Epilogue

Three years of ten

It was the American type-designer, calligrapher, and book designer William Addison Dwiggins (1880–1956) that was credited with coining the term ‘graphic designer’. He used it to describe his various activities in printed communications, like book design, illustration, typography, lettering and calligraphy. In his days printer, typographer, writer, and graphic designer were often one and the same person. In the 20th century, all these skills developed into separate skills.

Graphic design is the visual design and organizing of ideas in different media, with the aim to share these ideas with people. This requires the commitment of both artistic and technical skills. And although I’ve worked mostly as a graphic designer I evolved over the years into a user interface designer. In 2004 I lost my job. But in 2009 I came to the conclusion that I was in a luxury but risky position.

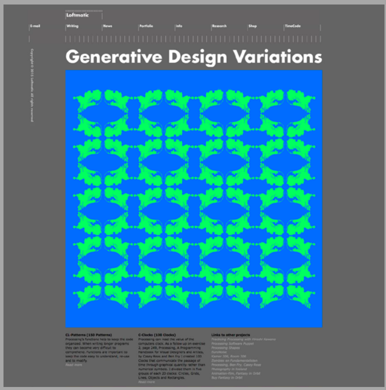
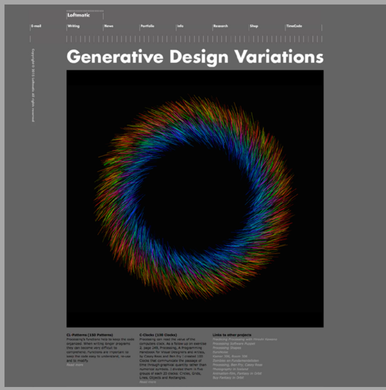
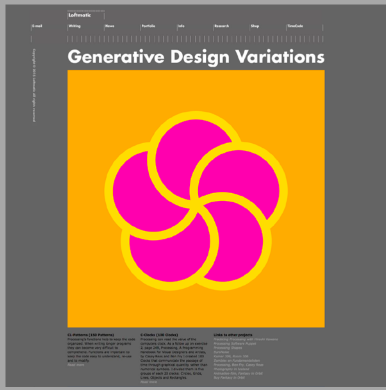
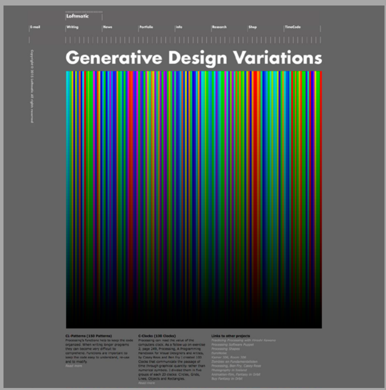
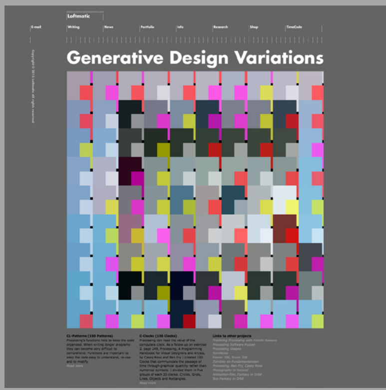
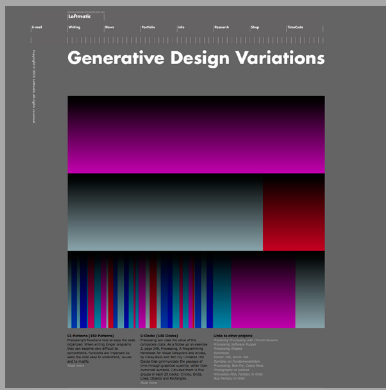
The luxurious position arose from the fact that beside my work I could afford myself to study again. This was not possible when I worked at companies. The position was risky because I wanted to qualify myself in a profession that I did not know. That is why I stated in 2009 that I wanted to update my design process. I wanted to learn to program and apply it in my design process. Which is not a new thing for designers in general, but it was new to me. I gave myself ten years to see if it would work.

The Generative Design Variations project was one way for me to learn programming. All studies are based on the programs that are listed in the Generative Design book. The ten publications plus this one arising from there describing the possibilities and difficulties I encountered. I would like to thank Jeanne for her continues support during this journey.

Teaching yourself programming is not an easy thing to do. And you are never finished with it.

