

Alexander R. Galloway

Jodi's Infrastructure

01/06

e-flux journal #74 — june 2016 Alexander R. Galloway
Jodi's Infrastructure

One might query any contemporary artist and, as a kind of litmus test, ask the following series of questions: Do you think of yourself as primarily working “on” the digital or primarily “within” it? Is the computer incidental to your work, a tool like any other? Or is the computer at the heart of what you do? Shall art orient itself toward the digital? Or shall art merely live inside the digital, while concerning itself with other topics entirely?

Digital aesthetics can refer to the “medium” of the digital, that is, all the tools and technologies that populate contemporary life. At the same time, digital aesthetics can refer to context, that is, a “digital context” or a “net condition” – the latter being the title of an influential 1999 net art exhibition at the ZKM in Karlsruhe. Artists have their own particular ideas about digital aesthetics, of course, as do computer scientists. Sometimes these ideas overlap and sometimes they don’t. Can digitality be beautiful? How hopeless a question to pursue: it depends on so many complicated things, not least of them the definitions of digitality and beauty.

Technologists tend to wrestle with similar issues. Some programmers or engineers think of the machine as a tool to be used in pursuit of some larger design strategy. Thus there are many workaday technologists for whom digitality is a “context” or “condition,” with all of its attendant issues, from proletarianization and exploitation (be it unpaid overtime in Silicon Valley or harsh working conditions at Foxconn) to new forms of empowerment via social networking and communication in the public sphere. Still, when technologists reflect on themselves, when they narrate their own project, they tend to favor medium over context. I’m thinking of a text like *The Art of Computer Programming*, Donald Knuth’s monumental treatise on computer science. Here “art” is an entirely self-referential activity, and beauty is defined through the virtues of functionality, elegance, and simplicity. Context still matters, of course, but code derives its beauty, its very identity, from an analysis of function and its accurate formalization in logical and mathematical structures. (G. H. Hardy’s classic hymn to pure mathematics, *A Mathematician’s Apology*, is the literary obverse to Knuth, but it promulgates a similar set of virtues.)

Two basic activities emerge. A person may work “on” the digital or “within” it. In the former, one’s attention is directed from the outside in, taking the medium itself as its object, while in the latter one takes the perspective of the medium itself, radiating attention outward to other contexts and environments. To generalize from this, the first position (working “on”) is labeled modern or, when applied to art and

aesthetics, modernist. And the latter position (working “within”) is labeled non-modern, be it premodern, postmodern, or some other alternative.

In the modern “on” mode, infrastructure is everything. Content dissolves into context, and context itself becomes content. Hence the great mantra of modernity is “there is no content” – or, as Marshall McLuhan famously put it, “the medium is the message” – since all content is overwhelmed by context. By contrast, in the non-modern, premodern, or postmodern mode of working “within,” content is what it is, no more and no less. Here content provides its own context, and the environment grows in accordance with the emergent emanations of the inside. No larger transcendental category arrives like a conquistador to command and encompass it from outside. For the non-modern, the message is the message. And any other loftiness – from heaven above to down below – will always be legible right there within it. Indeed, only a modern would ever invent the word “content” in the first place.

Are mathematicians modernists? Perhaps they are, given the way in which math tends to return continuously to its own formal construction. We know that Plato, that great

devotee of math, was the first “modern” in this sense, as anachronistic as that may sound. But what about computer art? Is it modernist as well? Here the answer is not so clear, with modernist tendencies evident in certain micro-movements (late 1990s net art for instance) while less evident in others.

Where does the artist duo Jodi (or Jodi.org) stand in all of this? The answer seems clear enough. They are moderns through and through. There is no Jodi work that is not oriented toward the digital as its object and material. There is no Jodi work that is not on and about the material. They display in abundance that great modernist virtue of self-referentiality. The material of their work is quite simply the material itself.

Still, digitality in art today is, for the most part, not modernist. In contemporary art, digitality typically doesn’t signal medium specificity or a reflection on art’s conditions of possibility. Digitality today is usually understood in terms of the flexibility or inconstancy of the substrate (the so-called crisis of indexicality), or alternately in terms of network phenomena like circulation and dispersal (following the interesting work of David Joselit or Seth Price), or simply as a form of ambient environment, feeding and inflecting the kind of work being



xn--5cd.com

made.

