Gabriel Menotti Gonring
Goldsmiths College, London

Exhibition Design as High-Level Programming

Keywords: exhibition design, art circuits, montage, curating, programming.

The present paper analyzes some effects caused on the fields of art and cultural production by the technologies loosely grouped under the rubric of *new media*. This subject is far from original: much has already been debated about algorithmic programming as a tool for aesthetic creation, as well as a way to systematize and display ongoing processes. We have come to accept software as an ordinary artform, and online sites as appropriate exhibition spaces. Programming architectures such as *Processing*,[i] developed after MIT's project *Design by Numbers*,[ii] have been created specifically for producing of art and teaching artists to code. On the top of all this, respected curators such as Peter Weibel and Christiane Paul are making public efforts of setting up parameters for this new, emerging creative field.

The incorporation of new media within the art world can be confirmed in its acceptance by mass audiences and traditional venues. For instance, the exhibition 'Algorithmic Revolution – On the History of Interactive Art', programmed to stay just one year at Karlsruhe's *Zentrum für Kunst und Medientechnologie*, achieved so big success that its duration was extended for four more years.[iii] Even the time-honoured Whitney Biennial, one of the most influential events to the global trends of contemporary art, established in 1918, has been running an exclusive web-version since 2002.[iv]

In spite of these circumstances, not much interest has been risen in how the bare existence of the so-called new technologies affects the whole art system and obliges us to re-evaluate traditional dynamics and practices. Contemporary media enlarge the potential field for symbolic production, disrupting its boundaries. It is important to consider, for instance, that many new media artworks are but a designated solution for displaying data – we could say, a form of *exhibition* in themselves. According to

theorist Lev Manovich, it is the *computer* that transforms this mode of representation from exception into norm[v] – a semiotic shift that has probably to do with the fact that the symbolic output of a computer always is, first and foremost, a real-time effect of its computation.

Let's consider for example a piece such as the *Electric Sheep* screensaver,[vi] by Scott Draves. This work consists of a very complex fractal animation generated by distributed computing. The machines that have the software installed communicate with each other every time it runs, calculating together its visuals. At first sight, a spectator might believe that these images have the same integrity of a painting, being a simple, non-composite object. However, what is being shown in the screensaver is the process of evolution of an assorted robotic population. *Electric Sheep*'s seemingly concrete, closed form, is in fact the representation of the open interaction of discrete elements.

In the same measure that Electric Sheep visuals are a representation of this compound system, the artist's work consists of setting parameters for these different elements to be displayed together; i.e. *designing their integration.* Therefore, we could argue that this form of creation resembles more organizing a gallery space than painting a portrait. Albeit the result may look like an abstract canvas, programming conveys meaning and value in a way similar to the process of *curating*, the organization and displaying of selected pieces to achieve a desired meaning or effect. Not surprisingly, the processes Lev Manovich points out as key operations of new media – selection and composing[vii] – are fundamental activities of a curator's *métier*.

This analogy can lead us to several inquiries about the nature of art creation, evaluation and consumption within a new media environment. In the first place, since the basic operations of curatorial work became familiar as such and got over-popularized, it is expected that the curator's role and authority to be put into question.

This paper, however, will attempt to work on a more positive hypothesis: assuming the same parameters that allow us to consider software development as art, we intend to assert that designing an exhibition is not only a mere systemic expedient, but *an equivalent form of aesthetic creation in another level*. In order to do so, we aim to revise the concept of curating and the scope of curatorial process, in order to create a

common epistemology where software, artworks and exhibitions can be related and compared according to their specificities.

Programming languages

Computer programming can be done in as many ways as there are programming languages available. Of course, each tool is more appropriated for a particular situation. Some dedicated codes favour the creation of complex algorithms, but lack versatility; others may have a wider scope of applications, but depend on the meticulous articulation of several operators.