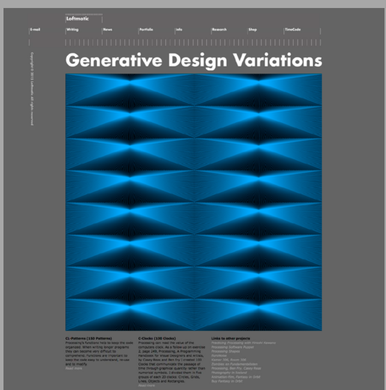
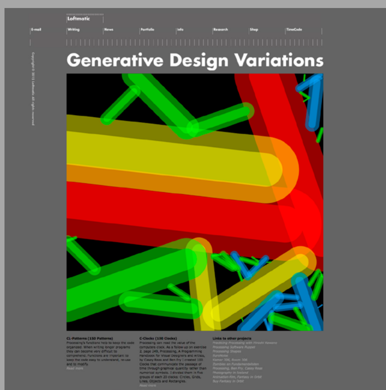
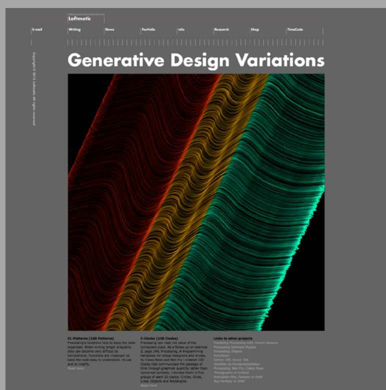
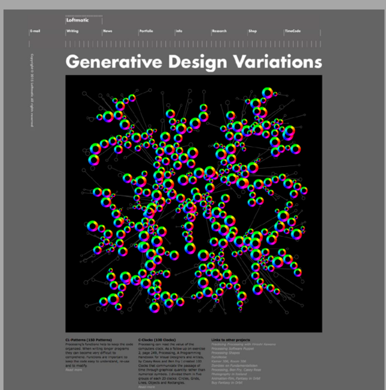
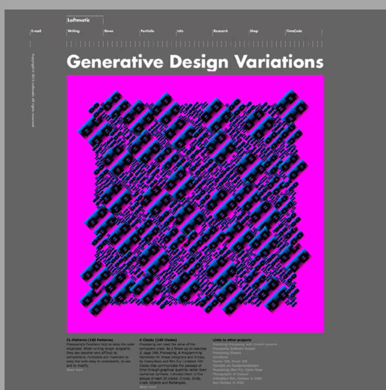
‘If I finish a book a week, I can read only a few thousand books in my lifetime, about a tenth of a percent of the contents of the greatest libraries of our time. The trick is to know which books to read.’

Carl Edward Sagan, 1934—1996, American astronomer, cosmologist, astrophysicist, astrobiologist and author


Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Discussion

Erik van Blokland, Henk Lamers

Could you tell something about how complexity evolves in your work?
I'm not really looking for complexity. It grows spontaneously by itself as I discover how a program works. But I always try to get all what's possible out of the program. Sometimes you get a lot back and sometimes you find only a few things which might be interesting. This partly happens because I am still developing my programming skills, and partly because I know certain things are available inside the Processing toolbox. I think that complexity grows as opportunities present themselves while your understanding of programming grows. Complexity can lead to fantastic results but when I use it in my daily design work, it can also scare your customers. For example five years ago I was asked to design a corporate identity for a new design company. The question was to start with a wordmark. I wanted to use my experience in programming and thought: 'Hmm… let's see how I can make use of those new skills.' When the first proposals were ready I invited my client to show the results. I had generated 100 variations for the wordmark, in a range of series but I could have created a lot more of them. As we went through the results, the client looked at them and said: 'Let me think about it!' She didn't want any espresso nor a cup of tea. She said: I will come back to it. Later that day I received an email: 'Hi Henk, please send me the bill of all the work you've done and let's keep it to this. Apparently it was far to complex for the client to come to a decision and to get involved in the design process. Let alone to be able to categorise 100 designs and select the one that fits best with the philosophy of the new design company. For me this was a big surprise since we humans have to make thousands of decisions throughout the day. A week later I convinced her not to stop but to have another look at the results. I made several selections, added a few things until both of us were very happy with the result.

What I personally like about generative design is that you can generate 100 alternatives, based on one algorithm, resulting in a large amount of results that vary a lot. Some will be good and some will be less good and some can even be terrible. After that I try to discover were the differences come from and direct it all to where I want to go. What's your view on the 100 results?

Sure, some results are better than others. But customers have their own preferences (or as in this case aversion) for specific designs. But I tend to have an overall preference of those 100 results. But as the deadline is coming closer and closer, my preferences get focussed to a specific design. If there is no deadline involved then I create my own limits. I never work directly towards a result but I try to find out what is possible through research and create as many as possible variants. But maybe you are right, you could also call this a way of directing. If there is a problem to be solved, I always keep the solution open as long as possible. It also generates nice invoices ;–)

But to come back on your previous question about complexity of the work. That is often an illusion. It seems complex for the outsiders but when you follow the process and you focus on the subject it is not complex at all. I try to do everything step by step. This has nothing to do with programming but this appears within any kind of art or science.

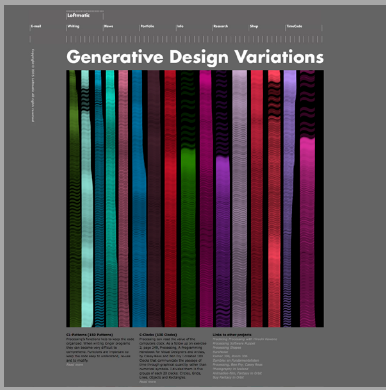
Erik van Blokland is a Dutch typeface designer, educator and computer programmer. He is the head of the Type & Media Master of Design program in Typeface Design at the Royal Academy of Art, The Hague in the Netherlands.

