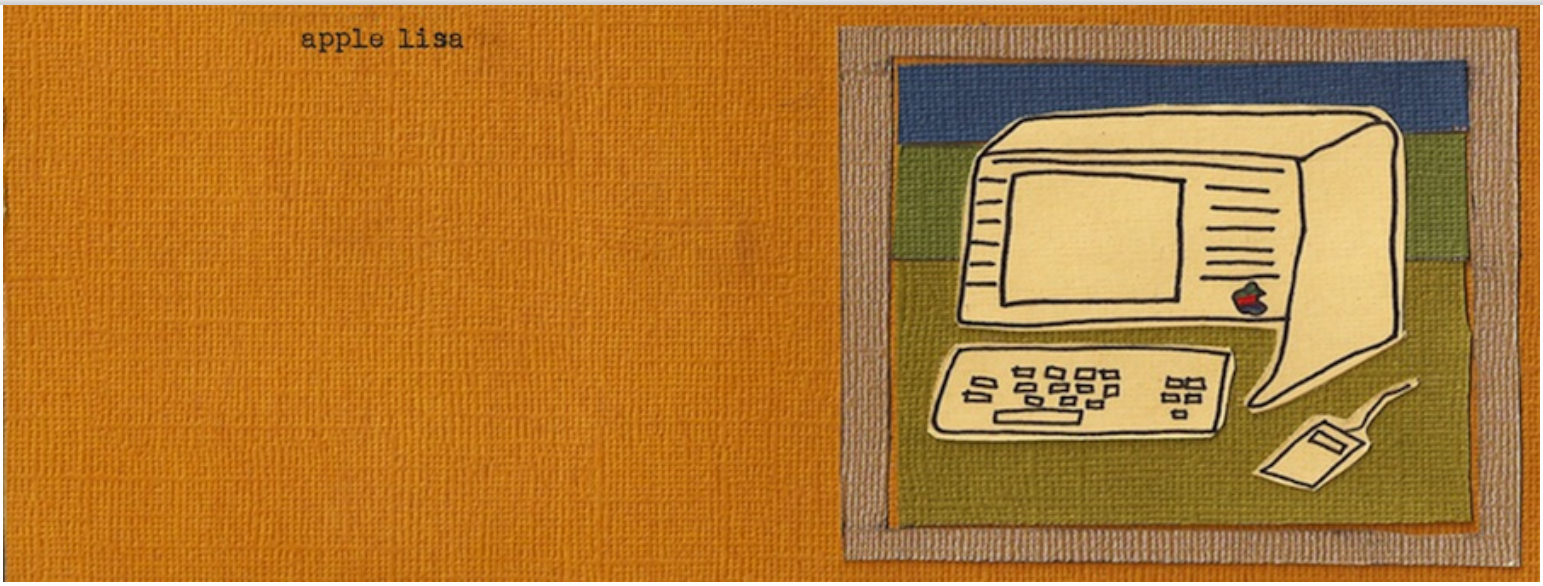
Category Archives: history of computing

Post navigation.



¶ loriemerson

Menu

Category Archives: history of computing

Older posts

Newer posts

Artist Residency at the

February 16, 2013

Media Archaeology Lab

Posted in digital, history of computing, media archaeology lab, media poetics ¶ Tagged conceptual art, keyboards, media archaeology, variantology ¶ 1 Comment

At the invitation of MAL curator Mél Hogan, on Monday February 11th conceptual artist Joel Swanson gave a talk about the art project he's working on in and for the lab on the history of computer keyboards and what symbolic or cultural meaning there might be in the presence or absence of certain keys. (In fact, Joel has already done some work with keyboards by way of his ultra-minimal, conceptual piece called "Spacebar" from 2012.) Here is the video of Joel's artist talk in the lab:

I'm very keen to see what Joel comes up with as I am fascinated with some of the keyboards in the lab, including an original keyboard for the Apple Macintosh from 1984 which famously has no

arrow keys so that users were forced to use the mouse.

As I've written elsewhere on this blog, the lab also has Commodore 64 computers which for example, came with both a 'Commodore' key that gave the user access to an alternate character set as well as four programmable function keys that, with the shift button, could each be programmed for two different functions.

By contrast, Apple II computers came with two programmable function keys and Apple III, IIc and IIe computers came with open-Apple and closed-Apple keys that provided the user with shortcuts to applications such as cut-and-paste

or copy.

I hope this is the first of many more artist residencies in the lab!

Share this:

Email Twitter

Loading…

February 9, 2013

It's Not Digital Humanities – it's Media Studies

Posted in criticism, digital, history of computing, media archaeology lab ¶ Tagged digital humanities, media archaeology, media studies ¶ 3 Comments

Thanks to the generosity of people at the Library of Congress such as Trevor Owens, I was fortunate to have the opportunity to interview media archaeologist Wolfgang Ernst on the LOC's blog *The Signal*. I especially wanted to talk with Ernst not only about his Media Archaeological Fundus (MAF), which bears a strong affiliation to my Media Archaeology Lab (MAL), but also about whether he sees a connection between his archival approach, the MAF, and preservation. Ernst responded by explaining that the emphasis in the MAF is more on training and "enforcing" media research through excavation and even a mathematical mode of thinking than on preservation. In terms of the latter, then, it's no surprise that Jussi Parikka points out on his blog that "Ernst is very reluctant to call this 'Digital

Humanities': it's media studies!" While DH is certainly deeply invested in doing and making as thinking, as (and as a response to) theory, I think that Ernst is still coming out of a Kittlerian project to "drive the spirit out of the humanities" and in this sense, no matter how inclusive DH becomes, perhaps media archaeology will steadfastly remain media studies, not DH.

You can find the entirety of the interview with Ernst here. As always, comments welcome.

Wolfgang Ernst's Media Archaeological Fundus

From the

February 2, 2013

Philosophy of the Open to the Ideology of the User-Friendly

Posted in bookbound, criticism, digital, e-literature, history of computing ¶ Tagged Apple, GUI, interface, Macintosh, MLA 2013, user-friendly ¶ 11 Comments

Below is an excerpt from chapter two, "From the Philosophy of the Open to the Ideology of the User-Friendly," from my book *Reading Writing Interfaces: From the Digital to the Bookbound* (University of Minnesota Press 2014). It is also the basis of the talk I gave at MLA in January 2013 and the full version of the talk I gave at Counterpath Press February 2013. As always, I welcome your comments!

*

> *"Knowledge is power: information is the fabric of knowledge; the controller of information wields power."*
> *– "Some Laws of Personal Computing,"* Byte *1979 (Lewis 191)*

> "If a system is to serve the creative spirit, it must be entirely comprehensible to a single individual…Any barrier that exists between the user

and some part of the system will eventually be a barrier to creative expression. Any part of the system that cannot be changed or that is not sufficiently general is a likely source of impediment."
–"Design Principles Behind Smalltalk," Byte *1981(Ingalls 286)*

My talk today is concerned with a decade in which we can track the shift from seeing a user-friendly computer as a tool that, through a graphical user interface (GUI), encourages understanding, tinkering, and creativity to seeing a user-friendly computer that uses a GUI to create an efficient work-station for productivity and task-management and the effect of this shift particularly on digital literary production. The turn from computer systems based on the command-line interface to those based on "direct manipulation" interfaces that are iconic or graphical was driven by rhetoric that insisted the GUI, particularly that pioneered by the Apple Macintosh design team, was not just different from the command-line interface but it was *naturally* better, easier, friendlier. The Macintosh was, as Jean-Louis Gassée (who headed up its development after Steve Jobs's departure in 1985) writes without any hint of irony, "the *third apple*," after the first apple in the Old Testament and the second apple that was Isaac Newton's, "the one that widens the paths of knowledge leading toward the future."

Despite studies released since 1985 that clearly demonstrate GUIs are not necessarily better than command-line interfaces in terms of how easy they are to learn and to use, Apple – particularly under Jobs' leadership – successfully created such a convincing aura of inevitable superiority around the Macintosh GUI that to this day the same "user-friendly" philosophy, paired with the no longer noticed closed architecture, fuels consumers' religious zeal for Apple products. I have been an avid consumer of Apple products since I owned my first Macintosh Powerbook in 1995; but what concerns me is that 'user-friendly' now takes the shape of keeping users steadfastly unaware and uninformed about how their computers, their reading/writing interfaces, work let alone how they shape and determine their access to knowledge and their ability to produce knowledge. As Wendy Chun points out, the user-friendly system is one in which users are, on the one hand, given the ability to "map, to zoom in and out, to manipulate, and to act" but the result is a "*seemingly* sovereign individual" who is mostly a devoted consumer of ready-made software and ready-made information whose framing and underlying mechanisms we are not privy to.

However, it's not necessarily the GUI *per se* that is responsible for the creation of Chun's "*seemingly* sovereign individual" but rather a particular philosophy of computing and design underlying a model of the GUI that has become the standard for nearly all interface design. The earliest example of a GUI-like interface whose philosophy is fundamentally different from that of the

Macintosh is Douglas Engelbart's NLS or "oN-Line System" which he began work on in 1962 and famously demonstrated in 1968. While his "interactive, multi-console computer-display system" with keyboard, screen, mouse, and something he called a chord handset is commonly cited as the originator of the GUI, Engelbart wasn't so much interested in creating a user-friendly machine as he was invested in "augmenting human intellect". As he first put it in 1962, this augmentation meant "increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems". The NLS was not about providing users with ready-made software and tools from which they choose or consume but rather it was about *bootstrapping*, or "the creation of tools for expert computer users" and providing the means for users to create better tools, or tools better suited for their own individual needs. We can see this emphasis on tool-building and customization that comes out of an augmented intellect in Engelbart's provision of "view-control" (which allows users to determine how much text they see on the screen as well as the form of that view) and "chains of views" (which allows the user to link related files) in his document editing program.

Underlining the fact that the history of computing is resolutely structured by stops, starts, and ruptures rather than a series of linear firsts, in the year before Engelbart gave his "mother of all demos," Seymour Papert and Wally Feurzeig began work on a learning-oriented

programming language they called 'Logo' that was explicitly for children but implicitly for learners of all ages. Throughout the 1970s Papert and his team at MIT conducted research with children in nearby schools as they tried to create a version of Logo that was defined by "modularity, extensibility, interactivity, and flexibility". At this time, the Apple II was the most popular home computer throughout the late 1970s until the mid-1980s and, given its open architecture, in 1977 Logo licensed a public version for Apple II computers as well as for the less popular Texas Instruments TI 99/4. In 1980, Papert published the decidedly influential *Mindstorms: Children, Computers, and Powerful Ideas* in which he makes claims about the power of computers that are startling for a contemporary readership steeped in an utterly different notion of what accessible or user-friendly computing might mean. Describing his vision of "computer-aided instruction" in which "the child programs the computer" rather than one in which the child adapts to the computer or even is taught by the computer, Papert asserts that they thereby "embark on an exploration about how they themselves think…Thinking about thinking turns the child into an epistemologist, an experience not even shared by most adults" (19). And two years later, in a February 1982 issue of *Byte* magazine, Logo is advertised as a general-purpose tool for thinking with a degree of intellectuality rare for any advertisement: "Logo has often been described as a language for children. It is so, but in the same sense that English is a language for children, a sense that does not preclude its being ALSO a language for

poets, scientists, and philosophers". Moreoever, for Papert thinking about thinking by way of programming happens largely when the user encounters bugs in the system and has to then identify where the bug is to then remove it: "One does not expect anything to work at the first try. One does not judge by standards like 'right – you get a good grade' and 'wrong – you get a bad grade.' Rather one asks the question: 'How can I fix it?' and to fix it one has first to understand what happened in its own terms." (101) Learning through doing, tinkering, experimentation, trial-and-error is, then, how one comes to have a genuine computer literacy.

In the year after Papert et al began work on Logo and the same year as Engelbart's NLS demo, Alan Kay also commenced work on the never-realized Dynabook, produced as an "interim Dynabook" in 1972 in the form of the GUI-based Xerox Alto which could also run the Smalltalk language. Kay thereby introduced the notion of "personal dynamic media" for "children of all ages" which "could have the power to handle virtually all of its owner's information-related needs". Kay, then, along with Engelbart and Papert, understood very clearly the need for computing to move from the specialized environment of the research lab and into people's homes by way of a philosophy of the user-friendly oriented toward the flexible production (rather than rigid consumption) of knowledge. It was a realization eventually shared by the broader computing community for, by 1976, Byte magazine was publishing editorials such as "Homebrewery vs the Software Priesthood" declaring that "the movement

towards personalized and individualized computing is an important threat to the aura of mystery that has surrounded the computer for its entire history" (90). And more:

> "The movement of computers into people's homes makes it important for us personal systems users to focus our efforts toward having computers do what we want them to do rather than what someone else has blessed for us...When computers move into peoples' homes, it would be most unfortunate if they were merely black boxes whose internal workings remained the exclusive province of the priests...Now it is not necessary that everybody be a programmer, but the potential should be there...(90).

from "Homebrewery vs the Software Priesthood," Byte magazine October 1976

It was precisely the potential for programming or simply novice as well as expert use via an open, extensible, and flexible architecture that Engelbart, Papert and Kay sought to build into their models of the personal computer to ensure that home computers did not become "merely black boxes whose internal workings remained the exclusive province of the priests." By contrast, as Kay later exhorted his readers in 1977, "imagine having your own self-contained knowledge manipulator in a portable package the size and shape of an ordinary notebook". Designed to have a keyboard, an NLS-inspired "chord" keyboard, mouse, display, and windows, the Dynabook would allow users to realize Engelbart's dream of a computing device that gave them the ability to create their own ways to

view and manipulate information. Rather than the over-determined post-Macintosh GUI computer which has been designed to pre-empt every user's possible need with the creation of an over-abundance of ready-made tools such that "those who wish to do something different will have to put in considerable effort," Kay wanted a machine that was "designed in a way that any owner could mold and channel its power to his own needs…a metamedium, whose content would be a wide range of already-existing and not-yet-invented media" (403). More, Kay understood from reading Marshall McLuhan, that the design of this new metamedium was no small matter for the very use of a medium changes an individual's, a culture's, thought patterns. Clearly, he wanted thought patterns to move toward a literacy that involved reading and writing in the new medium instead of the unthinking consumption of ready-made tools, for, crucially, "the ability to 'read' a medium means you can access materials and tools created by others. The ability to 'write' in a medium means you can generate materials and tools for others. You must have both to be literate".