Jodi are thus stubbornly out of step with the dominant rhythms of contemporary art. Less obsessed with the cultural or social effects of new media, Jodi orient themselves toward the specificities of hardware and software. The resulting aesthetic is, in this way, not entirely specified by the artists' subjective impulses. Instead, the texture of code and computation takes over, and computing itself – its strange logic, its grammar and structure, and often its shape and color – produces the aesthetic.

Jodi's *IDN* is chiefly concerned with the infrastructure of the Web. The work focuses on two technologies, the Internationalized Domain Name (IDN) implementation (which lends its name to Jodi's project) and Unicode, the universal character encoding standard. Jodi have made works about domain names in the past. "Wrong Browser" is a series of web browsers with names like "CO.JP" alluding to the top-level domains used for countries around the world; each browser renders a different series of abstract compositions, while frustrating the user's sense of interactivity. A different project, "LVY," is a group of three dubious domain names (LinchedIn.com, Vodacone.com, YouTuhb.com) that guide the user to an enigmatic, shivering

animation. Or, in an early work labeled simply *map* (map.jodi.org), the duo provided a low-tech mapping of the landscape of domains and sites that most interested them at the time, sites like re-move.org or irational.org.

As for Unicode, the character forms and glyphs of computer alphabets have long been exploited for their graphical qualities. ASCII art arranges the letters and punctuation of the ASCII character set into a crude palette of tones from light to dark, drawing pictures from the tones. And games like *Dwarf Fortress* use exotic glyphs to represent characters and objects. A system of “expanded punctuation” has also long been used to convey mood in text, both online and off (via the typewriter), using simple emoticons like ;) and more complex faces and pictures like (^;) or (^;)_(^;). After the gradual adoption of Unicode, applications and operating systems could render a vast array of graphical signs, both for all the world’s alphabets but also a variety of icons and emoji. Gamers quickly learned simple hacks for usernames and text chatting, adding a bit of flare with a name like “Κίττη” (instead of Kitten). Artists Jörg Piringer and Nick Montfort have both made work that plumbs the length and breadth of various encoding schemes, from Unicode and ASCII to the character-encoding

system used on the Commodore 64. The newly developed programming language Swift has also added Unicode support, not simply in what the language can compute but in the very text of the source code itself. In Swift the number 3.141592 may be assigned to a variable named π rather than pi , a small but significant distinction. Indeed it may be assigned to an emoji, a kanji character, or anything else available in Unicode. Swift's support for Unicode has facilitated new kinds of source-code chicanery beyond anything seen in the most notorious Perl poetry of yore.

Infrastructure changes slowly. Even as word processors and other applications began to support Unicode, many of the internet's internal technologies were late adopters. The IDN standard, only in use since 2010, was designed to allow Unicode characters within domain names, both top-level suffixes and server names. Thus, after the implementation of IDN, users may surf to pages with addresses ending in .рф instead of .ru (Russia) or .中国 instead of .cn (China). Likewise each server and hostname may be rendered in languages that don't use the Latin alphabet, such as Arabic or Greek. Still, the apparent universality of IDN is something of an illusion. Certain characters are prohibited outright to help avoid phishing attacks using similar looking glyphs. And, in fact, each Unicode domain is transcoded into an ASCII string behind the scenes. ASCII's much smaller character set,

04/06

consisting of the letters A-Z, the numerical digits 0-9, plus a few forms of simple punctuation, is considered to be simpler and more difficult to spoof. And, given that the Web was built on ASCII, it is easier to add Unicode support as a special form of ASCII encoding than change the Web's entire naming technology. For instance, a browser aimed at a Unicode address like “简.net” will first translate the address to the corresponding ASCII version, in this case “xn--417a.net,” and then fetch that address instead. In other words, even if a user sees Chinese or Russian characters on the screen, it's still ASCII underneath.

Jodi's *IDN* is a series of websites using single Unicode glyphs as domain names, all under the .net or .com top-level domains. Besides the primary glyph domains, additional websites are referenced via internal links. For example, ☺.com refreshes to ☻.com which refreshes back to ☺.com in a continuous loop. A few of the domains are as yet still empty, and a few others proffer short messages or other information. ¶.com and °.com both simply repeat the project's opening salutation, that "Apache is functioning normally."

