Marius Watz SOFTWARE ARTIST, LECTURER

mariuswatz.com
marius@mariuswatz.com
oslo / new york

WORK mariuswatz.com
BLOG generatorx.no
PARTNER randomnumber.nu
TWITTER twitter.com/mariuswatz
FLICKR flickr.com/photos/watz

Closed systems: Generative art and Software Abstraction

- System: "1. A group of interacting, interrelated, or interdependent elements forming a complex whole."

- Closed system: "An isolated system having no interaction with an environment [...] a SYSTEM whose BEHAVIOR is entirely explainable from within, a system without INPUT."

void setup();

The notion of generative systems, where formal constructs are described in terms of parametric processes executed autonomously of their author , has recently gained popularity in art and design. Conceptually these systems provide a computational model of creativity, combining principles of unpredictability with the purity of logic. As Constructivism and Futurism attempted to invoke the possibility of a world view based on industrial processes, so generative art presents us with a Weltanschauung of computation.

Forms produced by generative systems often take on a complex nature, exploiting principles of emergence to produce structures that could not be made by human hands. Inspiration taken from processes found in nature is common, with the tension between organic and mechanical forms ever-present. A common challenge in computational aesthetics is the simulation of organic behavior and spontaneous irregularities, phenomena that appear in nature without prompting but which can only be replicated by computers with the explicit encoding of such behavior.

On a more pragmatic level, generativity is a useful strategy to harness the power of the computer, performing rote tasks and crunching numbers. By using parametric processes to produce an infinite series of possible outcomes, the author is allowed to take the privileged position of harvesting the most successful results. Most generative work is marked by a tendency towards formal complexity made possible by having software filling in the details. The difference between having 10 and 10 thousand particles interacting is a simple adjustment of parameters and allowing for additional computing time. Increases in processing power is certainly a factor, allowing ever more complex computations to be used even for realtime applications.

This is not to say that the use of parametric systems is trivial or simply a matter of powerful hardware. The process of abstracting aesthetic processes into executable computer code requires formal stringency and a talent for reverse-engineering a desired result in order to identify its causal elements. Every aspect of the system must be explicitly described, including details that might seem insignificant when considered individually. When considered as a whole, however, these series of decisions become the very body of the work. Furthermore, even the most experienced programmer will encounter unexpected results in the process of designing a process, whether as byproducts of errors in the code or tendencies given by the algorithms used. Programming errors can lead to serendipitous discoveries, and one does well to embrace one's mistakes.

A piece of software can be written in a dozen different ways, each with subtly different biases. One strategy might yield stable and predictable behavior that falls just short of being aesthetically interesting. Another might lead to an unstable system that produces frequent crashes but also occasional displays of

genius. Despite the essential immateriality of computer code algorithms nevertheless display material properties, often showing a specific bias towards certain outcomes. Generative art requires that the artist be able to express herself through the manipulation of these systems, choosing computational strategies and appropriate parameters in a combination of technical skill and aesthetic intuition.

void loop();

In popular discussion of generative art two aspects are often forgotten: Firstly, that it does not in fact constitute an art movement as such. Rather it describes a strategy for the invention of works that share a certain methodology, but which may present themselves in myriad ways. Secondly, the aesthetic application of rules comes with a wealth of historical precedents, dating back as far as humans have been known to employ scientific principles. The fascination with systems has been a constant of human culture, from early astrology to the mechanical automata of medieval times. These observations are important because they point to a weakness in the current discourse while simultaneously providing a possible solution. By looking at how generative art differs from classic media art, a clearer understanding might emerge.

The current popularity of generative art can be traced historically to several factors: The introduction of personal home computers in the late 1970's with their easily accessible programming languages like BASIC and LOGO provided many artists with their first taste of computation. Later, the near-complete invasion of creative production by digital tools from the early 1990's onwards meant that the transition from simple tool use to exploring code as material became a logical next step. Ironically, it was the introduction of relatively primitive code frameworks like HTML, JavaScript and Flash that jump-started the interest in code-based aesthetics.