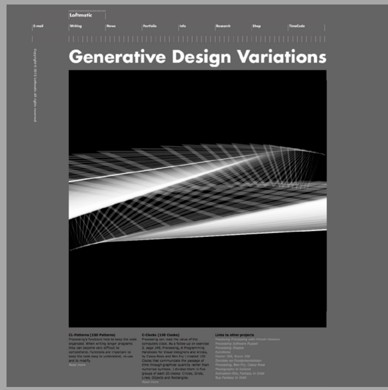
He collaborated with fellow Dutch typographer Just van Rossum to fuse programming and letterform design in a hack to the PostScript programming language. The result was the FF Beowolf typeface, the first dynamically generated typeface, which modified letterforms on the fly.

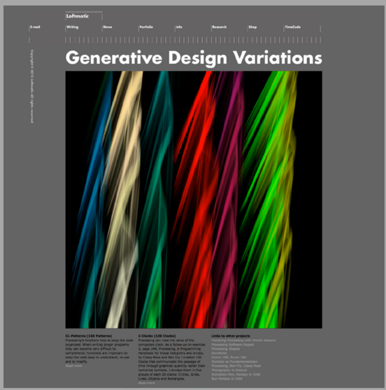
Erik made several programming contributions related to type design. He is the co-author of Unified Font Object (UFO), the open, XML based file format for font data. He co-authored WOFF, the Web Open Font Format. He developed the RoboFab type design extensions for Python with Just van Rossum and Tal Leming. He's the author of Superpolator, the application for interpolating font styles.

