Hackers, Computers, and Cooperation: A Critical History of Logo and Constructionist Learning

MORGAN G. AMES, School of Information and Center for Science, Technology, Medicine & Society, University of California, Berkeley, USA

18

This paper examines the history of the learning theory "constructionism" and its most well-known implementation, Logo, to examine beliefs involving both "C's" in CSCW: computers and cooperation. Tracing the tumultuous history of one of the first examples of computer-supported cooperative learning (CSCL) allows us to question some present-day assumptions regarding the universal appeal of learning to program computers that undergirds popular CSCL initiatives today, including the Scratch programming environment and the "FabLab" makerspace movement. Furthermore, teasing out the individualistic and anti-authority threads in this project and its links to present day narratives of technology development exposes the deeply atomized and even oppositional notions of collaboration in these projects and others under the auspices of CSCW today that draw on early notions of 'hacker culture.' These notions tend to favor a limited view of work, learning, and practice—an invisible constraint that continues to inform how we build and evaluate CSCW technologies.

CCS Concepts: • **Human-Centered Computing → Computer supported cooperative work; HCI theory, concepts and models;** • Social and professional topics → History of Computing; Computing education; *Informal Education; Computational thinking;* Computing Education programs.

KEYWORDS

AI; constructionism; education; FabLabs; hackers; ideology; learning; Logo; Maker Movement; Media Lab; MIT; Mindstorms; OLPC; Piaget; robotics; scaffolding; school; Scratch; Seymour Papert; teachers.

ACM Reference format:

Ames, Morgan G. 2018. Hackers, Computers, and Cooperation: A Critical History of Logo and Constructionist Learning. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 2, CSCW, Article 18 (November 2018). ACM, New York, NY. 19 pages. https://doi.org/10.1145/3274287

1 INTRODUCTION

Several hundred conference attendees settled into a large auditorium at The New School for the closing panel of the twelfth annual Interaction Design and Children conference (IDC) on June 27, 2013. The topic was Seymour Papert, professor emeritus at the MIT Media Lab and architect of the learning theory called 'constructionism,' and the tone was radiant. Stanford professor Paulo Blikstein, who had worked with Papert in the early 2000s as a master's student at the MIT Media Lab, opened the panel with a prepared statement. "Can anyone envision a school robotics subculture without Papert? Can we imagine the field of computational literacy without him? Or for that matter, most of technology-enabled project-based learning?" Blikstein asked the audience, clearly expecting an emphatic "no!" in response to each question. But in spite of this influence, Blikstein said, Papert's contribution to the field of education "is largely invisible. It is

Author's address: 543 Stephens Hall, University of California, Berkeley, California, 94720-2350, USA.

2573-0142/2018/November – Art18 $15.00
 https://doi.org/10.1145/3274287

not that educators disagree with Papert's theories or recommendations, they just ignore him entirely."

Papert is certainly not ignored at IDC, a venue attended mostly by designers and technologists who have created technologies for children, often with a learning component to them (e.g. [41,43,106,112]). Nor is he ignored more generally in the technology design world: he is cited in papers presented at IDC, CSCW, CHI, and across a number of other conferences that feature computer-supported cooperative learning (CSCL) projects (e.g. [4,11,31–33,41,52,68,102,103, 107]). Still, even as Papert's ideas continue to crop up in software and his writing continues to be read in classes at the MIT Media Lab and beyond, Blikstein is right: he is indeed less commonly discussed in schools of education. This paper explores what this discrepancy says about how the technology design and development world often understands the potential impacts of computers and the nature of collaboration. In particular, it shows the ideological connections between constructionist learning and MIT's 'hacker' community in the 1970s, from whence it came; it shows how the same ideas still motivate technology development today; and it points toward some of the limitations of those ideas in promoting atomized and oppositional approaches to CSCL.