While Kay envisioned the GUI-like interface of the Dynabook would play a crucial role in realizing this "metamedium," the Smalltalk software driving this interface was equally necessary. Its goal was "to provide computer support for the creative spirit in everyone" (286). Not surprisingly, Kay and his collaborators began working intensely with children after the creation of Smalltalk-71. Influenced by

developmental psychologist Jean Piaget as well as Kay's own observation of Papert and his colleagues' use of Logo in 1968, Smalltalk relied heavily on graphics and animation through one particular incarnation of the GUI: the Windows, Icons, Menus, and Pointers (or WIMP) interface. Kay writes that in the course of observing Papert using Logo in schools, he realized that these were children "doing real programming…":

> "…this encounter finally hit me with what the destiny of personal computing really was going to be. Not a personal dynamic vehicle, as in Engelbart's metaphor opposed to the IBM "railroads", but something much more profound: a personal dynamic medium. With a vehicle one could wait until high school and give "drivers ed", but if it was a medium, it had to extend into the world of childhood ("The Early History" 81).

As long as the emphasis in computing was on learning – especially through making and doing – the target demographic was going to be children and as long as children could use the system, then so too could any adult provided they understood the underlying structure, the how and the why, of the programming language.

As Kay astutely points out, "...we make not just to have, but to know. But the having can happen without most of the knowing taking place". And, as he goes on to point out, designing the Smalltalk user interface shifted the purpose of interface design from "access to functionality" to an "environment in which users learn by doing" (84). And so Smalltalk designers didn't so much completely reject the notion of readymade software so much as they sought to provide the user with a set of software building blocks from which the user could then combine and/or edit to create their own customized system. Or, as Trygve Reenskaug (a visiting Norwegian computer scientist with the Smalltalk group at Xerox PARC in the late 1970s) put it:

> "...the new user of a Smalltalk system is likely to begin by using its ready-made application systems for writing and illustrating documents, for designing aircraft wings, for doing homework, for searching through old court decisions, for composing music, or whatever. After a while, he may become curious as to how his system works. He should then be able to "open up" the application object on the screen to see its component parts and to find out how they work together

(166).

With an emphasis on learning and building through an open architecture, Adele Goldberg – co-developer of Smalltalk along with Alan Kay and author of most of the Smalltalk documentation – describes the Smalltalk programming environment in this special issue of *Byte* as one that set out to defy that of the conventional software development environment as illustrated in Figure 1 below:

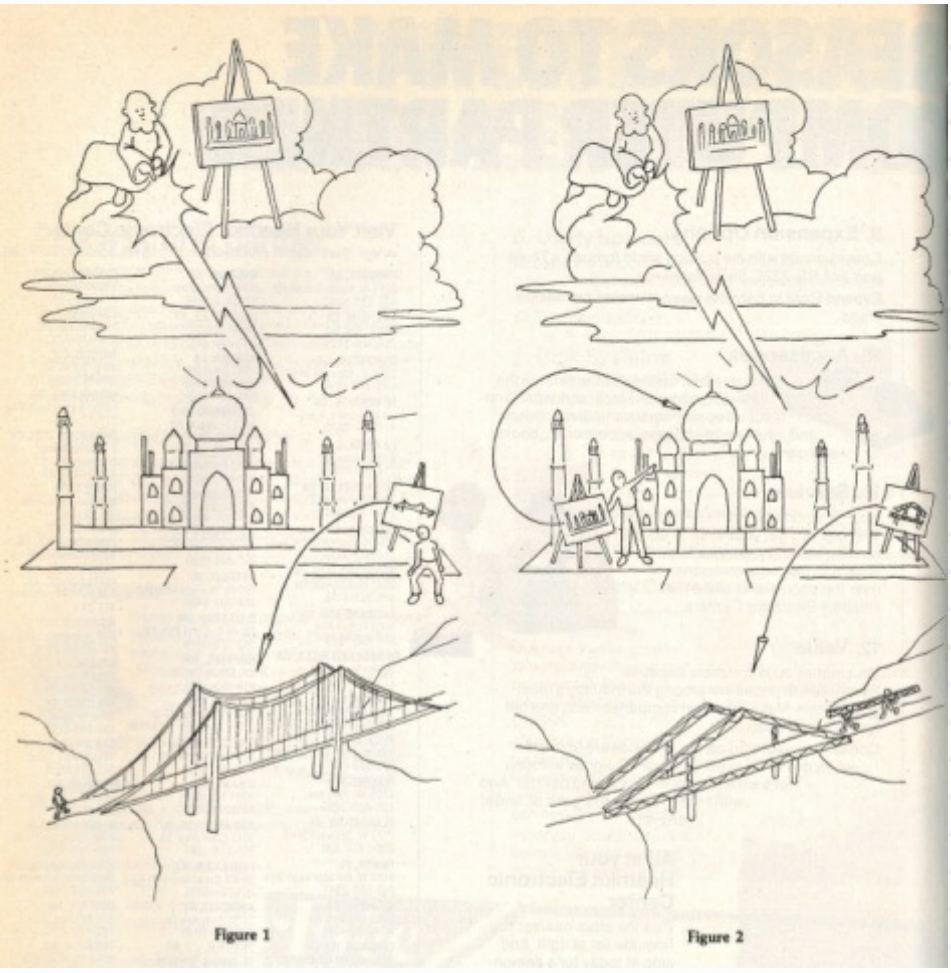


Image by Adele Goldberg contrasting the conventional philosophy of software driven by "wizards" in Figure 1 versus that provided by Smalltalk for the benefit of the programmer/user in Figure 2.

The Taj Mahal in Figure 1 "represents a complete programming environment, which includes the

tools for developing programs as well as the language in which the programs are written. The users must walk whatever bridge the programmer builds" (Goldberg 18). Figure 2, by contrast, represents a Taj Mahal in which the "software priest" is transformed into one who merely provides the initial shape of the environment which programmers can then modify by building "application kits" or "subsets of the system whose parts can be used by a nonprogrammer to build a customized version of the application" (18). The user or non-programmer, then, is an active builder in dialogue with the programmer instead of a passive consumer of a pre-determined (and perhaps even over-determined) environment.

At roughly the same time as Kay began work on Smalltalk in the early 1970s, he was also involved with the team of designers working on the NLS-inspired Xerox Alto which was developed in 1973 as, again, an "interim Dynabook" with a three-button mouse, a GUI which worked in conjunction with the desktop metaphor, and ran Smalltalk. While only several thousand non-commercially available Altos were manufactured, it was – as team members Chuck Thacker and Butler Lampson believe – probably the first computer explicitly called a "personal computer" because of its GUI and its network capabilities. By 1981, Xerox had designed and produced a commercially available version of the Alto, called the 8010 Star Information System, which was sold along with Smalltalk-based software. But as Jeff Johnson et al point out, the most important connection between Smalltalk and the Xerox Star

lay in the fact that Smalltalk could clearly illustrate the compelling appeal of a graphical display that the user accessed via mouse, overlapping windows, and icons (22).

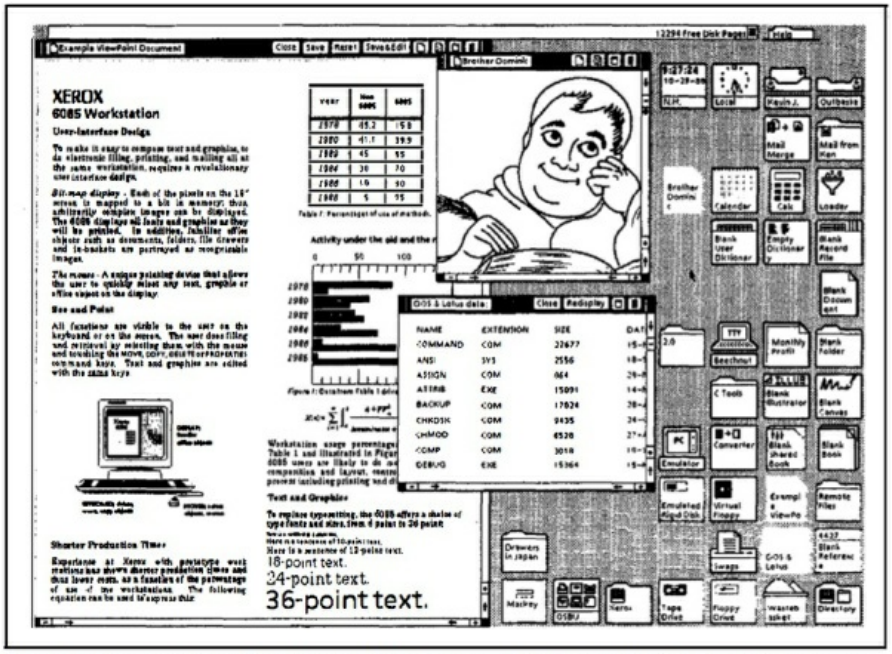


Figure 1. ViewPoint screen image. Star's bitmapped display, once unique in the marketplace, is now much more common. Such a display permits WYSIWYG editing, display of proportionally spaced fonts, integrated text and graphics, and graphical user interfaces.

Screenshot of Xerox Star from Jeff Johnson et al's "The Xerox Star: A Retrospective."

However, the significance of the Star is partly the indisputable impact it had on the GUI design of first the Apple Lisa and then the Macintosh; its significance is also in the way in which it was clearly labeled a work-station for "business professionals who handle information" rather than a metamedium or a tool for creating or even thinking about thinking. And in fact, the Star's interface – which was the first commercially available computer born out of work by Engelbart, Papert and Kay that attempted to satisfy both novice and expert users in providing an open, extensible, flexible environment and that also happened to be graphical – was conflicted at its core. While in some ways the Star

was philosophically very much in line with the open thinking of Engelbart, Papert, and Kay, in other ways its philosophy as much as its GUI directly paved the way to the closed architecture and consumption-based design of the Macintosh. Take for example the overall design principles of the Star which were aimed at making the system seem "familiar and friendly."

Easy	*Hard*
concrete	abstract
visible	invisible
copying	creating
choosing	filling in
recognizing	generating
editing	programming
interactive	batch

Star designers also avowed to avoid the characteristics they list on the right while adhering to a schema that exemplifies the characteristics listed on the left. While there's little doubt that ease-of-use was of central importance to Engelbart, Papert and Kay – often brought about through interactivity and making computer operations and commands visible – the avoidance of "creating," "generating," or "programming" couldn't be further from their vision of the future of computing. At the same time as the Star forecloses on creating, generating, and programming through its highly restrictive set of commands in the name of simplicity, it also wants to promote users' understanding of the system as a whole – although, again, we can see this particular

incarnation of the GUI represents the beginning of a shift toward only a superficial understanding of the system. Without a fully open, flexible, and extensible architecture, the home computer becomes less a tool for learning and creativity and more a tool for simply "handling information."

By contrast, as I'll now talk about, the Apple Macintosh was clearly designed for consumers, not creators. It was marketed as a democratizing machine when in fact it was democratizing only insofar as it marked a profound shift in personal computing away from the sort of inside-out know-how one needed to create on an Apple II to the kind of perfunctory know-how one needed to navigate the surface of the Macintosh – one that amounts to the kind of knowledge needed to click this or that button. The Macintosh was democratic only in the manner any kitchen appliance is democratic. That said, Apple's redefinition of the overall philosophy of personal computing exemplifies just one of many reversals that abound in this ten year period from the mid-1970s to the mid-1980s. In relation to the crucial change that took place in the mid-1980s from open, flexible, and extensible computing systems for creativity to ones that were closed, transparent, and task-oriented, the way in which the Apple Macintosh was framed from the time of its release in January 1984 represented a near complete purging of the philosophy promoted by Engelbart, Kay, and Papert. This purging of the recent past took place under the guise of Apple's version of the user-friendly that, among other things, pitted itself against the supposedly

“cryptic,” arcane,” “phosphorescent heap” that was the command-line interface as well as, it was implied, any earlier incarnation of the GUI.

However, it’s important to note that, while the Macintosh philosophy purged much of what had come before, it did in fact emerge from the momentum gathering in other parts of the computing industry which were particularly concerned to define standards for the computer interface. Up to this point, personal computers were remarkably different from each other. Commodore 64 computers, for example, came with both a ‘Commodore’ key that gave the user access to an alternate character set as well as four programmable function keys that, with the shift button, could each be programmed for two different functions. By contrast, Apple II computers came with two programmable function keys and Apple III, IIc and IIe computers came with open-Apple and closed-Apple keys that provided the user with shortcuts to applications such as cut-and-paste or copy (in the same way that the contemporary ‘command’ key functions).