Letterror

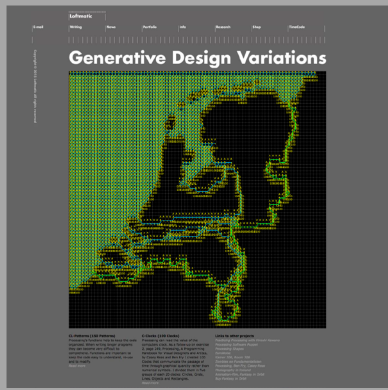

Generative Design Variations

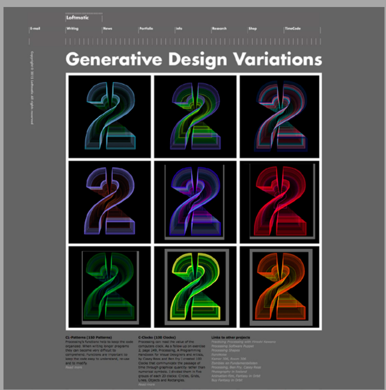

Generative Design Variations

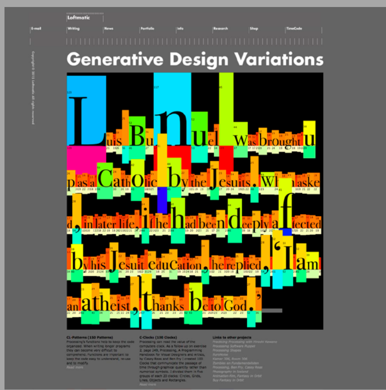

Generative Design Variations

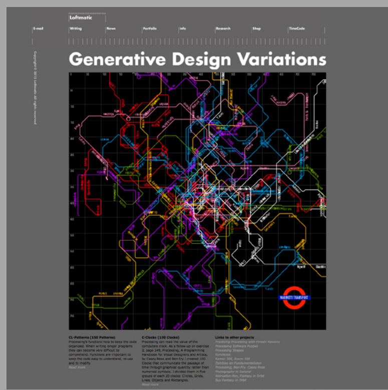

Generative Design Variations

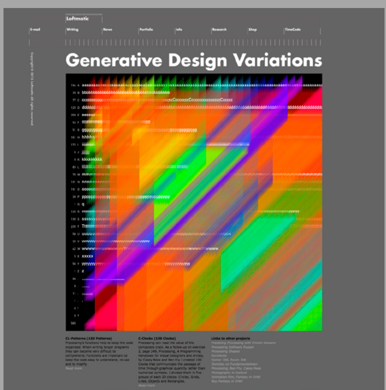

Generative Design Variations

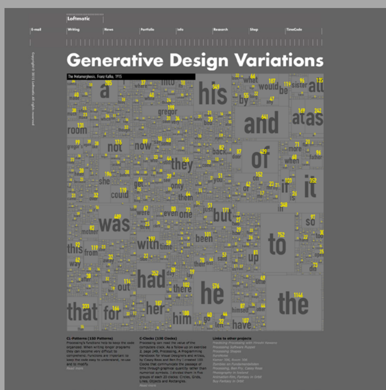

Generative Design Variations

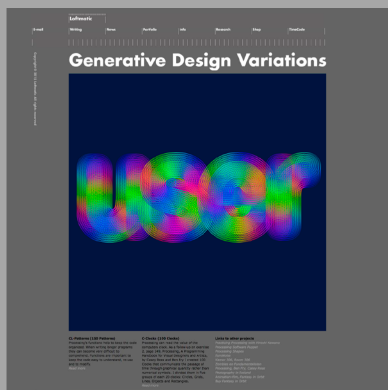

Generative Design Variations

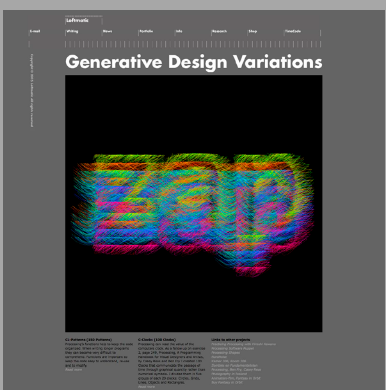

Generative Design Variations

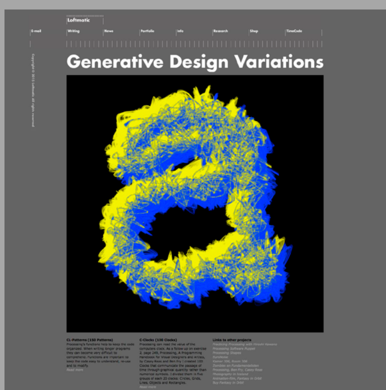

Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations

Discussion continued

Was it different in the beginning?
Completely different... I assume you are talking about the fact that I decided to learn to program in 2009. In the beginning programming seemed very mysterious to me. That was also how I experienced the programmers around me. They did something I couldn't understand. But than there was no reason to understand it. On a certain moment I asked questions to Just van Rossum and he said: 'Henk! You really do not need to learn programming!' No idea why he said that to me. But I will ask him within a short while.

Where do you find the domains for your projects?
I just pick an unknown subject. Last year it was data visualization. The year before it was Generative Design. This year it will be MYNOC, My Nature Of Code. A book written by Daniel Shiffman. It explains how you can use Processing to simulate natural ecosystems. In 2018 I am going to focus on animation and Geometry or Basic Math and Pre-Algebra. After that I will start to create projects where I can combine everything I have learned with programming. I have no idea where I will end but that is also a good thing. I like to keep it that way.

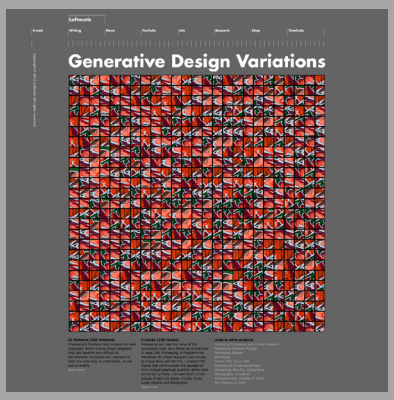
Are you not yet curious after what you can do with animation?
But yes of course I'm curious what's coming up and where I am going to. A part of MYNOC covers animations that you can find on Facebook and on Vimeo, and the still images on Flickr.

Can you tell me something about the results you expect and what in the end will appear!
I expected nothing! MYNOC, for instance, started completely blank to me. I worked through the first chapter of the book. The results were images that I didn't think off or expected. For example: at a certain moment Daniel presents a program where he says that he replaced the QUAD_STRIP's by QUADS because it did not produce good results. I liked to find out what would happen when I did use QUAD_STRIP's. This curiosity lead to the best animation of the entire chapter. Curiosity is very important during programming. You also have to like to think in abstract terms. Which doesn't mean that I am a perfect programmer at this moment.

Of course you can program. Or do you mean that you don't know everything yet? Or that things could be better?
Well I mean ... it's ongoing research. When you have learnt to fish, then there is little more to develop. You fish, try another rod, fish at a different location, just fine tuning, optimising. But programming lets you develop your skills endlessly. And there is a limitless search for what is possible. But of course there will always be people that can program much better than I can. It also depends how you use the knowledge. At Philips Research you will find far better programmers than I will ever be. But they are not good in design. Or the combination of design and programming. I find both skills very important. I would love to be good in both however they are hard to measure. What is good programming and what is good design? When are you good in both, and at the same level? I only know that programming brings solutions that appeal to me. I love the unpredictable factor of it. You do something and you get something else in return which you didn't expect.

Another example: when I worked as a graphic designer at Total Design in Amsterdam, I often had to create instructions for the typesetter. Depending on the typesetter, I received results that hardly looked like what I expected or wanted. In such a case I always asked myself is this better or is it worse then the results I expected. In some cases it was better I would simply move on with my work. Only when it was worse, I made corrections to improve the end result. Sometimes the interpretation by someone else can lead to an enrichment of your idea and I have no problem to admit that. I don't mind who made the interpretation, the typesetter or someone else. But now the interpretations and coincidences come from my computer. I see that as an advantage of programmed design opposite to the analog design process.

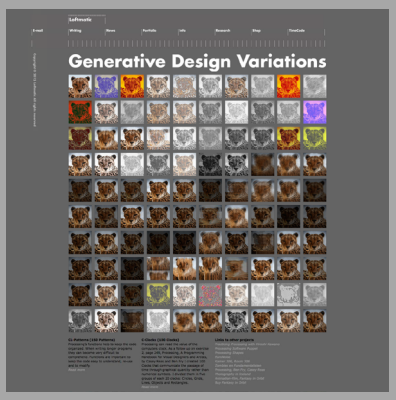
Something else, something very essential did change. I used to make many sketches on paper, but that has stopped. I do write notes now and then but I don't use them often. My form of sketching with programming is that I create several sketches of programs until I feel that the result is good. After that I show them to Jeanne who says: but what if...? And after a while I created more or improved versions of the program or I left it for what it was.


