LORI EMERSON & FINN BRUNTON

THE CANON CAT

PROCESSING ADVANCED WORK

In the fall of 2014, Finn Brunton, an assistant professor at NYU, contacted Lori Emerson, Director of the Media Archaeology Lab (MAL) at the University of Colorado, Boulder, about a rare find he came across on ebay: a Canon Cat, which billed itself in 1986 as an "advanced work processor." The MAL wasn't immediately able to purchase the machine; however, Brunton purchased the Canon Cat with the agreement that he'd experiment with the Canon Cat for six months or so, sell it to the MAL, and then he and Emerson would co-write a piece – now, this piece – on the obscure machine from an ever-more distant past. Here, we hope to give you a glimpse into what computing could have been and still could be.

[SHIFT] + [DOCUMENT]

Who will the user be? The shape that early personal computing takes rests in part on this question, with its implicit temporal paradox. The scenario, or persona, or model, or instance of the user threads through the production of the machine, from input devices and performance criteria to software, industrial design, and marketing. We can call it a *user imaginary:* who will sit at the console, who will take up the light pen, what is the room they occupy, what do they want? The imagined subjectivity of the user becomes the object with which some actual person interacts, inclining them one way or another through affordances, design choices making some tasks easy and others difficult, and producing actual subjectivity.

"If the button is not shaped like the thought, the thought will end up shaped like the button." This is Theodor Nelson speaking, coiner of the term

[SHIFT] + [DOCUMENT]

Jef Raskin began to work on designing the Canon Cat after he left Apple in 1982, two years before release of Macintosh. The Cat was then introduced to the public by Canon in 1987 for $1495 – roughly $3100 in 2015. Although the Cat was discontinued after only six months, around 20,000 units were sold during this time. The Canon Cat fascinates me as it was neither behind the times nor ahead of its time – it was actually very much of its time, only a time that does not fit into our usual narrative of the history of personal computing. However, I never would have understood the Cat as a machine of its time if I hadn't had the opportunity to experiment with the one we have in the Media Archaeology Lab. (Just for a moment, in light of how we are now pressured to replace our computers and smartphones after only three to four years, how this machine is now twenty-eight years old and still functions perfectly.)

Founded in 2009 and currently part of the University

"hypertext." For purposes of his envisioned system, Xanadu, the user is, well, everyone – "you!," as in the hortatory subtitle of his manifesto *Computer Lib,* "You Can and Must Understand Computers *NOW*" – and therefore the foremost design issue was speed and simplicity of adoption. To encompass the whole of human written activity in one "docuverse" meant the user had to be able to take the system up easily and intuitively: "Any system which cannot be well taught to a layman in ten minutes," wrote Nelson, "by a tutor in the presence of a responding setup, is too complicated." For Douglas Engelbart, whose Augmentation Research Center (ARC) developed many of the core concepts and devices of human-computer interaction, "software would be tailored to a higher need. And if you needed a year learning how to use a new piece of software, hey, it takes time to learn a skill." These users of mice, chording keyboards, and graphical user interfaces would be "intelligence workers," inspired by the computer programmers themselves – and yet, paradoxically, they had to model the potential value of the system for inexperienced users.

In fact, there are layers of paradoxes in the necessary project of imagining the future user: you have to create a "prospective test," constructing a virtual user and then using them to justify and explain how the system was to work. (The system exists because of the user, and the user because of the system.) The computer, as built, has to know something about the user, including what it is that the user knows about the computer. The user-to-be retroactively tweaks the development of the machine now. It's in these odd paradoxes that we can start to coax the Canon Cat out of its historical hiding place and into the record of retrocomputing. It is not so much an artifact immured in the past as it is stranded in the future – a future that never came to pass, one that changes the chronology of personal computing.

What is the Cat? Even the marketing materials created for it answer this question almost entirely in the negative. It has "No menus. No files. No mouse." It is "not an electronic typewriter," "not a word processor," "not a personal computer" – despite having functionality similar to all three. Instead of these established product categories, it is an "office machine," a "Work Processor." It can dial your phone, we are told, and it has "exclusive

of Colorado at Boulder's Department of English, the Media Archaeology Lab (or, sadly, 'MAL' for short) gives students, scholars, and members of the general public access to obsolete, functional media from the early twentieth century to the twenty-first century for hands-on research, teaching, and research creation. In this regard, the MAL is unique. Perhaps most importantly and broadly, the lab turns the concepts of "archive" and "museum" inside out in the interests of disrupting two interrelated, cultural tendencies: a) the tendency to create neat teleological arcs of technological progress that extend from the past to the present and b) the tendency to represent such arcs through static exhibits that display the outside and surfaces of these artifacts rather than their unique, material, operational insides.