These different frameworks are normally classified according to their levels of *abstraction*. Lower-level languages are those closer to the pure ordination of the electronic flux that runs within a computer; in raw machine code, for instance, a command or value is directly translated into a given voltage in the computer system. Their signifiers can be so *material* that a layman make mistake them by the physical platform in which they are inscribed, as in the case of the popular misconception that some of the first computers were programmed with hole-punched tapes. In fact, these machines were programmed with the ASCII codes these tapes carried.

The higher the level, the further a language gets from the essential process of computation. While the lowest codes tend to correspond directly to the machine structure, the most abstract ones operate with very well defined functions, and tend to mimic natural human languages. According to N. Katherine Hayles, some of the most evolved programming languages, object-oriented frameworks like C++, even 'create a syntax using the equivalent nouns (that is, objects) and verbs (process in the system design)'.[viii] Since they comprise a familiar set of elements and rules, these languages 'allow programmers to conceptualize the solution in the same terms used to describe the problem'[ix] – in other words, they bridge the gap between the engineering and the efficient logics of computation.

That means computer programs do not have to be composed in incomprehensible machine language. High-level code makes it possible for them to be created as abstract

instructions that will be later interpreted as binary formalities. By using intermediating mechanisms – *compilers*, *interpreters* or *virtual machines* – the abstract orders are translated into bit patterns to be applied directly to the processor, where computation really happens. Then, as the system runs, the bit patterns are re-abstracted and the algorithm manifests its results as surface effects.

By and large, as programming languages go through this gain in *semantics*, they gave up on machine-like *syntax* – e.g. the highly fluid vocabulary of an object-oriented language does not have to be articulated in the same precise manner as a set of two, ten or sixteen low-level operators, in order to produce the same algorithm.

As the ratio between the operations of selection and composing seems to change consistently throughout the programming scale, it could be used to characterize the difference between languages. In lower levels, composing seems preponderant: the precise organization of a few functions and variables. At higher ones, it is the practice of selection that matters the most: to define, among several available objects, which is the most appropriate one for the intended function. This divergence of paradigms is hinted by the very nomenclature of certain languages: while the highest codes are often *object-oriented* (a name that characterizes the integrity of discrete operators), among the lowest ones we find *assembly* and *procedural* languages (a name that characterizes their process of integration).

Programming visuals

As society is permeated by digital technologies and becomes intertwined with informatics systems, computer architectures seem to be integrated to those of symbolic production in a common epistemological chain. That means not only that cultural objects become codified in immaterial patterns, but also that the operations that define software manipulation can be found 'at work in the culture at large'.[x]

One of the most obvious examples of this influence are the forms of creative expression that depends mostly on the coordinated operation of databases, such as audiovisual remixes. In the same manner a computer program is built upon the articulation of

immanent functions, variables and objects, a video work may be composed by the mere conjunction of discrete audiovisual samples captured from other movies. That is the case of *Pixel Pirate II: Attack of the Astro Elvis Video Clone* (2006),[xi] a feature film made by Australian collective Soda_Jerk without any original audio or video footage: it was entirely 'pirated from over 300 film and music sources'.

The difference between Pixel Pirate II and a work made through traditional filmmaking process could be put in the same terms used to compare high- to low-level programming: whereas the later is created by the sophisticated operation of simple elements (such as light and sound), the former results from the combination of highly-abstract symbols (in that case, movie scenes). Moreover, it should be noticed that these highly-abstract symbols actually *derive* from the regular filmmaking practice, just as a low-level framework establishes the basis for higher-level programming in computer architectures.

As processing machines, computers 'modify, move, and combine symbols at a low level to construct higher level representations.'[xii] In that sense, their architecture resembles the way moviemaker Vsevolod Pudovkin characterizes filmmaking: 'the construction of a scene from shots; of a sequence from scenes; of a part of the film (a reel, for example) from sequences; and so on and so forth'.[xiii]

We should not ignore that Lev Manovich's theorization of new media was highly influenced by Russian constructivist cinema. One of his greatest inspirations was Dziga Vertov, who once described the practice of montage as 'an inventory of documents that is related to the theme […] put into a rhythmic order in which the concatenation of meaning coincides with the concatenation of visuals'[xiv] – a concept that could be easily summarized by the pair selection/composing. Therefore, in as much as this logic that drives both programming and filmmaking practices is favoured by digital technologies, it is likely to exist prior to new media.