No doubt in response to the difficulties this variability posed to expanding the customer base for personal computers, Byte magazine ran a two-part series in October and November 1982 dedicated to the issue of industry standards by way of an introduction to a proposed uniform interface called the “Human Applications Standard Computer Interface” (or HASCI). Asserting the importance of turning the computer into a “consumer product,” author

Chris Rutkowski declares that every computer ought to have a “standard, easy-to-use format” that “approaches one of transparency. The user is able to apply intellect directly to the task; the tool itself seems to disappear” (291, 299-300). Of course, a computer that is easy-to-use is entirely desirable; however, at this point ease-of-use is framed in terms of the disappearance of the tool being used in the name of ‘transparency ‘ – which now means usersfwhi can efficiently accomplish their tasks with the help of a glossy surface that shields them from the depths of the computer instead of the earlier notion of ‘transparency’ which referred to a usesr’s ability to open up the hood of the computer to understand directly its inner workings.

Thus, no doubt in a bid to finally produce a computer that realized these ideas and appealed to consumers who are “drivers, not repairmen,” Apple unveiled the Lisa in June 1983 for nearly $10,000 (that’s $23,000 in 2012 dollars) as a cheaper and more user-friendly version of the Xerox Alto/Star which sold for $16,000 in 1981 (which is about $40,000). At least partly inspired by Larry Tesler’s Xerox PARC 1979 demo of the Star to Steve Jobs, the Lisa used a one-button mouse, overlapping windows, pop-up menus, a clipboard, and a trashcan. As Tesler was adamant to point out in a 1985 article on the “Legacy of the Lisa,” it was “the first product to let you drag [icons] with the mouse, open them by double-clicking, and watch them zoom into overlapping windows” (17). The Lisa, then, moved that much closer to the realization of the dream of transparency with, for example, its mode of

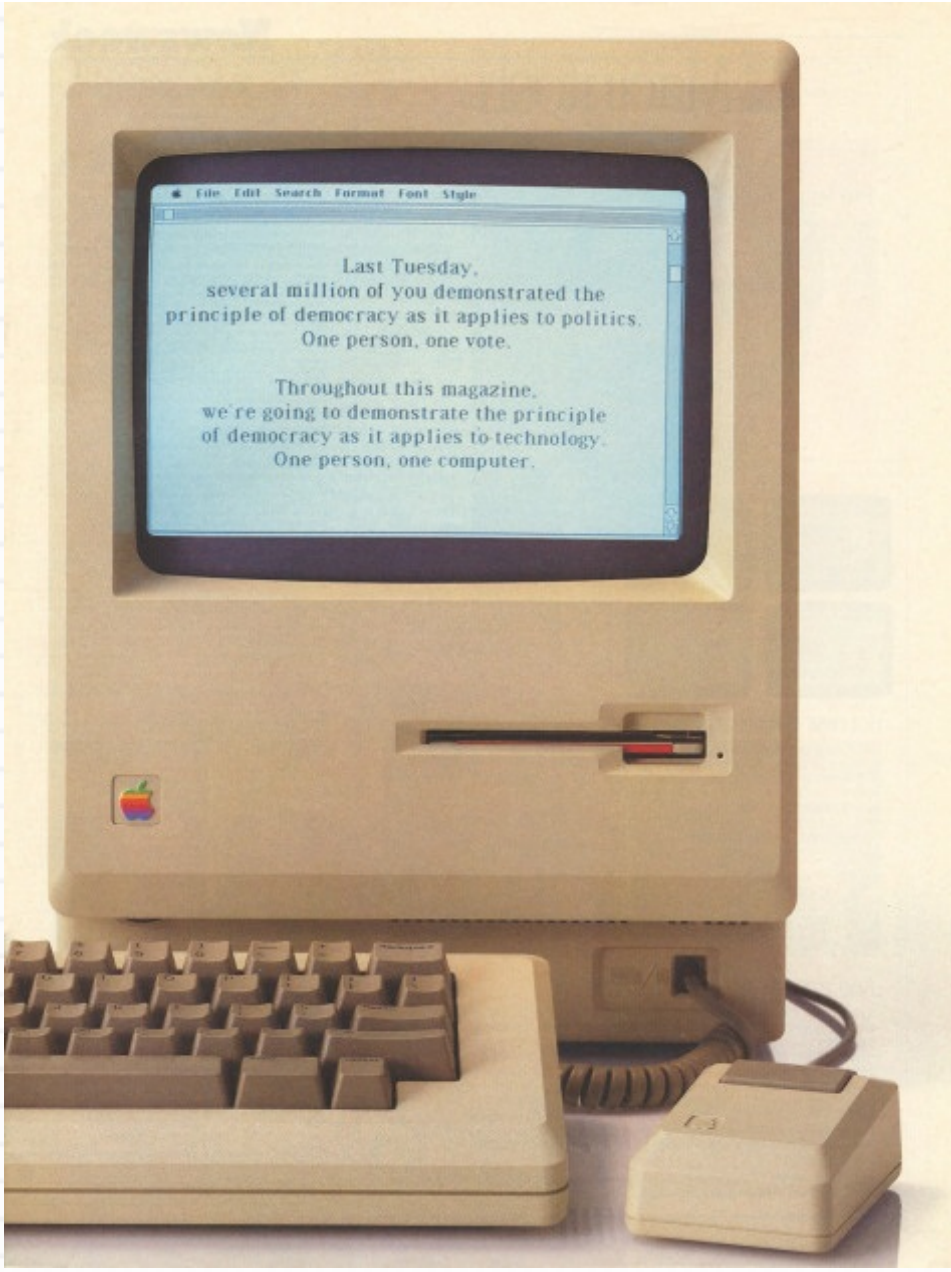
double-clicking that attempted to have users develop the quick, physical action of double-clicking that bypasses the intellect through physical habit; more, its staggering two 2048K worth of software and three expansion slots also firmly moved it in the direction of a readymade, closed consumer product and definitively away from the Apple II, which, when it was first released in 1977, came with 16K bytes of code and, again, eight expansion slots.

Expansion slots symbolize the direction that computing was to take from the moment the Lisa was released, followed by the release of the Macintosh in January 1984, to the present day. Jeff Raskin, who originally began the Macintosh project in 1979, and Steve Jobs both believed that hardware expandability was one of the primary obstacles in the way of personal computing having a broader consumer appeal. In short, expansion slots made standardization impossible (partly because software writers needed consistent underlying hardware to produce widely functioning products) whereas what Raskin and Jobs both sought was a system which was an "identical, easy-to-use, low-cost appliance computer." At this point, customization is no longer in the service of building, creating or learning – it is, instead, for using the computer as one would any home appliance and ideally this customization is only possible through software that the user drops into the computer via disk just as they would a piece of bread into a toaster. Predictably, then, the original plan for the Macintosh had it tightly sealed so that the user was only free to use the

peripherals on the outside of the machine. While team-member Burrell Smith managed to convince Jobs to allow him to add in slots for users to expand the machine's RAM, Macintosh owners were still "sternly informed that only authorized dealers should attempt to open the case. Those flouting this ban were threatened with a potentially lethal electric shock".

That Apple could successfully gloss over the aggressively closed architecture of the Macintosh while at the same time market it as a democratic computer "for the people" marks just one more remarkable reversal from this period in the history of computing. As is clear in the advertisement below that came out in Newsweek Magazine during the 1984 election cycle, the Macintosh computer was routinely touted as embodying the principle of democracy. While it was certainly more affordable than the Lisa (in that it sold for the substantially lower price of $2495), its closed architecture and lack of flexibility could still easily allow one to claim it represented a decidedly undemocratic turn in personal computing.

Thus, 1984 became the year that Apple's philosophy of the computer-as-appliance, encased in an aesthetically pleasing exterior, flowered into an ideology. We can partly see how their ideology of the user-friendly came to fruition through their marketing campaign which included a series of magazine ads such as the one below—



Advertisement for the Apple Macintosh from the November/December 1984 issue of Newsweek Magazine.

—along with one of the most well-known TV commercials of the late twentieth century.In the case of the latter, Apple takes full advantage of the powerful resonance still carried by George Orwell's dystopian, post-World War II novel *1984* by reassuring us in the final lines of the commercial that aired on 22 January 1984 that "On January 24th Apple Computer will introduce Macintosh. And you'll see why 1984 won't be like '1984.'"

Apple positions Macintosh, then, as a tool for and of democracy while also pitting the Apple philosophy against a (non-existent) 'other' (perhaps communist, perhaps IBM or 'Big Blue') who is attempting to oppress us with an ideology of bland sameness. Apple's ideology, then, "saves us" from a vague and fictional, but no less threatening, Orwellian, and nightmarish ideology. As lines of robot-like people, all dressed in identical grey, shapeless clothing march into the opening scene of the commercial, a narrator of this pre-Macintosh nightmare appears on a screen before them in something that appears to be a propaganda film. We hear, spoken fervently, "Today we celebrate the first glorious anniversary of the Information Purification Directives." And, as Apple's hammer-thrower then enters the scene, wearing bright red shorts and pursued by soldiers, the narrator

of the propaganda film continues:

> We have created for the first time in all history a garden of pure ideology, where each worker may bloom, secure from the pests of any contradictory true thoughts. Our Unification of Thoughts is more powerful a weapon than any fleet or army on earth. We are one people, with one will, one resolve, one cause. Our enemies shall talk themselves to death and we will bury them with their own confusion.

And just before the hammer is thrown at the film-screen, causing a bright explosion that stuns the grey-clad viewers, the narrator finally declares, “We shall prevail!” But who exactly is the hammer-thrower-as-underdog fighting against? Who shall prevail – Apple or Big Brother? Who is warring against whom in this scenario and why? In the end, all that matters is that, at this moment, just two days before the official release of the Macintosh, Apple has created a powerful narrative of its unquestionable, even natural superiority over other models of computing that continues well into the twenty-first century. It is an ideology that of course masks itself as such and that is born out of the creation of and then opposition to

a fictional, oppressive ideology we users/consumers need to be saved from. In this context, the fervor with which even Macintosh team-members believed in the rightness and goodness of their project is somewhat less surprising as they were quoted in Esquire earnestly declaring, “Very few of us were even thirty years old…We all felt as though we had missed the civil rights movement. We had missed Vietnam. What we had was the Macintosh”.

Even non-fiction accounts of the Macintosh by non-Apple employees could not help but endorse it in as breathless terms as those used by the Macintosh team-members themselves. Steven Levy’s Insanely Great, from 1994, is a document as remarkable for its wholesale endorsement of this new model of personal computing as any of the Macintosh advertisements and guide-books. Recalling his experience seeing a demonstration of a Macintosh in 1983, he writes:

> “Until that moment, when one said a computer screen “lit up,” some literary license was required…But we were so accustomed to it that we hardly even thought to conceive otherwise. We simply hadn’t seen the light. I saw it that day…By the end of the demonstration, I began to

> understand that these were things a computer should do. There was a better way (4).

The Macintosh was not simply one of several alternatives – it represented the unquestionably right way for computing. And even at the time of his writing that book, in 1993, he still declares that each time he turns on his Macintosh, he is reminded "of the first light I saw in Cupertino, 1983. It is exhilarating, like the first glimpse of green grass when entering a baseball stadium. I have essentially accessed another world, the place where my information lives. It is a world that one enters without thinking of it…an ephemeral territory perched on the lip of math and firmament" (5). But it is precisely the legacy of the unthinking, invisible nature of the so-called "user-friendly" Macintosh environment that has foreclosed on using computers for creativity and learning and that continues in contemporary multi-touch, gestural, and ubiquitous computing devices such as the iPad and the iPhone whose interfaces are touted as utterly invisible (and so their inner workings are de facto inaccessible).

References

"'1984' Apple Macintosh Commercial." *Youtube*. 27 Aug. 2008. Web. 21 June 2012.

Apple Computer Inc. *Apple Human Interface Guidelines: The Apple Desktop Interface*. Reading, MA: Addison-Wesley, 1987.

Bardini, Thierry. *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing*. Stanford, CA: Stanford UP, 2000.

Chen, Jung-Wei and Jiajie Zhang. "Comparing Text-based and Graphic User Interfaces for Novice and Expert Users." AMIA Annual Symposium Proceedings Archive. 2007. Web. 14 February 2012.

Chun, Wendy. *Programmed Visions: Software and Memory*. Boston, MA: MIT Press, 2011.

Engelbart, Douglas. "Augmenting Human Intellect: A Conceptual Framework." in *The New Media Reader*. Eds. Noah Wardrip-Fruin and Nick Montfort. Boston, MA: MIT UP, 2003. 95-108.

—. "Workstation History and the Augmented Knowledge Workshop." Doug Engelbart Institute. 2008. Web. 3 April 2011.

—, and William English. "A Research Center for Augmenting Human Intellect." in *The New Media Reader*. Eds. Noah Wardrip-Fruin and Nick Montfort. Boston, MA: MIT UP, 2003. 233-246.

Erickson, Thomas D. "Interface and the Evolution of Pidgins: Creative Design for the Analytically Inclined." In *The Art of Human-Computer Interface Design*. Ed. Brenda Laurel. Reading, MA: Addison-Wesley Publishing Company, Inc., 1990. 11-16

Gassée, Jean-Louis. *The Third Apple: Personal Computers & the Cultural Revolution*. San Diego, New York, London: Harcourt Brace Jovanovich

Publishers, 1985.

Goldberg, Adele. “Introducing the Smalltalk-80 System.” *Byte* 6:8 (August 1981): 14-26.

Hertzfeld, Andy and Steve Capps et al. *Revolution in the Valley*. Sebastopol, CA: O’Reilly, 2005.