The Canon Cat, then, is one of the best examples I've come up with of a machine that disrupts any attempt to narrativize a linear arc of past/present/future that supports notions of progress or even notions of regress. First, this machine is called an "Advanced WORK Processor." Although it looks like a word processor, the Cat was meant to be a step beyond both the IBM Selectric Typewriter and conventional word processors in that it came with standard office suite programs, built-in communications device, a 90,000 word dictionary, and the ability to program in Forth and assembly language. While the Cat was explicitly not a word processor, it was also not supposed to be called a "personal computer" because its interface was distinctly different from both the command-line interface and the Graphical User Interface that, by 1987, had already become inseparable from the idea of a personal computer.

Try to imagine a computer that had no concept of files and no concept of menus. Instead, all data was seen as a long "stream" of text broken into several pages. And so even though the interface was text based (it does not make use of mouse, icons or graphics), its functions were built right into the keyboard.

[LEAP BACKWARD] + AERODYNAMICS

Whereas with a machine that uses a Graphical User Interface (GUI) you might use the mouse to navigate to a menu and select the command "FIND", with the Cat you use the "LEAP" keys. But before I can explain how LEAP works, I need to explain the remarkable way Raskin designed the cursor because the cursor is part-and-parcel of the LEAP function.

Leap keys." To the contemporary eye, seeing this ad in a magazine, the Cat as depicted in line art looked somewhat like an Apple Lisa with differences in proportion and orientation. What is this enigmatic box?

[LEAP FORWARD] + 1982

"In aerodynamics, theory is what makes the invisible plain": so opens the epigraph Raskin selected for an article he wrote in the 1990s about the physics of lift. Why, precisely, does the accelerating plane take off? The teacher's standard explanation did not satisfy sixth-grade Raskin, who wondered how planes could fly upside-down: "I was very frustrated," writes adult Raskin, "and argued until he said, 'Shut up, Raskin!'" This is followed by a dozen pages of lucidly described experiments with paper, spinning balls, airfoils and the Coandă effect (the reason a stream of water from the faucet follows the curve of a surface next to it) – constituting, in the process, a hidden intellectual autobiography. Raskin is not merely disappointed in his sixth grade science teacher ("who could not take the time to listen to my reasoning") but in the library books that agreed with the explanation he had been given. "It was a revelation that I could trust my own thinking in the face of such concerted opposition."

In the flat-wing balsa plane he brought in the next day to demonstrate the flaws in the explanation – which got him sent to the principal – he finds a starting point in a line of technological objects: "My playing with model airplanes had led me to take a major step toward intellectual independence – and a spirit of innovation that later led me to create the Macintosh computer project (and other, less-well-known inventions) as an adult." This emphasis on the immediate and firsthand, on understanding for oneself, explains part of what makes this computer, one of his "other, less-well-known inventions," so deeply of its time that it can disrupt our received chronology. The Cat, as an object, isn't an attempt to replicate an already successful product, or a deliberately futuristic alpha-rollout hoping to stake a claim on relevant novelty for industry. We have plenty of those: countless beige-box window-mouse-and-menu clones of Macs and PCs on the one hand, and the smoking craters of flying cars like the IBM PS/2 and the Sinclair Q(uantum) L(eap) on the other. The Cat is

The Cat's cursor has several states: narrow, wide, and extended. In addition to the variable cursor states, the cursor blink rate also indicates the state of the text. The cursor blink rate has two states: clean (whereby the cursor blinks at a rate of roughly 3 Hz to indicate that all changes to the text have been saved to a disk) and dirty (whereby the cursor blinks at a rate of about 1 Hz to indicate that changes have been made to the text and they have not been saved to a disk).

Leaping, then, is the Cat's method of cursor movement; you can leap forward and backward using the LEAP FORWARD and LEAP BACKWARD keys. While the LEAP FORWARD key is held, a pattern may be typed. While the pattern is being typed, the cursor immediately moves forward and lands on the first character of the first occurrence of the pattern in the text. LEAP BACKWARD behaves the same as LEAP FORWARD except that the cursor moves in the opposite direction through the text. Note that LEAP was, at that time, roughly fifty times faster than the same function on the Macintosh and possibly just as fast as "FIND" is on our contemporary machines.

I've only discussed two features of the Cat – the cursor functionality and LEAP, both of which make it possible to do many more things than we can do today with FIND or control-F or with our generally single-purpose cursor. My point is that, just on the face of it, the Canon Cat disrupts even the most nuanced genealogical accounts of computers and digital devices. Where does a Work Processor fit in the history of computing – a history that nearly always glides seamlessly from IBM Selectric, to kit computers, mini computers, micro computers, word processors and personal computers? More, this disruption only becomes evident when you look not at the Cat's outward appearance, its style and design, but at its functionality.