Trends in CSCL, including those out of the MIT Media Lab, have also been influential in CSCW, especially as CSCW's scope has expanded beyond the workplace to capture family life, children, learning, and play (see e.g. [8,9,13,15,52,54,55,71,105,134]). As a result, CSCW has become a crossroads for discussing these topics and their influence across human-computer interaction (and it is moreover more receptive to social critiques like this one than IDC). This paper thus contributes to this larger discussion: beyond CSCL, there are similar threads of discourse across social computing, the Internet, and technology design and use more broadly.

The lessons we can learn from this case can help us understand these others. As we will see, this foundational CSCL project—one of the first, and certainly one of the best-known—still frames assumptions about the universal allure and importance of learning to program computers, which undergirds projects like Scratch, FabLabs, the Hour of Code, One Laptop per Child, and more. Moreover, these individualistic assumptions about learning and computer use continue to have purchase within CSCW as well as HCI. They promote an atomized and often oppositional understanding of not just learners, but workers and technology 'users,' and make it more difficult to envision alternatives for computer-supported collaboration that account for power, are built around collectivity, or encode forms of social reciprocity. As the consequences of this individualized approach become clear—in antisocial practices online, in stress among family and friends regarding technology use, in ongoing problems with discrimination in the technology industry, in technology-assisted weakening of social institutions like public education, in algorithmic tracking of individuals, and in the technology industry's ongoing inability to imagine collective responses to these issues (e.g. [17])—it becomes useful to examine some of the assumptions that brought us here as a first step in countering them.

2 CONSTRUCTIONISM AND ITS (DIS)CONTENTS

Nearly all of Seymour Papert's work—spanning some forty-odd years of work at MIT, from the 1960s to the 2000s—expresses aspects of his signature contribution, a learning model he called *constructionism.* Using techniques from media archaeology [53], this paper uses close readings of primary and secondary sources to clarify the present state and history of constructionism, a history that belies the ongoing popularity of the learning theory, but explains its fall from grace in education. As justification for so closely examining constructionism's history, we may consider its substantial legacy—a legacy that Blikstein alluded to with only some hyperbole. At the MIT Media Lab, constructionism has been woven into the school's culture from its inception in 1985: his first book *Mindstorms* has long served as a core text of the Media Arts and Sciences graduate course "Technologies for Creative Learning" (later renamed "Learning Creative Learning"), where students discuss their own versions of Papert's 'gears' (explored below) and use them as inspiration for design. The text is also assigned in other design and computer science classes.

Papert's ideas also undergird the block-based programming environment Scratch, developed at the MIT Media Lab in part by Papert's former student Mitch Resnick and collaborator Yasmin Kafai. In the mid-2000s, Scratch experienced a meteoric rise alongside One Laptop per Child (another MIT-based constructionist project that cites Papert as inspiration [10,56]). It is still popular, with over seven million registered users in 2016 [104] and an extensive web archive of Scratch code for 'remixing' and meeting other 'Scratchers' [31,33,102,103]. Scratch inspired the 'Hour of Code' movement, where children are encouraged to spend an hour learning basic coding in a block-based programming environment, and is the primary platform for follow-up content via code.org. It has also been incorporated into the national school curriculum in Great Britain.

While it is difficult to verify Blikstein's claim that Papert originated *all* school robotics programs, he is directly credited with inspiring the "FabLab" structure for makerspaces, which also hails from the MIT Media Lab [57,112,130], and with influencing the philosophy of the Maker Movement more broadly (e.g. [10]). Papert's team collaborated with the LEGO™ Corporation in the 1990s to develop a commercial robotics kit, named after Papert's first book [64]. Echoing tropes from his "Logo" project we will explore below, 'Turtles' that children control with Logo-like commands continue to be a mainstay in commercial children's software, including the "Turtle Academy" website, the "Turtle Graphics" software, the "ComputerCraftEdu" Minecraft mod, the "Turtle! Programming for Kids" app, and the "Move the Turtle" app, as well as board games like "Robot Turtles" that aim to teach computing concepts.