Generative Design Variations

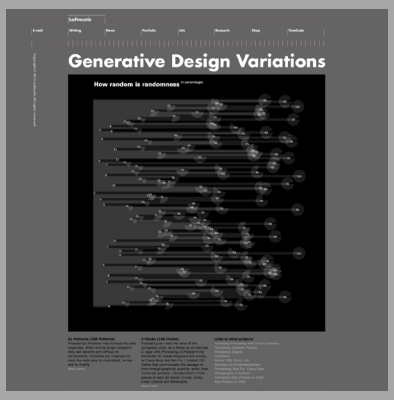

Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations

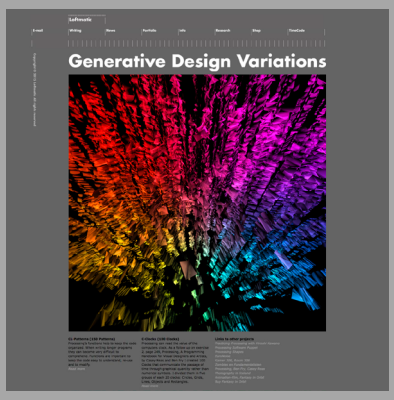

Generative Design Variations
How random is randomness

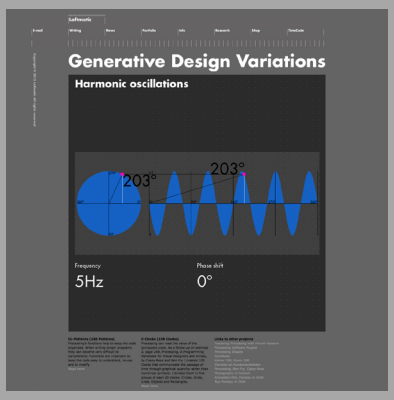

Generative Design Variations

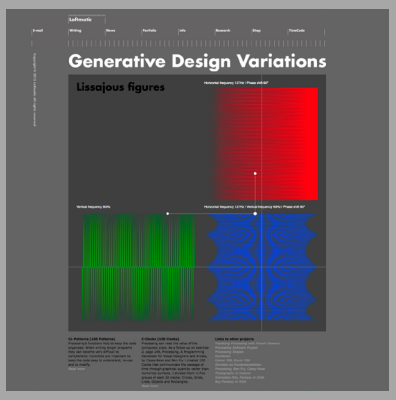

Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations
Harmonic oscillations
203°
203°
Frequency
5Hz
Phase shift
0°


Generative Design Variations
Lissajous figures


Generative Design Variations

Albert Kiefer

I first met Henk and his partner Jeanne well over fifteen years ago when I was connected to a training institute where I taught 3D graphics. We connected very well on our mutual sense of dry humour and our interest in the visual side of computers. At that time, 3D modelling, rendering and a bit of animation in a program called Lightwave.

What I share with Henk is a fascination for the sheer endless possibilities that the medium, or tool, called computer can give a designer or visual artist. Obviously at first you grasp for a hold on the basic concepts in a totally new visual world. Starting out with very simple things that look more or less like all others starting out along that same route. But then you can see a separation starting with people who are quickly bored by standard routines and push button effects.

Henk is one of those gifted artist/designers who, along with a strong talent for creating strong design and image, also has the focused drive to explore that 'what is beyond the current horizon'. But even more important than having a drive to explore is the will to 'stick with it' and really drill down deep and try and master a new skill.

This is exactly what he has done over the last couple of years, since early 2009. He has taken a basic working knowledge of working in the open source program called Processing, working with a laser guided focus with the book Generative Design (by Hartmut Bohnacker (Autor), et al.) and in this way acquire a deep working knowledge of that software to create and explore the very fascinating world of visual design by programming.

Now I would not be talking about this if that was the end of the story... there are countless artists who learn 'something new' every day in their quest to get better at what they do. But Henk has taken a slightly different route. He has decided to meticulously share his experiences with the online community for those that are interested, on his blog (https://mycodehistory.wordpress.com). And not only that, together with designer Jeanne de Bont he also spent quite some time in presenting his findings and results in a very aesthetically pleasing form in a series of ten digital pamphlets that reflect on a topic, provide context, and even add an occasional dry humorous remark to downplay some of the fantastic magic that will be displayed in the pages that follow.

Now, at the end of this particular journey there is the extensive blog online and this volume of ten richly filled documents. Each of them a visual feast and each one excellently designed to present information, image and process in a clear inviting and inspiring manner.

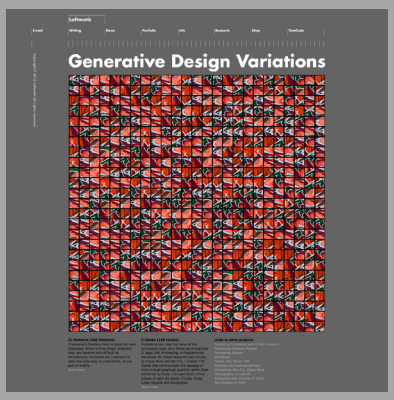
Enjoy in abundance!

Visual Persuasion
'The core of my business is to visualize thoughts and ideas.'

Albert Kiefer has well over 20 years of experience in computer illustration, animation and image manipulation.

He works as a visual designer and technical expert in his field. His work ranges from consulting jobs to creating powerful visuals for multinationals, government and software developers.