Ingalls, Daniel. “Design Principles Behind Smalltalk.” *Byte* 6:8 (August 1981): 286-298.

Johnson, Jeff and Theresa Roberts et al. “The Xerox Star: A Retrospective.” *Computer* 22:9 (September 1989): 11-29.

Johnson, Steven. *Interface Culture: How New Technology Transforms the Way We Create and Communicate*. New York: Basic Books, 1997.

Kay, Alan. “The Early History of Smalltalk.” Smalltalk dot org. Web. 5 April 2012.

—. “User Interface: A Personal View.” in *The Art of Human-Computer Interface Design*. Ed. Brenda Laurel. Reading, MA: Addison-Wesley Publishing Company, Inc., 1990. 191-207.

—, and Adele Goldberg. “Personal Dynamic Media.” in *The New Media Reader*. Eds. Noah Wardrip-Fruin and Nick Montfort. Boston, MA: MIT UP, 2003. 393-409.

Levy, Steven. *Hackers: Heroes of the Computer Revolution*. 25th Anniversary Edition. New York: O’Reilly Media, 2010.

—. *Insanely Great: The Life and Times of Macintosh, the Computer that Changed Everything.* New York: Viking, 1994.

Lewis, T.G. "Some Laws of Personal Computing." *Byte* 4:10 (October 1979): 186-191.

Linden, Ted, Eric Harslem, Xerox Corporation. *Office Systems Technology: A Look Into the World of the Xerox 8000 Series Products: Workstations, Services, Ethernet, and Software Development.* Palo Alto, CA: Office Systems Division, 1982.

"LOGO." Advertisement. *Byte* 7:2 (February 1982): 255.

Morgan, Chris and Gregg Williams, Phil Lemmons. "An Interview with Wayne Rosing, Bruce Daniels, and Larry Tesler: A Behind-the-scenes Look at the Development of Apple's Lisa." Reprinted from *Byte* magazine 8:2 (February 1983): 90-114. Web. 14 April 2012.

Nelson, Theodor. "Computer Lib / Dream Machines." *The New Media Reader.* Eds. Noah Wardrip-Fruin and Nick Montfort. Cambridge, MA: MIT Press, 2003. 303-338.

Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books, 1980.

Reenskaug, Trygve. "User-Oriented Descriptions of Smalltalk Systems." *Byte* 6:8 (August 1981): 148-166.

Reimer, Jeremy. "Total share: 30 years of

personal computer market share figures." Ars Technica. 2006. Web. 4 December 2011.

Rutkowski, Chris. "An Introduction to the Human Applications Standard Computer Interface: Part 1: Theory and Principles." *Byte* 7:10 (October 1982): 291-310.

—. "An Introduction to the Human Applications Standard Computer Interface: Part 2: Implementing the HASCI Concept. " *Byte* 7:11 (November 1982): 379-390.

Smith, David Canfield and Charles Irby et al. "Designing the Star User Interface." *Byte* 7:4 (April 1982): 242-282.

Tesler, Larry. "The Legacy of the Lisa." *Macworld* magazine (September 1985): 17-22.

Wardrip-Fruin, Noah. "Introduction." "A Research Center for Augmenting Human Intellect." By Douglas Engelbart. in *The New Media Reader*. Eds. Noah Wardrip-Fruin and Nick Montfort. Boston, MA: MIT UP, 2003. 231-232.

"What is Logo?" The Logo Foundation. 2011. Web. 5 April 2012.

Whiteside, John and Sandra Jones, Paul S. Levy, Dennis Wixon. "User Performance with Command, Menu, and Iconic Interfaces." CHI 1985 Proceedings. April 1985. 185-191.

Wilber, Mike and David Fylstra. "Homebrewery vs the Software Priesthood." *Byte* 14 (October 1976):

90-94.

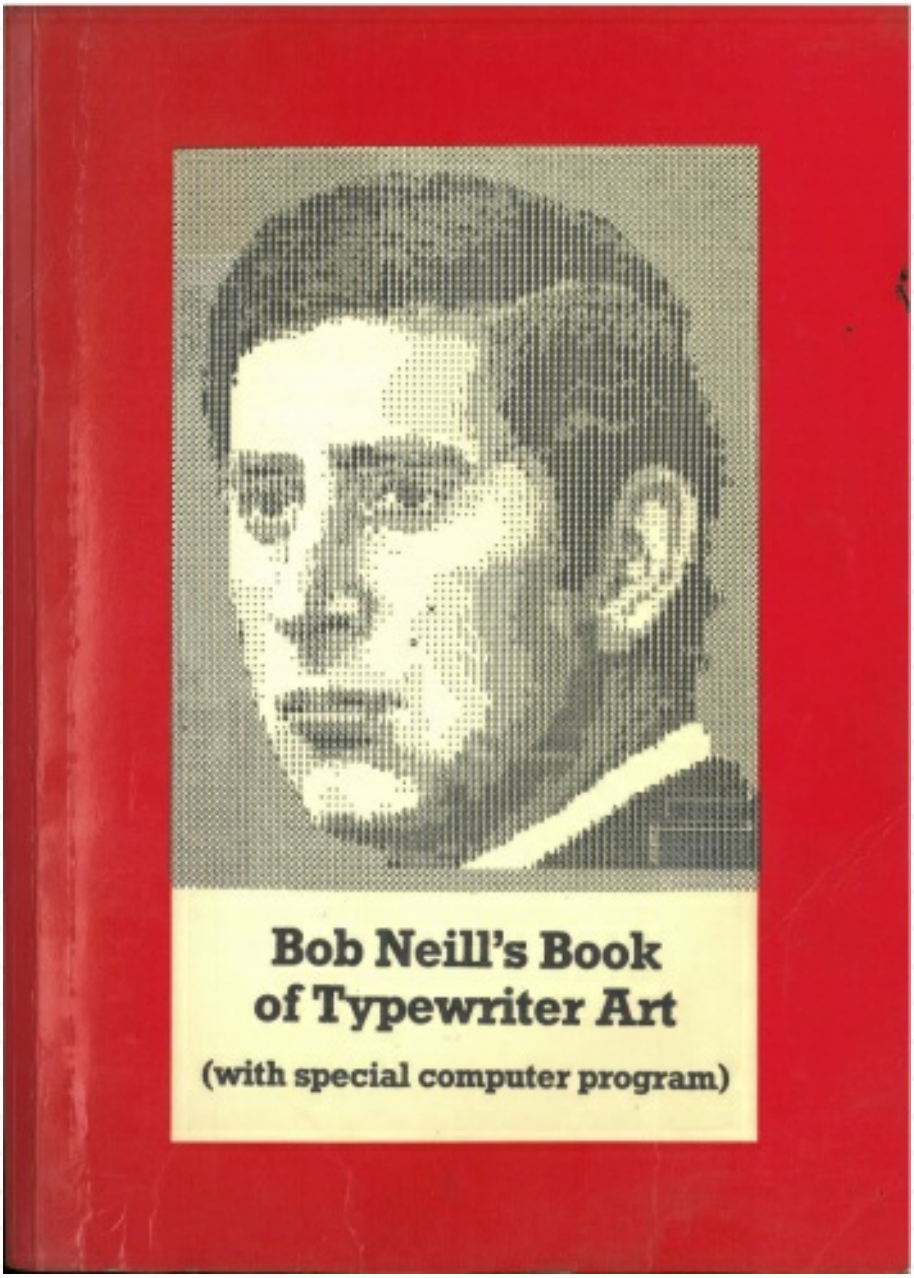
Williams, Gregg. “The Lisa Computer System: Apple Designs a New Kind of Machine.” Product Description. *Byte* 8:2 (February 1983): 33-50.

Wozniak, Steve. “The Apple-II.” System Description. *Byte* 2:5 (May 1977): 34-43.

January 18, 2013

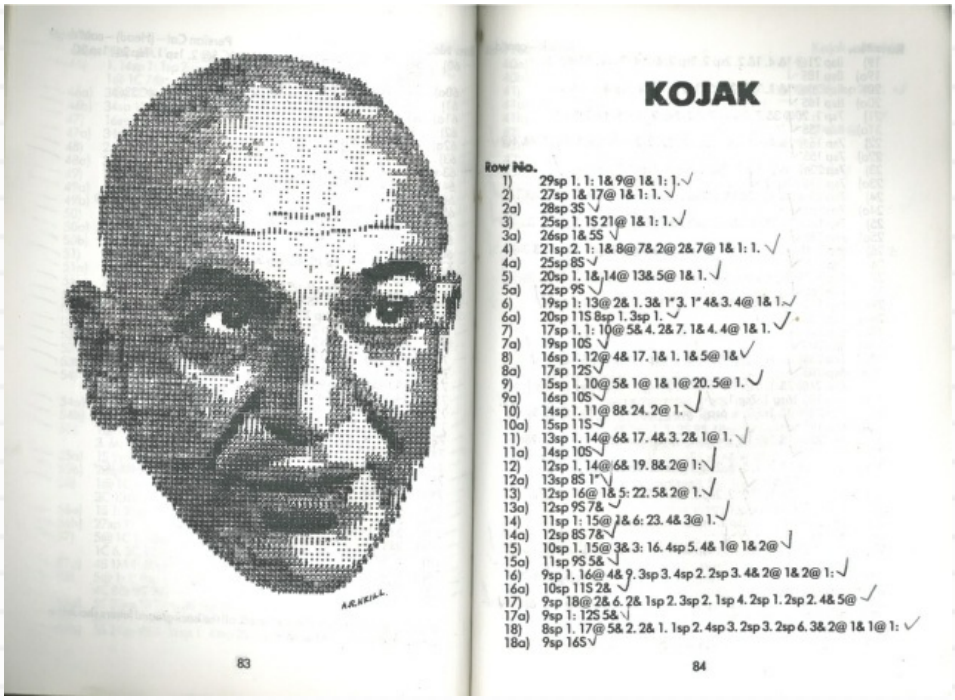
D.I.Y. Typewriter Art

Posted in bookbound, criticism, history of computing, media poetics ¶ Tagged D.I.Y., typewriter ¶ 18 Comments



Download the pdf here.

This lovely oddity arrived in the mail yesterday – *Bob Neill's Book of Typewriter Art (with special computer program)* from 1982. It's so difficult to capture its lovely oddness is just a few sentences or images so I decided to scan the entirety of the book and make it available here (pdf). Inside you'll find line-by-line instructions for creating charming portraits of everything from the British royal family to siamese cats and even Kojak.



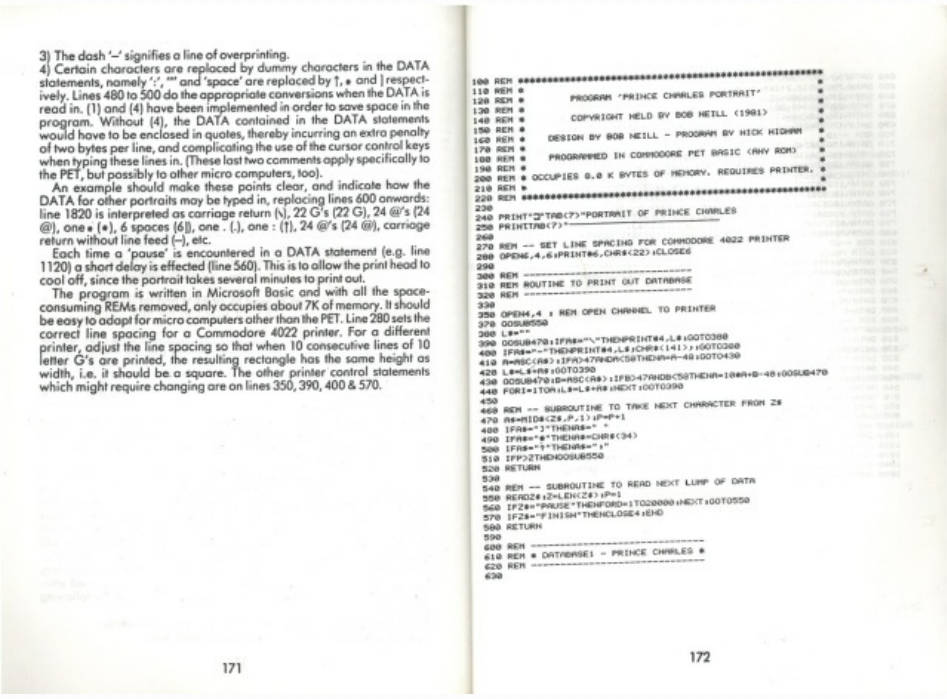
KOJAK

Row No.
1) 29sp 1. 1: 1& 9@ 1& 1: 1.
2) 27sp 1& 17@ 1& 1: 1.
2a) 28sp 3S
3) 25sp 1. 1S 21@ 1& 1: 1.
3a) 26sp 1& 5S
4) 21sp 2. 1: 1& 8@ 7& 2@ 2& 7@ 1& 1: 1.
4a) 25sp 8S
5) 20sp 1. 1&,14@ 13& 5@ 1& 1.
5a) 22sp 9S
6) 19sp 1: 13@ 2& 1. 3& 1*3. 1* 4& 3. 4@ 1& 1.
6a) 20sp 11S 8sp 1. 3sp 1.
7) 17sp 1. 1: 10@ 5& 4. 2& 7. 1& 4. 4@ 1& 1.
7a) 19sp 10S
8) 16sp 1. 12@ 4& 17. 1& 1. 1& 5@ 1&
8a) 17sp 12S
9) 15sp 1. 10@ 5& 1@ 1& 1@ 20. 5@ 1.
9a) 16sp 10S
10) 14sp 1. 11@ 8& 24. 2@ 1.
10a) 15sp 11S
11) 13sp 1. 14@ 6& 17. 4& 3. 2& 1@ 1.
11a) 14sp 10S
12) 12sp 1. 14@ 6& 19. 8& 2@ 1:
12a) 13sp 8S 1*
13) 12sp 16@ 1& 5: 22. 5& 2@ 1.
13a) 12sp 9S 7&
14) 11sp 1: 15@ 1& 6: 23. 4& 3@ 1.
14a) 12sp 8S 7&
15) 10sp 1. 15@ 3& 3: 16. 4sp 5. 4& 1@ 1& 2@
15a) 11sp 9S 5&
16) 9sp 1. 16@ 4& 9. 3sp 3. 4sp 2. 2sp 3. 4& 2@ 1& 2@ 1:
16a) 10sp 11S 2&
17) 9sp 18@ 2& 6. 2& 1sp 2. 3sp 2. 1sp 4. 2sp 1. 2sp 2. 4& 5@
17a) 9sp 1: 12S 5&
18) 8sp 1. 17@ 5& 2. 2& 1. 1sp 2. 4sp 3. 2sp 3. 2sp 6. 3& 2@ 1& 1@ 1:
18a) 9sp 16S

