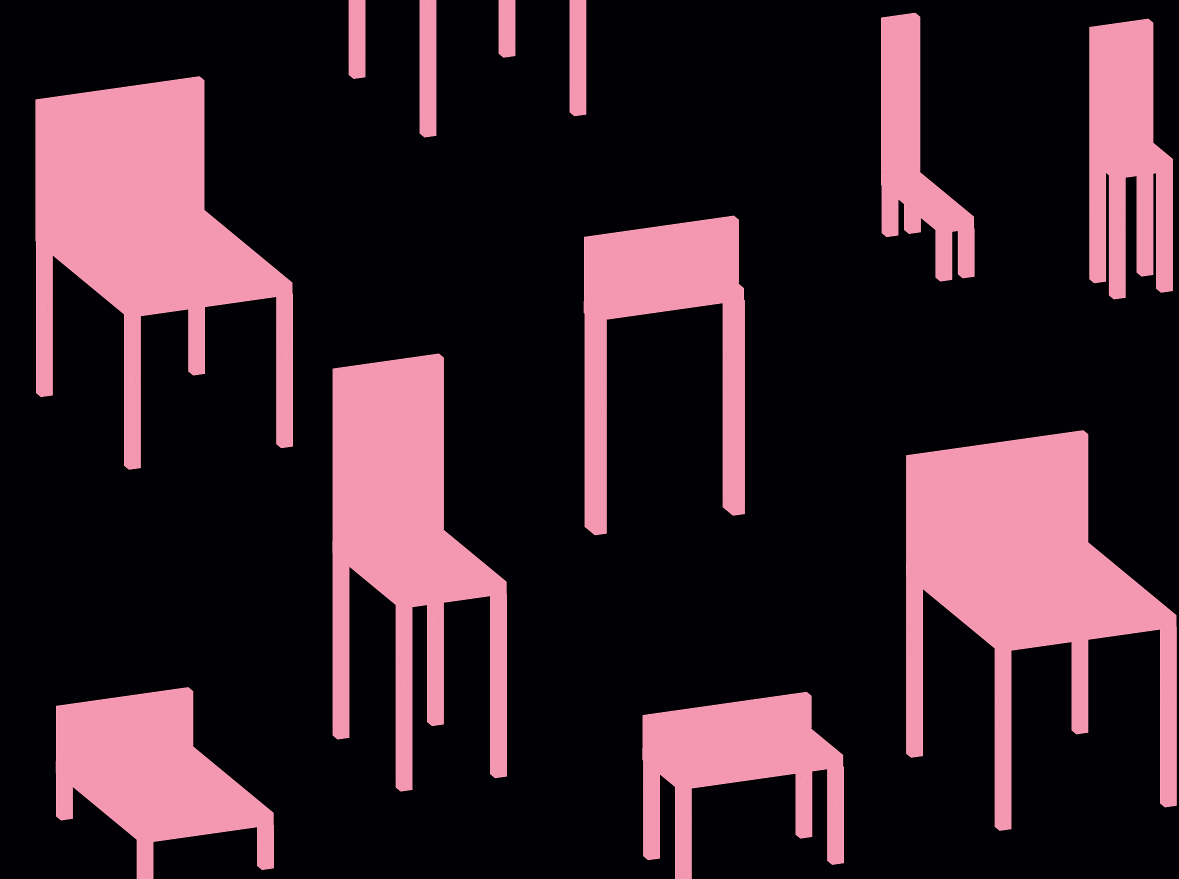
Cell Cycle, by Nervous System, 2009
This console interface is a jewelry design tool for controlling a mesh structure and subdividing cells without editing the code directly. The resulting geometry can be 3-D printed as a ring or bracelet.

CODE EXAMPLES
CHAIR

The first step in creating a parameterized design is to decompose the form into variables. A simple, parameterized chair has variables for the dimensions of the seat, the height of the back, and the thickness of the frame. One unique chair exists for each set of variable values. The design space is the range of all possible chairs that can be generated within the system.

Once the parameters have been identified, they can be used to draw the chair, and after a value has been set for each variable, the code responsible for translating those values into lines and shapes will display the chair on-screen. Choosing values for each parameter at random is one way to explore unexpected chairs within the design space.





Download the code at http://formandcode.com

CODE EXAMPLES
WAVE

Parameters are powerful tools for controlling a series of objects. The behavior or form of each object can be connected to the value of one or more variables. The designer can think about more than just the individual objects—in this case a rectangle—and instead consider how a population of those objects could combine to create a larger form. Here, the rotation of each rectangle is linked to the rotations of its neighbors. This type of parameterization lends itself to the creation of complex patterns.

The pattern begins with a row of rectangles, each slightly rotated to a random angle. The rotation of each row is determined by the rotation of the rectangle above it. At first, the rotations proceed clockwise, but to prevent overlap, the direction reverses if the new rotation value is too large or small.