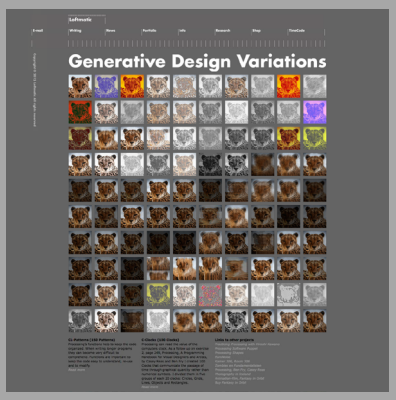
Kiefer Design


Generative Design Variations

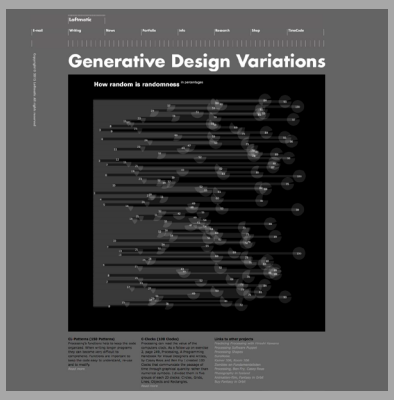

Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations

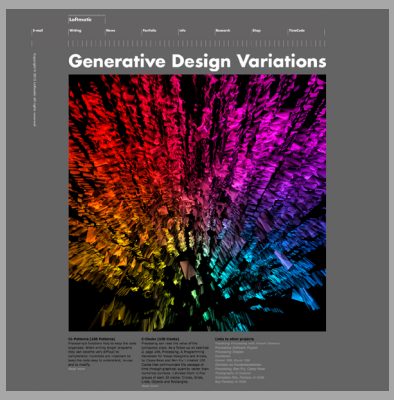

Generative Design Variations
How random is randomness

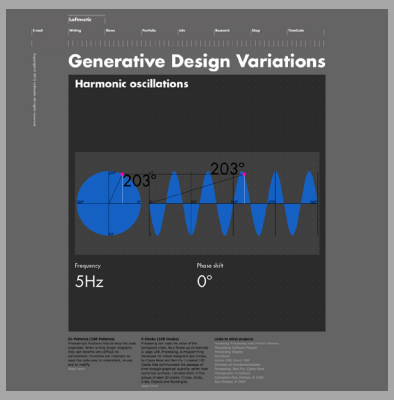

Generative Design Variations

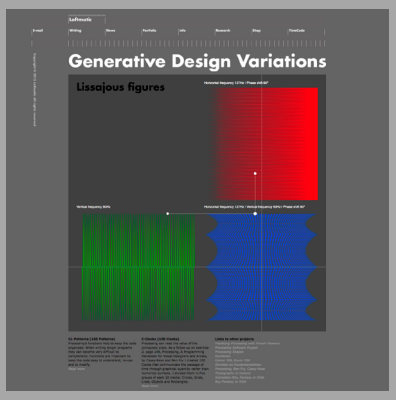

Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations
Harmonic oscillations
203°
203°
Frequency
5Hz
Phase shift
0°


Generative Design Variations
Lissajous figures


Generative Design Variations

Jeanne de Bont

Work till it's right
Before I start to write text, I always make a rough framework of the story. Most of the time this is a great help, so I listed these words: perseverance, huge feeling for aesthetics, able to create strong beautiful images, uncertainty / doubt / but still continue, searching for uniqueness and origin, independence, total freedom, work till it's right, never stop. The idea is that you give shape to the story based on these words. But even with this respectable framework I found it difficult to write something. The subject is so close by and sensitive and there is so much I can write about Henk and about his work. After a few days I tried it again and made this new frame work for a possible story: 40 years of fascination for art, music, movies, photography, traveling and good food, 1999 John Maeda's presentation at TUE, Processing language and book of Ben Fry and Casey Reas, Animation course with Andrew Glassner, Processing class by Daniel Shiffman Amsterdam, Introduction to Generative Art, by Marius Watz at BaltanLab, Generative Gestaltung, Generative design, online publishing, MyCode & MyNoc. A rich and steady framework but still I kept on struggling writing the text.

Just start...
One day Henk read an article stating: you can become anything you like in life, you only need to work hard for 10 years. In the article they gave the example: if you want to play piano, just start practising and after 10 years you are a professional piano player! Most people would freak out if they had to spend 10 years of their life. They would never have the courage, not knowing where to start either. At first I thought Henk was joking but soon I understood that it was serious. I admire his perseverance but there simply was also no other way. He gave himself the 10 years to learn programming. With the arrival of generative design, design changed tremendously. If you add data-enabled design on top of this you end-up in a new design world. This entire area is where you find the current work of Henk and although he was trained as a traditional graphic designer in the seventies, you should actually see him as a design pioneer. You can read all about it on the website MyCodeHistory where he describes how he learned himself programming design.