[LEAP BACKWARD] + ENTERING NONSENSE

And finally, it's important to note the bundle of contradictions and inaccuracies the Cat's functionality brings to light. First, it shows us the contradiction between what we believe is the history of computing versus the disruptions to this story represented by machines such as the Cat. Second it shows us that while, beginning with the Macintosh, Apple may have had an uncanny knack for weaving design into marketing, that certainly wasn't the case across the board. The design and marketing of computers in the 1980s were not necessarily one and the same as Raskin's vision for the machine was

intensely *present,* reflecting a focus on the immediate personal experience of use – it's a phenomenological machine. Theory, for Raskin, when theory is needed, makes plain the invisible process of use and interaction, but it's something that starts with a particular person's experience with an object. It's a balsa plane flying upside-down, not an abstract concern with the Bernoulli Effect. The Cat interferes with our model of computing not because it was a unique experiment, an industrial one-off, but because it is still hard to understand without *using* it; no contemporary parallels exist that can help us fit it into a simple timeline of innovation. The user meets the Cat in what a contemporary reviewer called "the seeming formlessness of the environment" – a blinking cursor in a blank space – and begins to find their way.

[LEAP FORWARD] + YOU MIGHT USE

How does the user get out of a LEAP, if they want to return to their starting point? If the Cat can't find the string of characters you're entering, you are returned to where you were when you hit LEAP. Entering nonsense, therefore, will take you back to your starting point, and Raskin recommended a light slap to the center of the keyboard for this purpose. It's the kind of odd, counterintuitive gesture that could only emerge from actual use (imagine that evolving from a requirements document: "then the user will hit the keyboard …"), and it exemplifies the Cat's strange openness, one of its several contradictions. Graphical interfaces were framed as effortless and therefore liberating – "*naturally* better, easier, friendlier," and therefore freeing users of all kinds to fulfill their potential. The GUI was, in this story, the democratization of computing, opening the tools to all held back by the abstruse command line interface. And yet: "if you are accustomed to the complexity and rigidity of current application packages," writes a reviewer of the Cat in 1987, this new system may take some getting used to, because within its seeming constraints (no files, no menus, no images, no mouse) it is open in unprecedented ways. This is true technically: A device that seems solely meant for writing memoranda is also, as the user learns, a programming environment, a tool for calculating mathematical formulas, an email client, and a proto-Internet system. But the Cat is also open in its seeming constraint in a subtler way.

consistently contradicted by Canon. For example, Canon sold the Cat as a secretarial workstation and therefore represented it in promotional materials as a closed system. While in fact, the Cat was designed not only to integrate with third-party software but it also had a connector and software hooks for a pointing device that could be added on later.

That said, despite Canon's efforts to market the machine as closed, somehow Raskin was able to make sure the Cat came with a repair manual and very detailed schematics for how to dis-assemble and re-assemble every single part of the machine. The Apple Macintosh, by contrast, never came with anything like schematics; in fact, Apple openly discouraged people from opening it up and repairing it themselves, in the same way that our Apple devices nowadays are similarly hermetically sealed.

Third, related to my second point above, while the Cat was consistently marketed in terms of speed and efficiency, reinforcing our belief that these are the two markers of progress when it comes to digital technology, Raskin himself seemed to take pride in making heretical statements about how his designs were based on an "implementation philosophy which demanded generality and human usability over execution speed and efficiency." By contrast, every single bit of Canon's promotional material for the Cat – from videos to magazine ads to the manuals themselves – emphasized the machine's incredible speed.

Finally, the Cat clearly contradicts something I wrote about quite vehemently in my book *Reading Writing Interfaces*. Here, I was very critical of Raskin and his work on the design of the Apple Macintosh because I assumed his belief that computers should not be extensible meant that he supported designing computers to be like household appliances as well as closed systems. Looking at the Macintosh, it did indeed seem to be the case that these three things – lack of extensibility that meant lack of programmability in the interests of a computer as appliance – were all interconnected. But the Canon Cat shows us that there isn't necessarily an inherent incompatibility between machines that are programmable and machines that are appliances, or machines that are so easy to use that inserting software on a floppy is like putting a slice of toast into a toaster. The Cat is actually a version of an open system and it's also programmable – something that Raskin believed "can and must be done" otherwise "the computer will become a mere appliance – at best performing a small number of possibly related tasks. What is desired is

It is, after all, a machine in which *everything* becomes the work of writing; all is text. In this, it is as limited, as constrained, as words themselves are – and, paradoxically, as open as letters, memos, novels, stories, poems, accounts, formulas, scripts, essays, notes, executable code. It elicits a strange kind of archaic freedom in the limits of its modeless interface: to learn more (about the system, about the world, about yourself), write more.

[LEAP FORWARD] + CONTRADICTIONS

for the computer to become an appliance, but not a mere appliance. Its presence must be taken for granted by its user, but in the long run, the act of programming itself must be taken for granted as well." Perhaps in Raskin's words here we can also find one explanation for why the Cat never took hold: the years leading up to the release of the Cat were marked by the quiet dismantling of a culture of programming (with BASIC taught in high schools across North America) that was on its way to being "taken for granted," and in its place, the construction of the user as consumer – as one who merely chooses, clicks, and drags.