These technological developments, coupled with the World Wide Web as a sandbox for personal expression, produced an explosion of web sites in the late 1990's, quickly becoming an international subculture for digital media experimentation. Coinciding with (but conceptually separated from) the rise of net.art, a particular subsection of these sites concerned themselves with software abstractions. Ranging from simple interactive soundtoys to more complex generative compositions, these early experiments should be considered the direct predecessors of today's generative art scene.

While much of the early work of the 1990's was naive both conceptually and aesthetically, it did establish an important distinction from the interactive artworks that dominated the media art scene at the time. Interactive art exploits the feedback loop of interaction between a system and its user(s), with custom software systems generally considered a necessary evil rather than an end in itself. Generative art is primarily interested in closed system, self-contained constructs investigated for their formal and material qualities. This might seem like a trivial difference, but it places the concerns of generative art closer to traditions of drawing or painting than to the relational aesthetics so common in the media art field.

From the more traditional art world obvious connections can be made between generative art and movements like Conceptual Art, Minimalism and Op Art, both in terms of formal and conceptual similarities. Sol Lewitt's use of terse text instructions as the means of describing his famous wall drawings has become something of a golden standard , an art historical reference providing generative art with mainstream legitimacy beyond comparisons to screensavers and audio visualizers. Similarly, the strictness of Minimalism with its elimination of subjective gesture is inevitably appealing to the computationally minded.

A direct link is found in the work of early computer art pioneers like Charles Csuri, Manfred Mohr, Vera Molnar or Frieder Nake. Created in the hey-day of Minimalism and Op Art, their explorations of parametric processes predates the current scene by more than 30 years. Along with a larger group of artists experimenting with computer code they laid out a conceptual groundwork that was sadly semi-forgotten and ignored by media art discourse, until its "rediscovery" in recent years. Generally created before the advent of interactive screens and mostly realized using plotter hardware, artworks from this era mirror the ideas popular in painting of the time.

These pre-existing art movements certainly provide part of the conceptual framework for code-based work being created now. But to get the full picture one must address the radically different cultural context of the current scene to that of the late 1960's. The utopian world view of Modernism has been fragmented by Postmodernism and complexity theory, undermining the straight-forward world view of

reductionist science with quantum uncertainty and emergent phenomena. On a technical level, early computer systems exhibited only a fraction of the complexity of today's technological infrastructure, which goes beyond the individual computer to encompass an interconnected world of networks and public APIs. A flourishing Open Source scene now supports artistic endeavour, serving up tools and code examples to accomplish all sorts of computational miracles.

Developments in electronic music have provided us with new compositional ideas such as sampling, glitch and microbeats, as well as a renewed desire for synaesthetic experiences. Concepts like live cinema and media facades are closely linked to generative art, as are other fields such as information visualization and computational architecture. Only by considering all these developments along with the historical precedents can one glean an understanding of the concerns at hand.

void reboot();

I would like to propose that we are currently at a crossroads in the field of generative art. Code-based artworks have reached a level of maturity, going beyond simple visual experimentation to expressing more complex visions. Artists like Lab[au] use software processes as an integral part of their work, formulating an artistic project based on the material qualities of computation. Throughout their various projects one can trace the merging of architectural concepts of space with code structures.

The "chronos" series takes a simple mapping of time to color as its starting point, following its logic to provide a visualization of temporal space. The generative art consoles Lab[au] have developed are simultaneously beautiful objects and an extendible delivery platform for software artworks. But perhaps their most complex achievements lie in projects like "5x5x5", where generative principles are manifested in physical form, escaping the screen altogether.

This move beyond projections and the screen as mediating surfaces is one of the more exciting recent devel opments in computational systems. The use of digital fabrication technology to literally extrude virtual objects into physical space challenges the screen as a default output device, providing the means for an algorithmic conception of space. Already we are seeing architects using scripting in CAD software to design parametric structures that can respond to environmental input. Meanwhile, artists are experimenting with "data sculptures", representing normally intangible information flows as physical manifestations.

While generative art is inextricably linked to the computer as a means of production, the work is not about the computer itself. While screen-based work and the investigation of realtime self-contained systems remain an important aspect of generative art, it would be a mistake to think generative work is primarily expressed in pixels. I for one look forward to an extended rethinking of computational aesthetics that encompasses a much wider range of possible outputs.

Marius Watz, May 2010
http://mariuswatz.com