The majority of pages produce graphical compositions that animate slowly in the browser via the HTTP “refresh” command. Some of them, like [glass.com](http://www.glass.com), animate solely in the address bar of the browser. Others, like [4chan.org](http://www.4chan.org) or [7.net](http://www.7.net), produce

Jodi, *IDN* (a screenshot of the work), 2015. Copyright: Jodi.org

large textual patterns that mimic or otherwise reference the shape of the glyph itself. (Here it's useful to compare HTML source to screen output, since the two often have different text justification or line wrapping; Jodi has explored this interesting inconsistency since their earliest work on the Web.)

05/06

Jodi, *IDN* (a screenshot of the work), 2015. Copyright: Jodi.org

At first glance, Jodi's *IDN* seems to resemble ASCII art or concrete poetry. I'm reminded of Carl Andre's typewriter poems where text appears on the page as geometric shapes. But despite this superficial similarity, *IDN* is in fact doing something a bit different. Jodi have woven their geometric shapes from out of a complicated hypertextual structure. Less concrete poetry, this is a kind of infrastructure poetry. The project IDN.com, for instance, requires a whole series of elaborate if not absurd host names. And to a certain extent, the work itself is nothing but a series of such names. When all the names are combined in a vertical stack, they create a patterned field of text. (One can only assume that Jodi had to write a series of scripts to automatically generate these many dozens of web pages, a tedious task if attempted by hand.)

Where does the work reside? Two places. First, Jodi seem interested in isolating certain parts of the screen, even certain parts of the browser. The browser's address bar, for instance, is treated here as a kind of miniature canvas for slow-motion animation. Like the structural films of Tony Conrad, Jodi create each animation frame by frame from discontinuous elements. A glyph becomes a single frame in a slow-motion film. When the glyphs combine in series, they give the illusion of movement. Like a form of primitive cinema, entire animations appear solely inside the address bar.

But the work resides in a second place as well. The projects in *IDN* are assembled not so much from discrete web pages as from the negative space existing between such pages. Jodi are interested in HTML, to be sure, but here they also display a penchant for the very standards and protocols of the web itself – how pages are assigned addresses and how servers transfer pages to clients. The “infrastructure” in this infrastructure poetry is thus the agglomeration of server software (Apache), addressing technologies (IDN and DNS), transfer protocols (HTTP), and finally, web browsers and the HTML they are designed to display.

Unicode is big. But the web is infinitely big. Like a fractal with its endless regress of complexity, a new web page can always be inserted between two existing ones, creating an infinitely large system. Unicode, by contrast, is a technology of universality, not infinity. The goal of Unicode is to facilitate all the world's writing systems, to arrange and classify them, but then to stop. Oh, how small is this vast Unicode when compared to infinity!

In the end, I suspect that Jodi are more interested in the web than Unicode, more interested in the structure of infinity than the classification of universality. The glyphs are adjunct here, a needed ingredient perhaps, but only necessary to facilitate animation and pattern. The true subject of the work is infrastructure – the cables and lines, the standards and protocols, all the industrial transfer technologies that reside in the space beyond the screen.

Net art was always something of an orphan, but a particularly interesting orphan. Shunned by the art world during its formative years, net art never quite fit into the master narratives of art, or technology for that matter. Jodi's infrastructural modernism, if we can call it that, is interesting because it suspends the distinction between art and technology without making one subservient to the other. Jodi are artists who insist on the importance of seemingly uninteresting technical minutiae, such as character-encoding schemes and other tedious

matters. And they are technologists who insist that the beauty of code comes not from function and elegance but from a different set of virtues – dysfunction and inelegance to be sure, but also confusion and excitement, violence and energy. The result is not so much a mechanization of art, nor that clumsy concept “the art of the machine,” but a much more simple and mundane reality: the computer as medium.

x

A shorter version of this article was first published by West Den Haag.

06/06

Alexander R. Galloway is a writer and computer programer working on issues in philosophy, technology, and theories of mediation. Professor of Media, Culture, and Communication at New York University, he is author of several books on digital media and critical theory, including *The Interface Effect* (Polity, 2012). His collaboration with Eugene Thacker and McKenzie Wark, *Excommunication: Three Inquiries in Media and Mediation*, has recently been published by the University of Chicago Press. With Jason E. Smith, Galloway co-translated the Tiqqun book *Introduction to Civil War* (Semiotext[e], 2010). For ten years he worked with RSG on *Carnivore*, *Kriegspiel*, and other software projects.