Nevertheless, it seems that a scale of symbolic systems according to their level of abstraction may help us understanding how different forms of creative expression are inter-related within the contemporary media ecologies. In order to constitute this fractal paradigm, we could start with the regular scale of programming languages and rethink

its limits; even though the lowest position would still be occupied by the essential informatics mechanism, the highest one would expand beyond human language, towards the cultural *milieu* that circumscribes it.

Programming exhibitions

Almost in the threshold of the scale proposed above, we will find *exhibition design* – an activity that has always consisted of selection and composing, but has seldom been considered a form of creative expression beyond simple filtering. Nevertheless, even though a curator does not directly interfere with any artwork, he affects their value by placing them in relation to each other and the space. Gathering and arranging some *ouvres* (as well as excluding others), he creates a symbolic system that instantly becomes the context inside which these *ouvres* are appreciated.

Nicholas Serota, Director of Tate Gallery, describes this as the *principle of interpretation* – 'one of the fundamentals that has underwritten curatorial practice since the mid-nineteenth century.'[xv] Such practice involves 'combining works by different artists to give selective readings [...], establishing relationships that could not have existed in the minds of the makers of these objects.'[xvi] By doing so, the curator promotes the integration of the discrete artworks into the concrete system of the exhibition – 'a matrix of changing relationships to be explored by visitors according to their particular interests and sensibilities.'[xvii]

As we can conclude by Serota's description, this process of integration is only fulfilled during the trajectory-to-be of the public, which the curator tries to shape by articulating physical constraints and the works themselves. Therefore, as a mode of creative expression, exhibition design must presuppose the dynamic relation between the public and the gallery space, actuating precisely in the domain of what philosopher Paul Virilio calls *trajectivity*. Virilio conceives the *trajective* as *a state of being in motion*, an intermediary condition between subjectivity and objectivity.[xviii] Through selection and composing, exhibition design intends to respond to this mobile situation by disposing the works and setting up a *display*.

This ability to *display* – that is, to define *what is meant to be seen* and *how it must be seen* – has always been connected to social dynamics power and prestige, being in different times affiliated with the aristocracy, the State and the capital.[xix] From this perspective, it is easy to understand why exhibition design has been elevated to a systemic, presumably transparent expedient of the art circuit – a 'completely natural interface.'[xx] The curator has a paramount role in the art world economy, similar to that of the editor in printed media: by embodying authority itself, he determinates the value of artworks and controls their diffusion. The curator tells *producers* apart from *consumers* and mediates the relations between them. In order to do so, he has to be *above* the artists – that is why, in the traditional art circuit, exhibition design must be considered essentially different from artistic production.

On the other hand, new media circuits – whether Web 2.0 websites or free software development communities – dispense the role of the curator by making it over-necessary; the amount of data circulating in these environments makes it impossible to restrict the activity to a few specialists. As practices of filtering and data-mining turn to be vital to the system operation, they also become more vulgar and must be easier to perform. Therefore, new media may inspire a re-evaluation of the concept of exhibition design that would deeply affect the present balance between art production and consumption, increasing the public awareness of the curator activity.

An increasing number of artists is assuming the role of curators as a natural expansion of their creative production. The aforementioned Soda_Jerk collective, for example, regularly organizes exhibitions and video screenings. They see curating as 'an extension of our remix art practice. Like cutting a mix tape, curating involves bringing together different works and forging new connections between them.'[xxi]

It is also interesting to take notice of the recent popularization of art pieces whose mode of production involve curatorial practices, often resulting in large exhibitions or databases, such as *Learning to Love You More* (Miranda July, Harrell Fletcher and Yuri Ono, 2002-2009).[xxii] This project consisted of a series of assignments posted to a public website – activities such as "draw a constellation from someone's freckles", "write your life story in less than a day" and "re-enact a scene from a movie that made someone else cry". Anyone could complete any proposed task and send the results back to the artists.