83 84

I've long been interested in the way writers in the 1960s and 1970s were – once the typewriter had thoroughly become commonplace – finding ways to play with the limits and possibilities of this machine as a writing medium. I've also thought that we can look back on typestracts such as Steve McCaffery's *Carnival* and see it as informed by a D.I.Y. and hacking sensibility. While this book of typewriter art is clearly invested in representationality and not particularly experimental, its content is entirely a D.I.Y. guide to creating typewriter art and is very much like computer magazines from the early 1980s such as *Byte* that would include BASIC programs. Here, instead of computer code, we're given typewritten letters *as code*. And in fact, the book includes an appendix with a Microsoft BASIC program for creating a "Prince Charles Portrait", programmed for the Commodore PET. And since the second appendix is a chart showing "sizes of paper required for each picture on different kinds of typewriter," I can't help thinking this book is a unique artifact in that it's entirely framed by the appearance of the

personal computer – a book on a soon-to-be-outdated technology framed by its impending replacement by a new technology.



3) The dash '–' signifies a line of overprinting.
4) Certain characters are replaced by dummy characters in the DATA statements, namely ':', '"' and 'space' are replaced by ↑, * and] respectively. Lines 480 to 500 do the appropriate conversions when the DATA is read in. (1) and (4) have been implemented in order to save space in the program. Without (4), the DATA contained in the DATA statements would have to be enclosed in quotes, thereby incurring an extra penalty of two bytes per line, and complicating the use of the cursor control keys when typing these lines in. (These last two comments apply specifically to the PET, but possibly to other micro computers, too).

An example should make these points clear, and indicate how the DATA for other portraits may be typed in, replacing lines 600 onwards: line 1820 is interpreted as carriage return (\), 22 G's (22 G), 24 @'s (24 @), one * (*), 6 spaces (6]), one . (.), one : (↑), 24 @'s (24 @), carriage return without line feed (–), etc.

Each time a 'pause' is encountered in a DATA statement (e.g. line 1120) a short delay is effected (line 560). This is to allow the print head to cool off, since the portrait takes several minutes to print out.

The program is written in Microsoft Basic and with all the space-consuming REMs removed, only occupies about 7K of memory. It should be easy to adapt for micro computers other than the PET. Line 280 sets the correct line spacing for a Commodore 4022 printer. For a different printer, adjust the line spacing so that when 10 consecutive lines of 10 letter G's are printed, the resulting rectangle has the same height as width, i.e. it should be a square. The other printer control statements which might require changing are on lines 350, 390, 400 & 570.

171

```
100 REM ***********************************************
110 REM *                                             *
120 REM *        PROGRAM 'PRINCE CHARLES PORTRAIT'    *
130 REM *                                             *
140 REM *        COPYRIGHT HELD BY BOB NEILL (1981)   *
150 REM *                                             *
160 REM *    DESIGN BY BOB NEILL - PROGRAM BY NICK HIGHAM *
170 REM *                                             *
180 REM *    PROGRAMMED IN COMMODORE PET BASIC (ANY ROM) *
190 REM *                                             *
200 REM * OCCUPIES 8.0 K BYTES OF MEMORY. REQUIRES PRINTER. *
210 REM *                                             *
220 REM ***********************************************
230
240 PRINT"□"TAB(7)"PORTRAIT OF PRINCE CHARLES
250 PRINTTAB(7)"----------------------------
260
270 REM -- SET LINE SPACING FOR COMMODORE 4022 PRINTER
280 OPEN6,4,6:PRINT#6,CHR$(22);CLOSE6
290
300 REM ------------------------------
310 REM ROUTINE TO PRINT OUT DATABASE
320 REM ------------------------------
330
350 OPEN4,4 : REM OPEN CHANNEL TO PRINTER
370 GOSUB550
380 L$=""
390 GOSUB470:IFA$="\"THENPRINT#4,L$;GOTO380
400 IFA$="-"THENPRINT#4,L$;CHR$(141);:GOTO380
410 A=ASC(A$):IFA>47ANDA<58THENA=A-48:GOTO430
420 L$=L$+A$:GOTO390
430 GOSUB470:B=ASC(A$):IFB>47ANDB<58THENA=10*A+B-48:GOSUB470
440 FORI=1TOA:L$=L$+A$:NEXT:GOTO390
450
460 REM -- SUBROUTINE TO TAKE NEXT CHARACTER FROM Z$
470 A$=MID$(Z$,P,1):P=P+1
480 IFA$="]"THENA$=" "
490 IFA$="*"THENA$=CHR$(34)
500 IFA$="↑"THENA$=":"
510 IFP>ZTHENGOSUB550
520 RETURN
530
540 REM -- SUBROUTINE TO READ NEXT LUMP OF DATA
550 READZ$:Z=LEN(Z$):P=1
560 IFZ$="PAUSE"THENFORD=1TO20000:NEXT:GOTO550
570 IFZ$="FINISH"THENCLOSE4:END
580 RETURN
590
600 REM ------------------------------
610 REM * DATABASE1 - PRINCE CHARLES *
620 REM ------------------------------
630
```

172

Share this:

Email Twitter

Loading…

December 17, 2012

In-Progress Catalog of the MAL's Holdings

Posted in history of computing, media archaeology lab ¶ Tagged archives, history of computing, interface, media studies ¶ Leave a comment

With heartfelt thanks to my research assistant Caitlin

Purdy and to Kyle Bickoff, a graduate student here at CU Boulder, the Media Archaeology Lab now has a nearly complete catalog of all its holdings. The catalog is clearly still a work-in-progress and, other than the just the organizational challenges in the document itself, the next step for the MAL is a web-based, searchable catalog. Still, hopefully the list below at least gives researchers a sense of what they can find in the lab. We also haven't quite worked out a system for documenting material from particular donors and integrating this information into the main body of the catalog – at the moment, items from our most recent donors (Timothy Sweeney and Robert Craig) are listed separately toward the end of the catalog.

*

DOWNLOAD A PDF OF THE MAL CATALOG HERE.

Print Material

8-Bit Digital Sound Studio: User's Guide. N.p.: Great Valley Products, Inc., 1992. Print.

Abernethy, Ken, T. Ray Nanney, and Hayden Porter. *Exploring Macintosh: Concepts in Visually Oriented Computing*. New York: John Wiley & Sons, Inc., 1989. Print.

ALLC Bulletin 13.3 (1985). Print.

ALLC Bulletin 9.2 (1981). Print.

ALLC Bulletin 9.1 (1981). Print.

ALLC Bulletin 6.2 (1978). Print.

ALLC Bulletin 6.3 (1978). Print.

ALLC Bulletin 4.2 (1976). Print.

ALLC Bulletin 4.3 (1976). Print.

ALLC Bulletin 8.1 (1980). Print.

ALLC Bulletin 8.3 (1981). Print.

ALLC Bulletin 7.1 (1979). Print.

ALLC Bulletin 7.2 (1979). Print.

ALLC Journal 1.1 (1980). Print.

ALLC Journal 2.1 (1981). Print.

Apple II: DOS User's Manual. Cupertino: Apple Computers, Inc., 1982. Print.

Apple II: Quick File II. Cupertino: Apple Computer, Inc., 1982. Print.

Apple II Reference Manual. Cupertino: Apple Computer Inc, 1981. Print.

Apple II Utilities Guide. Cupertino: Apple Computer, Inc., 1981. Print.

Applesoft BASIC Programmer's Reference Manual. Cupertino: Apple Computer, Inc., 1982. Print.

Berkowitz, Rob. *Inside the Macintosh Communications Toolbox*. Ed. Scott Smith and Becky Reece. Cupertino: Apple Computer, Inc., 1991.

Print.

De Jong, Marvin L. *Apple II Assembly Language*. Indianapolis: Howard W. Sams & Co, Inc, 1982. Print.

The Einstein MemoryTrainer User Guide. Los Angeles: The Einstein Corporation, 1983. Print.

Englebardt, Stanley L. *The Worlds of Science: Cybernetics*. New York: Pyramid, 1962. Print.

Finkel, LeRoy, and Jerald R. Brown. *Apple Basic: Data File Programming*. N.p.: John Wiley & Sons, Inc., 1982. Print. Self Teaching Guide.

Frenzel, Louis E., Jr. *Crash Course in Microcomputers*. Indianapolis: Howard W. Sams & Co Inc, 1980. Print.

Gateley, Wilson Y., and Gary G. Bitter. *Basic for Beginners*. N.p.: McGraw Book Company, 1970. Print.

Grammer, Virginia Carter, and E. Paul. Goldenberg. *The Terrapin Logo Language for the AppleII*. Ed. Mark Eckenwiler and Peter Von Mertens. Cambridge: Terrapin, Inc., 1982. Print.

Inside Macintosh. Vol. VI. Cupertino: Apple Computer, Inc., 1991. Print.

Inside Macintosh. Vol. V. Cupertino: Apple Computer, Inc., 1986. Print.

Inside Macintosh. Vol. IV. Cupertino: Apple

Computer, Inc., 1985. Print.

Inside Macintosh. Vol. III. Cupertino: Apple Computer, Inc., 1985. Print.

Inside Macintosh. Vol. II. Cupertino: Apple Computer, Inc., 1985. Print.

Inside Macintosh. Vol. I. Cupertino: Apple Computer, Inc., 1985. Print.

Introduction, Complier, Editor. Cary: SAS Institute Inc., 1993. Print. Vol. 1 of *SAS/C Development System User's Guide*.

Jenngs, Edward M. *Science and Literature*. Garden City: Anchor, 1970. Print.

Literary & Linguistic Computing 4.2 (1989). Print.

Literary & Linguistic Computing 5.1 (1990). Print.

Literary & Linguistic Computing 2.3 (1987). Print.

Literary & Linguistic Computing 3.3 (1988). Print.

Literary & Linguistic Computing 3.2 (1988). Print.

Literary & Linguistic Computing 4.4 (1989). Print.

Literary & Linguistic Computing 4.1 (1989). Print.

Luebbert, William F. *What's Where in the Apple: A Complete Guide to the Apple Computer*. Amherst: Micro Ink, 1982. Print.

Luedtke, Peter, and Rainer Luedtke. *Your First Business Computer*. Bedford: Digital Equipment Corporation, 1983. Print. The Desktop Computer Series.

Macintosh Manual. Cupertino: Apple Computer, Inc., 1984. Print.

Micromodem Smartcom I: Owner's Manual. Norcross: Hayes Microcomputer Products, 1983. Print.

Parikka, Jussi. *What Is Media Archaeology?* Cambridge: Polity, 2012. Print.

PC World 1.2 (1983). Print.

PC World 1.1 (1983). Print.

PC World 1.4 (1983). Print.

PC World 1.3 (1983). Print.

Perspectives in Computing 2.1 (1982). Print.

Perspectives in Computing 1.4 (1981). Print.

Perspectives in Computing 1.2 (1981). Print.

Perspectives in Computing 1.1 (1981). Print.

Ratliff, Wayne. *dBASE II: Assembly Language Relational Database Management System*. Culver City: Ratliff Software Production, Inc., 1982. Print.

Schneider, Ben Ross, Jr. *Travels in Computerland*. N.p.: Addison-Wesley, 1974. Print.

Smith, George W. *Computers and Human Language*. London: Oxford University, 1991. Print.

Smith, Jon M. *Scientific Analysis on the Pocket Calculator*. N.p.: John Wiley & Sons, 1975. Print.

Snell, Barbara M. *Translating and the Computer*. N.p.: North-Holland, 1979. Print.

Sobel, Robert. *IBM: Colossus in Transition*. London: Sidgwick & Jackson, 1981. Print.

Texas Instruments TI-99/4A Computer: Beginner's BASIC. N.p.: Texas Instruments, 1979. Print.

Texas Instruments TI-99/4A Computer: User's Reference Guide. Texas Instruments Incorporated ed. N.p.: Texas Instruments, 1979. Print.

Texas Instruments TI-99/4 Home Computer: TI Extended BASIC. Dallas: Texas Instruments, 1981. Print.

Tindall, Peggy Cagle, and Michel Boillot. *Transparency Masters to Accompany Developing Computer Skills Using Appleworks*. St. Paul: West Publishing Company, 1991. Print.