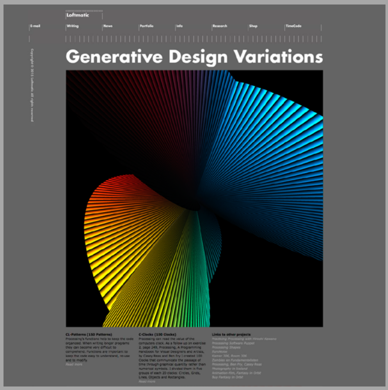
To start off, a broad variety of books about how to learn Processing were ordered and absorbed. Tutorials, online trainings and a life class with Daniel Shiffman and Marius Watz were part of the learning process. Great pain or huge relief the bigger the contrast the larger the sorrow or victory in the end when things did work out. Self-learning also results in self-problem-solving. You get easily stuck when things are badly explained or simply by an error in a piece of code printed in a book. However there was a major breakthrough with the Generative Design book. When you look at the page on Flickr you can get an overview of all the explorations Henk made based of that book. Each stack of images have a high level of quality no matter what the code had to do: manipulation of photographs, creation of patterns, 3D shapes, typography or anything else. In total 66 different topics were explored resulting in 5914 images he selected as being good enough to share with you. On that Flickr page he wrote: 'Generative Design Variations, In this project I will try to make variations on the programs presented in the Generative Design book by Hartmut Bohnacker, Benedikt Gross, Julia Laub and Claudius Lazzeroni. I stick with at least 5 variations of each program and sub-program.'

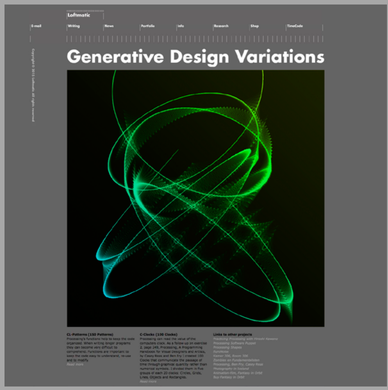
Where I would stop, Henk continues, and keeps on going. Worked till it's right, this attitude gave his work a lot of depth, quality and maturity and that is where his power is. One example: when he was working on the series GDV_M_3_5_1 after struggling with adjusting the code he got it right resulting in an explosion of fantastic 3D strong visual objects in simple green and blue. He created over 230 of these wonderful shapes, every shape, every space around the shapes in perfect balance. I was in a luxury position to admire new great images every day when I got home after work for weeks. Happy days and great memories. After every exercise it got nicely documented and posted on various social media platforms. Not to show off but to share and to inspire others as he was inspired by many coders. The Nature of Code is a book by Daniel Shiffman and the next in line to concur.

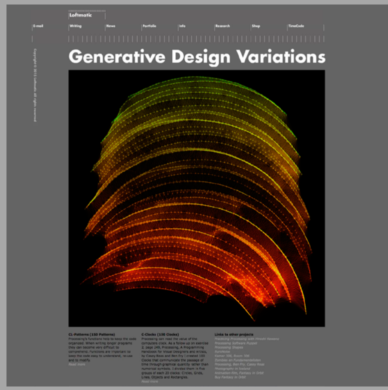
I know that Henk will manage to master programming design and might not even need to 10 years for it.

Jeanne de Bont, creative lead Data Design at Philips Design, started as a graphic designer and developed her self as a animation-, motion-, type-, interaction- and data designer.

I focus on making sense out of data to explore possible new innovative products. My goal is to use multiple data sources to improve healthcare for doctors and patients. By visualizing data in a way that it creates clear and trustworthy insights and better understanding of the best possible treatment of a patient.