In that way, each assignment turned into an online gallery – small collections of works of the most diverse formats: audio recordings, videos, texts, photographs, drawings – that afterwards unfolded in non-web presentations.

These examples intend to show how, far from being an innocuous form of interfacing users and data, displaying can be used as a strategy for creating complex, meaningful interactions. In both cases, artists are operating like editors, establishing criteria for the production, selection and exhibition of the work of others. They do not produce the pieces – they simply dispose them together. Exhibition design can be said to be this practice of organizing highly abstract symbolic objects in even more complex symbolic systems – an activity that seems to fit very well the expanded hierarchy of programming architectures we have proposed above.

Nowadays, when we talk about *programming* a movie theatre, we are not far from the literal truth – after all, digital screenings involve composing playlists, coordinating networked data transfers and encrypting movie files. On the other hand, software developers are constantly obliged to foresee the highest symbolic levels in order to perform their work. Is not there a traditional curatorial work in the creation of a Linux distribution, for example – an effort to balance the selection of applications and the system's performance?

Biography

Gabriel Menotti Gonring is an independent media curator and producer. He holds an MA in Communication and Semiotics by the Catholic University of São Paulo and, at the present time, is a PhD candidate at the Media and Communications Department of Goldsmiths, University of London.

[i] *Processing 1.0*, http://www.processing.org (29 September 2009).
[ii] *Design by Numbers*, http://dbn.media.mit.edu (29 September 2009).
[iii] *Algorithmic Revolution - On the History of Interactive Art*, http://www.zkm.de/ algorithmische-revolution (29 September 2009).
[iv] *Whitney Biennial 2002*, http://www.whitneybiennial.com (29 September 2009).

[v] Manovich, L. (2002) *Data Visualisation as New Abstraction and Anti-Sublime*, http://netart.incubadora.fapesp.br/portal/referencias/data_art_2.pdf (29 September 2009).
[vi] *about electric sheep*, http://community.electricsheep.org (29 September 2009).
[vii] Manovich, L. (2001) *The Language of New Media*, Cambridge: MIT, p. 118.
[viii] Hayles, K. (2005) *My Mother was a Computer – Digital Subjectivity and Literary Texts*, USA: University of Chicago, p. 57.
[ix] Ibid.
[x] Manovich (2001), p. 118.
[xi] *Soda_Jerk*, http://www.sodajerk.com.au/sj/ppii.html (29 September 2009).
[xii] *FAQ \ Processing 1.0*, http://www.processing.org/faq.html (29 September 2009).
[xiii] Pudokvin, V. (1983) 'Métodos de tratamento do material (Montagem estrutural)', *A Experiência do Cinema*, Xavier, I. (ed.), Rio de Janeiro: Graal, p. 57.
[xiv] Vertov, D. (1983). 'Extrato do ABC dos Kinoks (1929)', *A Experiência do Cinema*, Xavier, I. (ed.), Rio de Janeiro: Graal, p. 263-264.
[xv] Serota, N. (2000) Experience or Interpretation – the Dilemma of Museums of Modern Art, UK: Thames and Hudson, p. 9.
[xvi] Ibid.
[xvii] Id, p. 55.
[xviii] Virilio, P. (1993) *O Espaço Crítico e as Perspectivas do Tempo Real*, São Paulo: Editora 34, p. 107.
[xix] Cummings N & Lewandowska M. (2000) *The Value of Things*, Basel: Birkhäuser – Publishers for Architecture, p. 32.
[xx] Manovich (2001), p. 178.
[xxi] *Soda_Jerk*, http://www.sodajerk.com.au/sj/curate.html (29 September 2009).
[xxii] *Learning to Love you More*, www.learningtoloveyoumore.com (29 September 2009).