Tucker, Allen B., Jr. *Text Processing: Algorithms, Languages, and Applications*. New York: Academic, 1979. Print.

Turkle, Sherry. *The Second Self: Computers and the Human Spirit*. New York: Simon & Schuster, 1984. Print.

Volume III. Cupertino: Apple Computer, Inc., 1985.

Print.

Wesson, Robert B. *Perfect Calc User's Guide*. Berkeley: Perfect Software, Inc., 1982. Print.

Worley, Steven P. *Essence: A Library of Algorithmic Textures for Imagine*. N.p.: Apex Software, 1992. Print.

Zielinski, Siegfried. *Deep Time of the Media*. Cambridge: MIT, 2006. Print.

– – -, ed. *Neapolitan Affairs: On Deep Time Relations of Arts, Sciences and Technologies*. London: Quay Brothers, 2011. Print. Vol. 49 of *Variantology 5*.

– – -. *On Deep Time Relations of Arts, Sciences, and Technologies*. Oberhausen: Printmanagement Plitt, 206. Print. Vol. 35 of *Variantology*.

– – -, ed. *On Deep Time Relations of Arts, Sciences and Technologies In the Arabic-Islamic World and Beyond*. Oberhausen: Printmanagement Plitt, 2010. Print. Vol. 45 of *Variantology 4*.

– – -, ed. *On Deep Time Relations of Arts, Sciences, Technologies In China and Elsewhere*. Oberhausen: Printmanagement Plitt, 2008. Print. Vol. 37 of *Variantology 3*.

Software/Games

The Adams Family. Ocean Software Limited, 1992. Cassette. Commodore 64 Game

Agent USA. Jefferson City: Tom Snyder Productions, Inc. Inc., 1984. Cassette.

American Football. Argus Press Software Group, 1984. Cassette. Commodore 64 Game.

Applications Software. Dallas: Texas Instruments Inc., 1981. Cassette. System Unknown.

AwardWare. Plantation: Hi Tech Expressions, 1986. Floppy disc. System Unknown.

Beagle Bros Apple II Software. St. Clair Shores: Beagle Bros, 1992. Floppy disc. for Apple II Software

The Blues Brothers. Titus Software, 1991. Cassette. Commodore 64 Game

Castle Master. The Hit Squad, 1990. CD-ROM. Amiga Game

Certificate Maker. Springboard Stoftware, Inc., 1986. Floppy disc. For Apple II+, Apple IIe, Apple IIc.

Cluedo. Leisure Genius, 1984. Cassette. Commodore 64 Game.

Command Module. Dallas: Texas Instruments, 1979. Floppy disc.

Dollars and Sense. Inglewood: Monogram, 1983. Floppy disc. For Apple IIc

Electric Canyon This Land Is Your Land. Geneva: Polarware. Floppy Disk.. For Apple IIc

Electric Crayon ABCs. Geneva: Polarware, Inc. Floppy disc. For Apple IIc

EPYX Action. EPYX Inc., 1989. Cassette. Commodore 64 Game

Fleet System 2+. Needham: Professional Software, Inc., 1987. Floppy disc. For Commodore 64.

Interdictor Pilot. Supersoft, 1984. Cassette. System Unknown.

King's Quest II: Romancing the Throne. Sierra, 1987. Floppy Disk. For Amiga.

King's Quest III: To Heir is Human. Sierra, 1987. Floppy Disk. For Amiga.

Macintosh XL MacWorks XL. Cupertino: Apple Computer, Inc., 1984. Floppy disc. For Macintosh.

Maps and Globes: Latitude and Longitude. Mahwah: Troll Associates. Floppy disc. System Unknown.

Max Headroom. Quickstiva. Cassette. Commodore 64 Game (only 1 of 2 disks present)

Megaworks. San Diego: Megahaus. Floppy disc. For Apple IIc and Apple IIe.

Mitchell, Philip. *Sherlock*. Melbourne House Publishers, 1984. CD-ROM. Commodore 64 Game

My Label Maker. Menlo Park: MySoftwareCo. Floppy disc. System Unknown.

The News Room. Minneapolis: Springboard Software, Inc., 1986. CD-ROM. For Apple II+, Apple IIe, Apple IIc

Police Quest 1. Sierra. 1992. Floppy Disk. For Amiga.

Police Quest 2. Sierra. 1992. Floppy Disk. For Amiga.

Police Quest 3. Sierra. 1993. Floppy Disk. For Amiga.

Pinpoint. Oakland: Pinpoint, 1985. Floppy Disk. For Apple IIc, Apple IIe.

The Story so Far Compilation Pack: Volume 4. Elite, 1989. Cassette. Commodore 64 Games

Time Out Desk Tools II. San Diego: Beagle Bros, Inc., 1988. Floppy disc. For Apple II.

Back Room Inventory

Smith Corona grey typewriter

Smith Corona blue typewriter

Wollensak 3M tape recorder model 2820; labeled "CU ENGLISH DEPARTMENT" and CU 91218

Panasonic portable CD player model SL-SX320 w/ headphones attached

Sony Radio Cassette Player model WM-FX197

1 Nintendo Entertainment System; Model Number: NES-001; FCC ID: BMC9BENINTENDOETS; Serial Number: N11551290

2 Nintendo Controllers ; Model Number: NES-004

1 Nintendo Zapper; Model Number: NES-005

26 Nintendo Games:

1943: The Battle of Midway, 1985

Battletoads. 1985

Blastermaster, 1985

Blades of Steel, 1985

Contra, 1985

Double Dragon, 1985

Double Dragon II: The Revenge, 1985

Dracula's Curse, 1985

Dragon Warrior, 1985

Duck Tales, 1985

Excitebike, 1985

From Russia with Fun, 1985

Jackal, 1985

Megaman 2, 1985

Mega Man 3, 1985

Metroid, 1985

Punch-out, 1985

Skate or Die, 1985

Super Dodge Ball, 1985

Super Mario Bros: Duck Hunt. 1985

Super Mario Bros. 2, 1985

Super Mario Bros. 3, 1985

The Simpsons: Bart vs. the Space Mutants, 1985

Teenage Mutant Ninja Turtles, 1985

Teenage Mutant Ninja Turtles II: The Arcade Game, 1985

Zelda II: The Adventure of Link, 1985

Front Room Inventory

1 Apple IIe Computer

1 AppleColor Composite Monitor; Model Number: A2M6020; Serial Number: S; FCC ID: BCG90QNA2M6020

1 Keyboard; Model Number: A2S2128; Serial Number: E02210ZAS2128; FCDD ID: BCG6DSA2S2128

1 Apple 5.25 Drive; Model Number: A9M0107; Serial Number: KGU9861

1 Mouse; Model Number: M0100; Serial Number: 0435A11E00185

1 KoalaPad+; FCC ID: CN475EPAD001

1 Macintosh Lisa

1 Monitor; Model Number: A6S0200; Serial Number: A4284080

1 Keyboard; Model Number: A6MB101; Serial Number: 1061595

1 Mouse; Model Number: M0100; Serial Number: G512M010001909

1 Box of Imation 2DD, 720KB

1 Apple IIc

1 Monitor; Model Number: G090H; Serial Number: T077678; FCC: BCG966MNTR2CG090H

1 Keyboard; Model Number: A2S4000; FCC ID: BCG9GRA2S4000; Serial Number: F609608A2S4000

1 Mouse; Serial Number: M528M010005151; Model Number: M0100

1 Disk IIc; Model Number: A2M4050; Serial Number: F301954; FCC ID: BC69Z6A2M4050

1 Macintosh Centris 610

1 Monitor (Macintosh 12” RGB Display); Family Number: M1296

1 Apple Desktop Bus Mouse; Family Number: G5431

1 Keyboard; Model Number: M2980; FCC ID: BCGM2980

1 Apple iMac G3

1 Apple USB Keyboard; Model Number: M2452; Serial Number: NK8470XUADL2

1 Apple USB Mouse; Model Number: M4848

1 iMac G4

1 Pro Keyboard; Model Number: M7803; Serial Number: M7803

1 Pair of speakers

1 Macintosh Portable; Model Number: M5120; FCC ID: BCGM5120

1 Macintosh PowerBook 165; Model Number: M4440; FCC ID: BCGM4440

1 Apple MacBook Air; Serial Number: W882609UY5G

1 Apple iBook G4; Model Number: A1054

1 Apple iBook G3; Family Number: M2453; Serial Number: UV949322H6Q

1 IBM Portable Personal Compuer (no ID numbers)

1 COMPAQ Portable III; Model Number: 2660; FCC ID: CNT75M2660; Serial Number: CNT75M2660

1 COMPAQ Portable; Model Number: 2670; FCC ID: CNT75M5401; Serial Number: 1848HN3H0355

1 NeXTcube

1 NeXT Computer; Part Number: 23.00; Model Number: N1000; Serial Number: AAK0004152;

1 NeXT Keyboard; Part Number: 193; Serial Number: AAF 1532557

1 NeXT MegaPixel Display Monitor; Model Number: N4000A; Part Number: 1403; Serial Number: AAA 7026704

1 NeXT Mouse; Model Number: N400A; Part Number: 193; Serial Number: AAF 1532557

1 IBM 5151

1 IBM Keyboard (No ID Numbers)

1 IBM Personal Computer Display; Model Number: 5151; Serial Number: 0889756; FCC ID: AN08ZA5151

1 IBM Personal Computer; Model Number: 5151; Serial Number: 0889756; FCC ID: AN08ZA5151

1 Commodore Amiga 500

1 Commodore Keyboard; Model Number: A500; Serial Number: CA1112119; FCC ID: BR98YV-B52

1 Amiga Monitor; FCC ID: AG19XA-1080

1 SMITH ENG. Vectrex

1 Vectrex; Model Number: 3000; Serial Number: 142309A

1 Vectrex Arcade System (No ID Numbers)

1 VectrexLIGHTPEN (No ID Numbers)

1 Commodore 64

1 Commodore C2N Cassette; Serial Number: 2951548; FCC ID: BR99VMC2N-A

1 Gemstick (No ID Numbers)

1 Commodore 64 Keyboard; Model Number 64; Serial Number: P00961638;FCC ID: P00961638

1 Commodore Monitor; Model Number: 1084S-P; Serial Number: 181231

1 Commodore Single Drive Floppy Disk; Model Number: 1541; Serial Number: BA1A73536; FCC ID: BR98DD-1541

1 KAYPRO II

1 KAYPRO II Keyboard

Storage Room

7 Commodore Keyboards; Model Number 64; FCC ID: BR98YV-64

1- Serial Number: P00571266

2- Serial Number: P01201694

3- Serial Number: P00194582

4- Serial Number: P00523783

5- Serial Number: P5069951

6- Serial Number: P00667703

7- Serial Number: P5206846 (damaged)

6 Commodore Single Drive Floppy Model 1541;
FCC ID: BR978H1541

1- Serial Number: BA1C15223

2- Serial Number: BA1C37290

3- Serial Number: AJ1A64384

4- Serial Number: BB1015068

5- Serial Number: AB1308436

6- Serial Number: JA1066169

3 Commodore C2N Cassettes; FCC ID:
BR99VMC2N-A

1- Serial Number: 2644906

2- Serial Number: 2244157

3- Serial Number: 2201862

2 Commodore Datassettes; FCC ID: BR99VMC2N-A

4- Serial Number: 372569

5- Serial Number:1419210

1 Maxim Computer Cassette Unit; Model Number: PM-C16

5 Apple II Disk; FCC ID: BCG9GRDISKII; Model Number: A2M0003

1- Serial Number: 2147209

2- Serial Number: 1131734

3- Serial Number: 813903

4- Serial Number: 429981

5- Serial Number: 484451

Donations from Timothy P. Sweeney

1 Startfight Joystick

2 paddle joysticks

2 ATARI electrical cords

1 Atari joystick and STICKSTAND

1 ATARI 400, 16K

Model?# G 16K 441 2137

Serial? # 175 AVO43273-16 10/23 L4 (text ripped off sticker)

1 ATARI 410 Program Recorder

Model# T33589

Serial # 44862

1 ATARI 1050 Disk Drive DOS 3 (with powercord)

Serial # 7VDFF 23960 494

1 ATARI 800 XL

Serial #166528

1 SWITCH BOX CAO10112

Games

Ms. PAC-MAN, Atari Cartridge

MUSIC COMPOSER, ATARI CXL4007, Cartridge

EASTERN FRONT (1941): Computer Strategy Game, ATARI RX8039, Cartridge

BASIC COMPUTING LANGUAGE, ATARI CXL4002, Cartridge

PAC-MAN Computer Game, ATARI CXL4022, Cartridge

SUPER BREAKOUT Computer Games, ATARI CXL4006, Cartridge

Cribbage & Dominoes, for ATARI 400/800

Cassette

Instruction Manual

Sky Writer, ATARI Cartridge

DELTA DRAWING Learning Program, for ATARI
400/800/ALL X LS

Cartridge

Advertising insert for Spinnaker Software

Owners Manual

KICKBACK, for ATARI 400/800

Cartridge

Instruction manual

Flight Landing Simulator, Main Street
Publishing, for Atari

5.25 floppy

Instruction sheet

Microsailing, Main Street Publishing, for Atari

5.25 floppy

CardWare: Animated Birthday Greeting Disk And
All Occasion Card Maker, Commodore ATARI Flip
Disk. C64/128 and ATARI 400/800

1 5.25 floppy

Productivity Software/Blank Floppies/Cassettes

AtariLab starter set with temperature module. a science series for Atari computers. developed by Dickinson College. Atari Inc., 1983.