Generative Design Variations


Generative Design Variations


Generative Design Variations

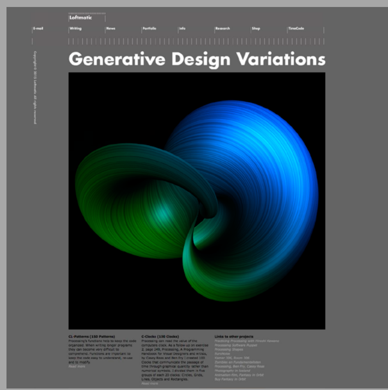

Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations


Generative Design Variations

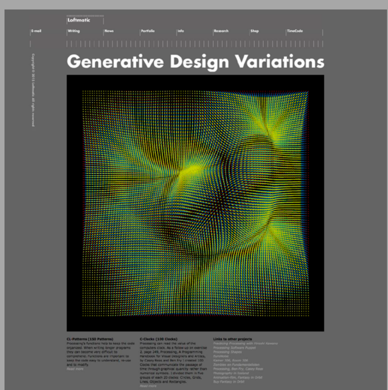

Generative Design Variations


Generative Design Variations

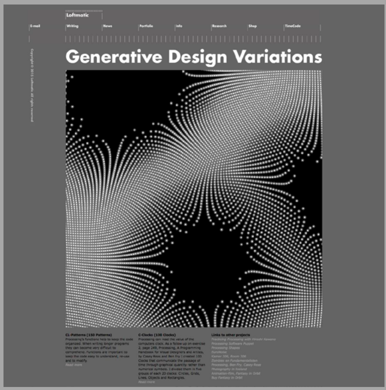

Generative Design Variations

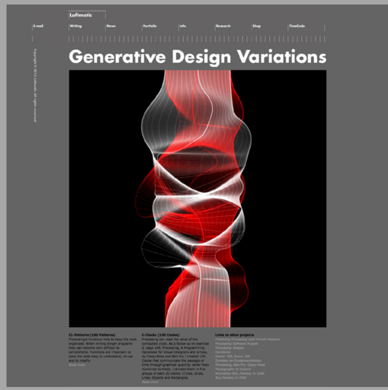
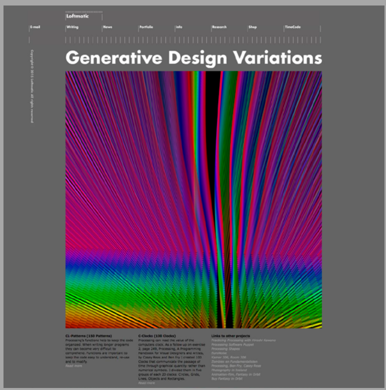
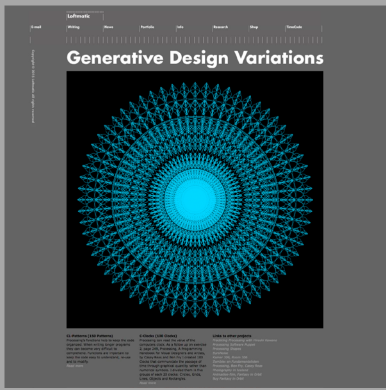

Generative Design Variations

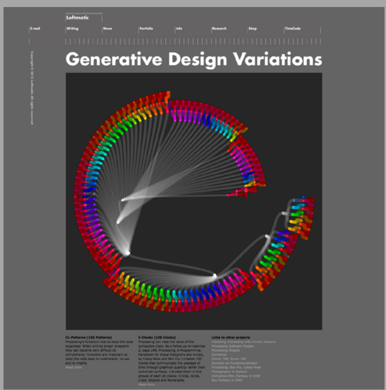
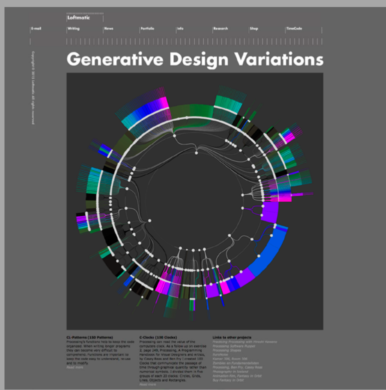

Generative Design Variations

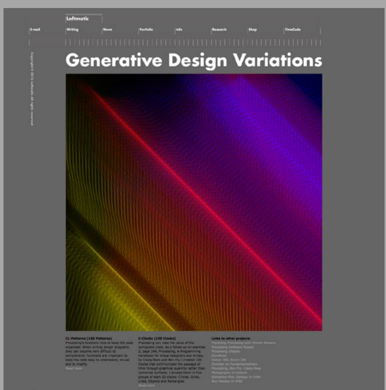
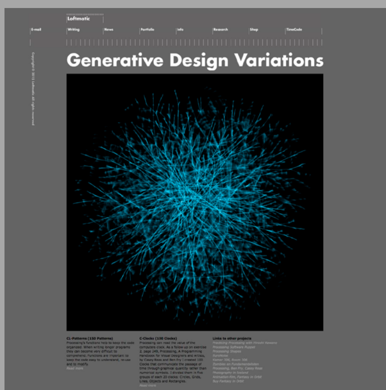
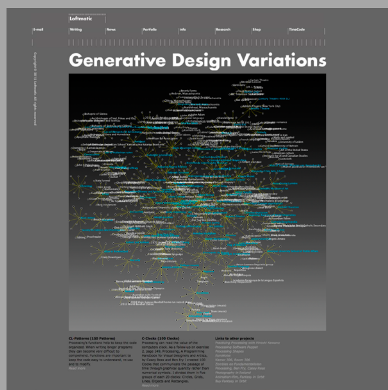
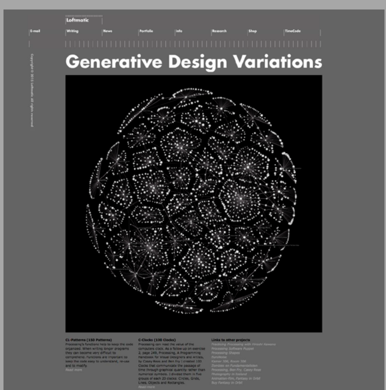
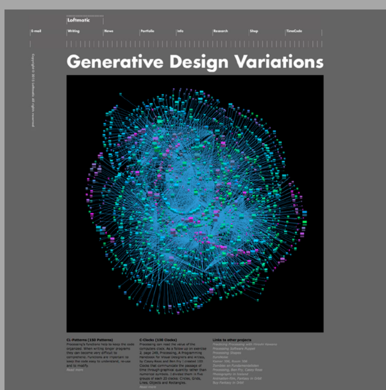
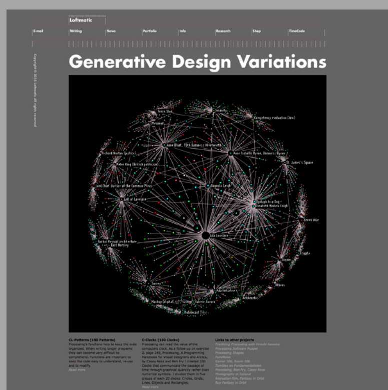

Generative Design Variations


Generative Design Variations


Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

Generative Design Variations

www.generative-gestaltung.de
Special thanks to Julia Laub, Hartmut Bohnacker, Benedikt Gross and Claudius Lazzeroni

Results can be found here:
mycodehistory.wordpress.com
www.loftmatic.com
www.flickr.com

Text and design, Loftmatic Henk Lamers and Jeanne de Bont