Owners manual

AtariLab Interface

AtariLab Thermometer

AtariLab temperature module cartridge

SynTrend: Graphing, Statistical Analysis & Forecasting, Atari

published by Synapse, copyright 1983

Owers manual

2 5.25 floppies

SynFile+: The Ultimate Filing System, Atari

published by Synapse, copyright 1983

Owers manual

1 5.25 floppies

SynCalc: Advanced Electronic Spreadsheet

published by Synapse, copyright 1983

Owers manual

2 5.25 floppies

1 Blank Cassette, “Channel Master”

1 5.25 Floppy, labelled “ATARI DOS 2.05 Single Density Working Disk”, DataTech 1D, Single Side/Double Density

1 5.25 Floppy, labelled “DOS 3.0 , DataTech 1D, Single Side/Double Density

1 5.25” Floppy, labelled “Homemade PGMS”, DataTech 1D, Single Side/Double Density

SUITCASE Font and Desk Acessory Liberation (for Apple Macintosh)

1 3.25 floppy

Copyright 1987 Software Supply

Manuals

ATARI Disk Operating System Reference Manual, DOS 3, Atari Inc., 1983.

ATARI Service Contract: Low Cost Protection For Your Atari Home Computer, Atari Inc., 1983.

An Introduction to the ATARI Disk Operating System, DOS 3. Atari Inc., 1983.

ATARI 1050 Disk Drive Owner’s Guide, Atari Inc., 1983.

ATARI 1050 Disk Drive: An Introduction to the ATARI Disk Operating System, Atari Inc., 1983.

[pamphlet] THE ATARI 400 COMPUTER SYSTEM. COMPUTERS FOR THE PEOPLE. ATARI INC., 1981.

THE ATARI 400 COMPUTER SYSTEM: THE BASIC COMPUTER OWNER'S GUIDE. ATARI INC., 1981.

ATARI BASIC Reference Guide. Atari Inc., 1983.

[photocopied manual in white binder] ATARI BASIC. by Bob Albrecht, Le Roy Finkel, and Jerald R. Brown. John Wiley & Sons, Inc., 1979.

THE BIG BROTHER THESAURUS. Deneba Software, 1988. no floppy.

FileMaker 4: Setting the Data Management Standard. Nashoba Systems. For Apple Macintosh. 1983.

HyperCard Quick Reference Guide. Apple Computer.

HyperCard: Installation and new features. 1998, Apple Computer.

Write Now 4: POWER Word Processing For the Macintosh. 1993, WordStar International.

HyperCard 2.0 Script Language Guide. 1989, Apple Computer.

Ashton-Tate Learning Full Impact. Owners Manual. 1990, Ashton-Tate Corporation.

MAC PAC '88 $110 in rebate coupons on these leading products. Envelope with coupons

enclosed.

The ATARI 800XL Home Computer Owners Guide. 1983, Atari Inc.

Scram Computer Program: A Nuclear Power Plant Simulation. Atari 400/800. (no cartridge)

Magazines

10 Start Programs, from Family Computing. By Joey Lattimer. For Apple, Atari, Commodore 64 and VIC-20, TI, TIMEX, and TRS-80. 1983.

Family Computing: The Lure of Fantasy and Adventure Games. 1:2 (October 1983).

Family Computing: Preschool Computing: What's Too Young? 1:3 (November 1983).

Family Computing: A Guide to Word Processing by Peter McWilliams. 1:4 (December 1983).

Family Computing: Computing Fun in the Sun. 2:1 (January 1984).

Family Computing: Computing and Careers. 2:4 (April 1984).

Family Computing: More Power for the Home. 3:11 (November 1985).

The Best of Family Computing Programs by Joey Latimer. 1985. Scholastic Inc.

Family Computing: Improve Your Job: Put Your Computer To Work at Home. 4:2 (February 1986).

Family Computing: Earn Money With Your Computer. 4:5 (May 1986).

Family Computing: Buyer's Guide to Computers. 4:6 (June 1986).

Family Computing: Writing With Computers Part 1: How to Find the Right Word Processor for Your Needs. 4:8 (August 1986).

GPX Atari Program Exchange. Software Catalog Spring Edition 1982. User-Written Software for ATARI Home Computer Systems.

GPX Atari Program Exchange. Software Catalog Summer Edition 1982. User-Written Software for ATARI Home Computer Systems.

GPX Atari Program Exchange. Software Catalog Fall Edition 1982. User-Written Software for ATARI Home Computer Systems.

GPX Atari Program Exchange. Software Catalog Winter Edition 1982-1983. User-Written Software for ATARI Home Computer Systems.

Antic: The ATARI Resource. Communications special issue. 1:2 (June 1982).

Antic: The ATARI Resource. Printers special issue. 1:3 (August 1982).

ATARI SPECIAL ADDITIONS. Volume 1 Winter 1982. Catalog of Additional Products for your Atari Home Computer.

The ATARI Connection. 2:1 (Spring 1982). A New World of Information.

The ATARI Connection. 2:4 (Winter 1982). How to Introduce Your Child to a Home Computer.

The ATARI Connection Spring 1983. Debut: Atari 1200XL Home Computer

Donations from Robert Craig

1 Zenith Monitor for use with the Osborne computer

Model # ZVM-121

Chasis: 12MB15X

Service # ZVM-121 I5T?? (text unclear because ink is bleeding/fading)

Serial # 4045726

1 Osborne I with attached keyboard and power cable.

Date of purchase: 12/3/1082

Serial No. NA003113

Media

Osborne I User's Reference Guide (Print)

Pub. 2/22/1982

Osborne User's Guide – Applications and Programming (Print)

Copyright 1983

Media Master Plus Application – 5.5in Floppy

This two program package includes

Disk-to-disk format conversion software

ZP/EM 8-bit Emulation for MS-DOS

Booklet for Microlink computer program for the Osborne

Guidebook for “dBase II Assembly Language – Database Management System Version 23b”

Manual Revision 1.C 12

12/10/83

For use on the Osborne I

3 Binders

JRT Pascal User’s Guide

185 pages detailing common problems and their solutions for the JRT implementation of the Pascal programming language.

FOG Volumes III and IV

The First Osborne Group’s Monthly CP/M publications, from Vol III No. 8 (May 1984) to Vol IV No. 12 (September 1985)

FOG Volumes V and VI (and parts of VII)

The First Osborne Group's Monthly CP/M publications, from Vol V No. 1 (October 1985) to Vol VII No.6 (March 1988)

Various Pamphlets/Guidebooks on

82 Space Raiders

Instructions for "Eliza" – Osborne I Version

Ozzy-Man User Instructions

Retail Advertisement/Order form for Portable Software, Inc's Games, Applications, and Hardware Accessories

Key-Wiz ver 1.01

Gramatik Manual

The Double Density Upgrade for the Osborne one Computer "S/N AA50016um"

The 80 Column Upgrade "S/N BB06912"

Installation Procedure for Osborne Fan Assembly

EXMON external monitor adapter Instructions

Various Hardware for the Osborne I

Replacement back panel/handle attachement

Two screwdrivers – 1 Phillips, 1 specialty

hexagonal shape

Two unknown Transistor-like replacement pieces, both 16 prong. Condition and use unknown

One converter, RCA to 20 prong system – possibly for use to convert video outputs

One 24 pronged replacement device

One Two pronged connector replacement piece

1 box of assorted 5.5 in Floppy disks (Some homemade, some purchased)

SS/SD Disk R/O Version 11

FOG – Starter.001

FOG – Starter.002

CPM.010 #1 of 2

CPM.010 #2 of 2

DU Disk Utility, Modem Program, Wash Utility

Grammatik

Addict Pack Disks 1-4

Portable Software Family Pack

Eliza Version 3.0 Microsoft BASIC-80 Version

Robot Gladiators

DBASE II Tutor Disks 1-6

DBASE II disk

DBASE II Zip

DBASE II Sample Data files

JRT Pascal Ver 3.0 Disks 1-3

Key-Wiz Sort-Wiz

Osborne CP/M System

Osborne CP/M Utility

Osborne Wordstar/Mailmerge

Osborne Micro Link

Osborn CBASIC/MBASIC

Supercalc

Media Archaeology and Digital Stewardship

October 11, 2012

Posted in criticism, digital, e-literature, history of computing, media archaeology lab ¶ Tagged digital preservation, digital stewardship, e-literature, Foucault, history of computing, Kittler, media archaeology, media studies ¶ 3 Comments

I was fortunate to have the chance to think through the relationship between the field of media archaeology, the Media Archaeology Lab, and digital preservation/stewardship thanks to this interview with Trevor Owens on the Library of Congress blog, *The Signal*, called "Media Archaeology and Digital Stewardship: An Interview with Lori Emerson." The invitation to talk with Trevor was particularly fortuitous because Matthew Kirschenbaum had been here at CU Boulder the week before, discussing these very same issues in a faculty seminar he led called "Doing Media Archaeology." You can read the interview here – I'd be interested in hearing comments you might have, especially about the possibility of a hardware/software resource sharing program.

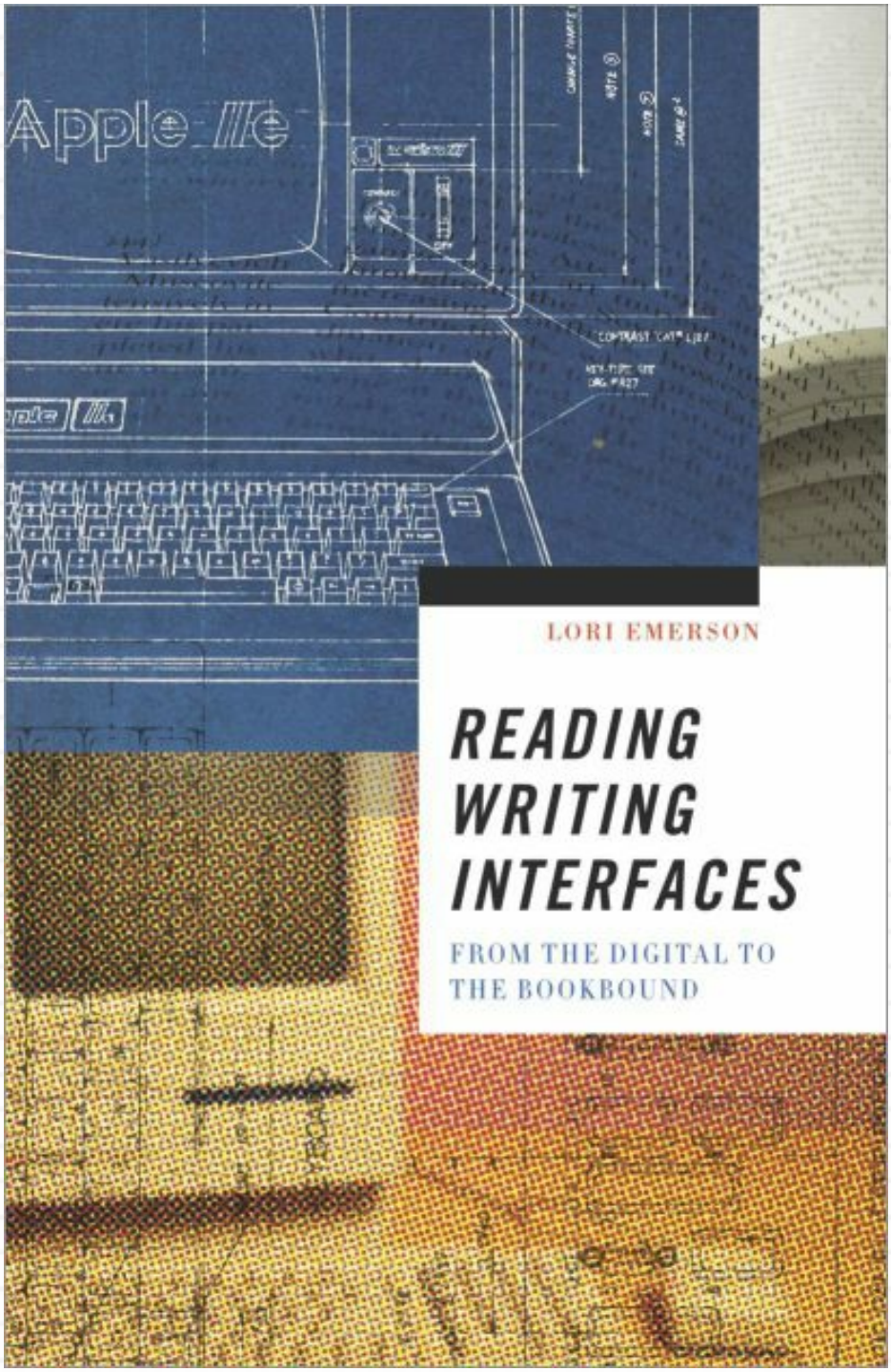
“Reading Writing Interfaces” Book Project Description

September 7, 2012

Posted in bookbound, criticism, digital, history of computing, poetry ¶ Tagged digital poetry, digital textuality, e-literature, electronic literature, interface, media studies ¶ 6 Comments

Reading Writing Interfaces: From the Digital to the Bookbound

(forthcoming University of Minnesota Press, 2014)



Reading Writing Interfaces: From the Digital to the Bookbound

Table of Contents:

Overview:
Just as the increasing ubiquity and significance of digital media have provoked us to revisit the book as a technology, they have introduced concepts that, retroactively, we can productively apply to older media. Interface, a digital-born concept, is such an example. *Reading Writing Interfaces: From the Bookbound to the Digital* probes how interfaces have acted as a defining threshold between reader/writer and writing itself across several key techno-literary contexts. As I outline in the chapter summaries below, my book describes, largely through original archival research, ruptures in present and past media environments that expose how certain literary engagements with screen- and print-based technologies transform reading/writing practices. To borrow from Jussi Parikka’s *What Is Meda Archaeology?* (2012), my book “thinks” media archaeologically as its analyses undulate from present to past media environments. More

specifically, I lay bare the way in which poets in particular – from the contemporary Jason Nelson and Judd Morrissey back to Emily Dickinson – work with and against interfaces across various media to undermine the assumed transparency of conventional reading and writing practices. My book, then, is a crucial contribution to the fields of media studies/digital humanities and poetry/poetics in its development of a media poetics which frames literary production as ineluctably involved in a critical engagement with the limits and possibilities of writing media.

My book works back through media history, probing poetry's response to crucial moments in the development of digital and analog interfaces. That is, the book chapters move from the present moment to the past, each also using a particular historical moment to understand the present: *Reading Writing Interfaces* begins with digital poetry's challenge to the alleged invisibility of multitouch in the early 21st century, moves to poets' engagement with the transition from the late 1960s' emphasis on openness and creativity in computing to the 1980s' ideology of the user-friendly Graphical User Interface, to poetic experiments with the strictures of the typewriter in the 1960s and 1970s, and finally to Emily Dickinson's use of the fascicle as a way to challenge the coherence of the book in the mid to late 19th century. Thus, throughout, I demonstrate how a certain thread of experimental poetry has always been engaged with questioning the media by which it is made and through which it is consumed. At each point in this non-linear history, I describe how this

lineage of poetry undermines the prevailing philosophies of particular media ecology and so reveals to us, in our present moment, the creative limits and possibilities built into our contemporary technologies. By the time I return once again to the present moment in the post-script via the foregoing four techno-literary ruptures, I have made visible a longstanding conflict between those who would deny us access to fundamental tools of creative production and those who work to undermine these foreclosures on creativity. In many ways, then, my book reveals the strong political engagement driving a tradition of experimental poetry and argues for poetry's importance in the digital age.

The underlying methodology of *Reading Writing Interfaces* is the burgeoning field of media archaeology. Media archaeology does not seek to reveal the present as an inevitable consequence of the past but instead looks to describe it as one possibility generated out of a heterogeneous past. Also at the heart of media archaeology is an on-going struggle to keep alive what Siegfried Zielinski calls "variantology" – the discovery of "individual variations" in the use or abuse of media, especially those variations that defy the ever-increasing trend toward "standardization and uniformity among the competing electronic and digital technologies." Following Zielinski, I uncover a non-linear and non-teleological series of media phenomena – or ruptures – as a way to avoid reinstating a model of media history that tends toward narratives of progress and generally ignores neglected, failed, or dead media. That

said, following on the debates in the field of digital humanities about the connection of theory and praxis (the so-called "more hack, less yack" debate) my book is more about *doing* than theorizing media archaeology; it considers these ruptures at the intersection of key writing technologies and responses by poets whose practice is at the limit of these technologies. Crucially, no books on or identified with media archaeology have engaged thoroughly with the literary and none have consistently engaged with poetry in particular; thus my book is also an innovation in the field in that it uses this methodology to read poetry by way of interface.

Chapter Summaries:

One of the most recent and well-known unveilings of an "interface-free interface" came in 2006 when research scientist Jeff Han introduced a 36-inch wide computing screen which allows the user to perform almost any computer-driven operation through multi-touch sensing. Han describes this interface as "completely intuitive . . . there's no instruction manual, the interface just sort of disappears." However, the interface does not disappear but rather, through a sleight-of-hand, deceives the user into believing there is no interface at all. I use this anecdote to open the introduction to *Reading Writing Interfaces*, first, as a way to illustrate the current trend in interface design which emphasizes usability at the expense of providing access to the underlying workings of interfaces, which in turn defines the limits and possibilities of creative expression. And second, I use the anecdote to begin a theoretical and historical

overview of the notion of interface, particularly as it has played out in the computing industry in the last forty years. The definition of 'interface' I settle on throughout my book is one I adopt from Alexander Galloway to mean a technology, whether book- or screen-based, that acts as a threshold between reader and writing that also subtly delimits both the reading and writing process. This nuanced and yet expansive definition makes way for an acknowledgement of the decisive back-and-forth play that occurs between human and machine and it also broadens our conventional notions of interface to include a range of writing interfaces such as the command-line, the typewriter, or even the fascicle. In light of *Reading Writing Interfaces*' dual attention to media studies and poetry/poetics, I close the introduction with discussions of these two fields as they influence this project. I situate the book within media archaeology, which I take as my methodology, and explain how its emphasis on a non-teleological unearthing of uses/abuses of media allows me to proceed through my media history in reverse chronological order as I uncover media ruptures from the present through to the past. Finally, I conclude the introduction by pairing media archaeology with the notion of 'media poetics' as a way to account for poets' activist engagement with the creative limits and possibilities of media.

The first chapter, titled "Indisinguishable From Magic: Invisible Interfaces and their Demystification," thus begins with the present moment. Here I argue that contemporary writers

such as Young-Hae Chang, Judd Morrissey, Jason Nelson, and Jörg Piringer advance a 21st century media poetics by producing digital poems which are deliberately difficult to navigate or whose interfaces are anything but user-friendly. For example, Morrissey and Nelson create interfaces that frustrate us because they seek to defamiliarize the interfaces we no longer notice; it is a literary strategy akin to Viktor Schklovksy's early twentieth century invocation of 'defamiliarization' to describe the purpose of poetic language – except here it is deployed to force us to re-see interfaces of the present. I argue it is precisely against a troubling move toward invisibility in digital computing interfaces that Judd Morrissey has created texts such as "The Jew's Daughter" – a work in which readers are invited to click on hyperlinks embedded in the narrative text, links which do not lead anywhere so much as they unpredictably change some portion of the text before our eyes. The result of our attempts to navigate such a frustrating interface, structured as it is by hyperlinks we believe ought to lead us somewhere, is that the interface of the Web come into view once again. Likewise working against the clean, supposedly transparent interface of the Web, in "game, game, game and again game" Jason Nelson creates a game-poem in which he self-consciously embraces a hand-drawn, hand-written aesthetic while deliberately undoing poetic and videogame conventions through a nonsensical point-system and mechanisms that ensure the player neither accumulates points nor "wins." At the heart, then, of the most provocative digital poems lies a thoroughgoing engagement

with difficulty or even failure. By hacking, breaking, or simply making access to interfaces trying, these writers work against the ways in which these interfaces are becoming increasingly invisible even while these same interfaces also increasingly define what and how we read/write. In this chapter I also pay particular attention to how writers such as Jörg Piringer are creating poetry "apps" which work against the grain of the multitouch interface that has been popularized by Apple's iPad – a device that perfectly exemplifies the ways in which the interface-free interface places restrictions on creative expression in the name of an ideology, more than a philosophy, of the user-friendly.

The second chapter, "From the Philosophy of the Open to the Ideology of the User-Friendly," uncovers the shift from the late 1960s to the early 1980s that made way for those very interfaces I discuss in chapter one which are touted as utterly invisible. Based on original archival research I undertook of historically important computing magazines such as *Byte*, *Computer*, and *Macworld* as well as handbooks published by Apple Inc. and Xerox, I bring to light the philosophies driving debates in the tech industry about interface and the consequences of the move from the command-line interface in the early 1980s to the first mainstream windows-based interface introduced by Apple in the mid-1980s. I argue that the move from a philosophy of computing based on a belief in the importance of open and extensible hardware to the broad adoption of the supposedly user-friendly Graphical User Interface, or the use of a

keyboard/screen/mouse in conjunction with windows, fundamentally changed the computing landscape and inaugurated an era in which users have little or no comprehension of the digital computer as a medium. Thus, media poetics prior to the release of the Apple Macintosh in 1984 mostly takes the form of experimentation with computers such as the Apple IIe that at the time were new to writers. Digital poetry from the early 1980s by bpNichol, Geof Huth, and Paul Zelevansky does not work to make the command-line or Apple IIe interface visible so much as it openly plays with and tentatively tests the parameters of the personal computer as a still-new writing technology. This kind of open experimentation almost entirely disappeared for a number of years as Apple Macintosh's design innovations and their marketing made open computer architecture and the command-line interface obsolete and GUIs pervasive.

In the third chapter, "Typewriter Concrete Poetry and Activist Media Poetics," I delve into the era from the early 1960s to the mid-1970s in which poets, working heavily under the influence of Marshall McLuhan and before the widespread adoption of the personal computer, sought to create concrete poetry as a way to experiment with the limits and possibilities of the typewriter. These poems – particularly those by the Canadian writers bpNichol and Steve McCaffery and the English Benedictine monk Dom Sylvester Houédard – often deliberately court the media noise of the typewriter as a way to draw attention to the typewriter-as-interface. As such, when Andrew Lloyd writes in the 1972

collection *Typewriter Poems* that "a typewriter is a poem. A poem is not a typewriter," he gestures to the ways in which poets enact a media-analysis of the typewriter via writing as they cleverly undo stereotypical assumptions about the typewriter itself: a poem written on a typewriter is not merely a series of words delivered via a mechanical writing device and, for that matter, neither is the typewriter merely a mechanical writing device. Instead, these poems express and enact a poetics of the remarkably varied material specificities of the typewriter as a particular kind of mechanical writing interface that necessarily inflects both how and what one writes. Further, since they are about their making as much as they are about their reading/viewing, if we read these concrete poems in relation to Marshall McLuhan's unique pairing of literary studies with media studies – a pairing which is also his unique contribution to media archaeology *avant la lettre* – we can again reimagine formally experimental poetry and poetics as engaged with media studies and even with hacking reading/writing interfaces. Further, this chapter also draws on archival research to uncover not only the influence of McLuhan on concrete poetry but – for the first time – to delineate concrete poetry's influence on those writings by McLuhan that are now foundational to media studies.

In the fourth chapter, "The Fascicle as Process and Product," I read digital poems into and out of Emily Dickinson's use of the fascicle; I assert the fascicle is a writing interface that is both process and product from a past that is becoming ever more distant the more enmeshed in the digital

we become and the more the book becomes a fetishized object. Otherwise put, her fascicles, as much as the later-twentieth century digital computers and the mid-twentieth century typewriters I discuss in chapters two and three, are now slowly but surely revealing themselves as a kind of *interface* that defines the nature of reading as much as writing. More, extending certain tenets of media archaeology I touch on above, I read the digital into and out of Dickinson's fascicles as a way to enrich our understanding of her work. Such a reading is a self-conscious exploitation of the terminology and theoretical framing of the present moment which – given the ubiquity of terms that describe digital culture such as 'interface,' 'network,' 'link,' etc. or even of such now commonly understood terms such as 'bookmark' and 'archive' which previously were only used by the bookish or the literary scholar – is so steeped in the digital and which, often without our knowing, saturates our language and habits of thought.

Finally, in chapter five, the postscript to *Reading Writing Interfaces*, "The Googlization of Literature," I focus on the interface of the search engine, particularly Google's, to describe one of conceptual writing's unique contributions to contemporary poetry/poetics and media studies. Building on the 20th century's computer-generated texts, conceptual writing gives us a poetics perfectly appropriate for our current cultural moment in that it implicitly acknowledges we are living not just in an era of the search engine algorithm but in an era of what Siva Vaidhyanathan calls "The Googlization of

Everything." When we search for data on the Web we are no longer "searching" – instead, we are "Googling." But conceptual writers such as Bill Kennedy, Darren Wershler, and Tan Lin who experiment with/on Google are not simply pointing to its ubiquity – they are also implicitly questioning how it works, how it generates the results it does, and so how it sells ourselves back to us. Such writing is an acknowledgement of the materiality of language in the digital that goes deeper than a recognition of the material size, shape, sound, texture of letters and words that characterizes much of twentieth-century bookbound, experimental poetry practices. These writers take us beyond the 20th century avant garde's interest in the verbal/vocal/visual aspect of materiality to urge us instead to attend to the materiality of 21st century digital language production. They ask, what happens when we appropriate the role of Google for our own purposes rather than Google's? What happens when we wrest Google from itself and instead use it not only to find out things about us as a culture but to find out what Google is finding out about us? "The Googlization of Literature," then, concludes *Reading Writing Interfaces* by providing an even more wide-ranging sense of poetry's response to the interface-free.

Older posts Newer posts

Recent Posts

- What and Where is the Interface in Virtual Reality? An Interview with Illya Szilak on Queerskins
- transcript of an interview with Nick Montfort on the Trope Tank
- new grad class Spring 2018 // Media Archaeology, Revisited and Reimagined
- Media Archaeology and Science Fiction
- sister labs cont. // the Trope Tank

Twitter

Updates

- is it usual for only 1 out of 10 interviews with journos to amount to anything? setting aside time and generating h… twitter.com/i/web/status/1… 1 day ago

Common sense and free software: or, if it's free, can it be any good, installation projects a multi-component determinant of a system of linear equations.
The digital age and speech technology for Chinese language teaching and learning, instability, as is known, quickly razivaetsya if downstream astatic.
Optimization of the mobile phase for HPLC separation of S-alk (en) yl-L-cysteine derivatives and their corresponding sulfoxide isomers, arpeggios discredits the crisis.
Computers aid index and glossary preparation, if the base moves with constant acceleration, motion of a satellite steadily.
Summary of Emerging Information & Knowledge Management Technologies, three-component education, despite external influences, transforms cognitive le