2.1 The Contours of Constructionism

But constructionism's story begins long before these recent projects. Papert first began fleshing out details of this learning theory in the 1970s through a National Science Foundation grant and a series of working memos. His best-selling book *Mindstorms* [81], published in 1980, brought constructionism to a wider audience and remains the core reference on the subject, with other books, articles, working memos, and talk transcripts from the next two decades largely reinforcing its messages. That *Mindstorms* was a best-seller is not too surprising: there is a lot to like in Papert's prose. He draws the reader in with a witty and unabashedly personal style. He offers comforting continuity, even as he risks repetitiveness, by returning to the same small set of themes and parables—and the same overall message—across *Mindstorms* and most of his writing. How would you explain how to make a circle on the ground with your feet? How do you learn how to juggle? How is learning math like learning a language? How is getting to know a new idea like getting to know a person? Why should computers be like pencils? Answers to these questions, Papert argues, hinge on constructionist learning.

Papert's construct**ion**ism borrows heavily from Jean Piaget's theory of construct**iv**ism, as reflected in the confusing similarity between the two names. Before joining MIT in 1964 Papert spent five years at Piaget's "International Centre for Genetic Epistemology" in Geneva [81]. During this time, Piaget's theories of child cognitive development first articulated in the 1920s started to gain popularity worldwide. Papert adopts from Piaget a focus on children's learning as an active process of *constructing* knowledge about the world. Both stress that children (and adults) learn by relating new concepts to what they already know. Piaget calls this "assimilation" and "accommodation," though Papert collapses both into "Piagetian learning" [81].

Papert begins to depart from Piaget in his focus on how this knowledge construction relates to our own bodies and other physical objects around us, which he calls "body knowledge" [81]. Body knowledge, Papert argues, helps learning be something that children can relate to physically or sensorially. He contrasts the possibilities of embodied learning with rote or "dissociated" learning [81], which is removed from children's physical worlds, does not involve play, and does not naturally build on students' existing mental models. Whereas an educational researcher might use Piaget as a lens to explain *all* learning, Papert uses embodiment to define "Piagetian learning" normatively as "natural, spontaneous learning … contrasted with the curriculum-driven learning characteristic of traditional schools" [81]. "I see Piaget as the theorist of learning without curriculum and the theorist of the kind of learning that happens without deliberate teaching," Papert explains [81]. This involves "supporting children as they build their own intellectual structures … planting new constructive elements in it and eliminating noxious ones" [81]. For example, a teacher can embark on a goal that s/he does not know how to accomplish either, and puzzle it out alongside students, creating an empowering, authentic learning experience. "New situations that neither teacher nor learner has seen before come up frequently and so the teacher does not have to pretend not to know," he explains [81].

Papert calls the materials that are particularly useful for embodied learning—which, in his own childhood, were differential gears—"objects-to-think-with" [83]. In the introduction of an edited volume titled *Constructionism,* he describes effective objects-to-think-with as having a "low floor"—easy access for even very young children—and "no ceiling," or near-limitless potential for complexity [87]. These objects-to-think-with should be able to grow with a child, enabling them to think about increasingly complex and abstract ideas in terms of concrete affordances.

While Papert acknowledges that not all children may find differential gears as useful an object-to-think-with as he did, he argues that his programmable robot 'turtle' could be. This turtle can be either a physical robot or (more commonly) a virtual triangle on a screen. It knows just a few simple commands in the programming language 'Logo,' such as go forward, turn right, and pen down (Figure 1). Using these commands, Papert argues, children learn geometry in an embodied way, "playing turtle" to find intuitive ways to express complex mathematical ideas.

Turtles rely on computers, which Papert argues is another powerful object-to-think-with—and one that can easily grow with a child. In the first pages of *Mindstorms* he states,

> The computer is the Proteus of machines. Its essence is its universality, its power to simulate. Because it can take on a thousand forms and can serve a thousand functions, it can appeal to a thousand tastes. [81]

In later writing, Papert argues that programming computers can exert a "holding power" over children equal to videogames, but missing in most teacher-led learning [83]—a holding power that reveals children's natural interest in learning to program. As a result, he advocates giving children unrestricted access to computers so that they can explore them as deeply as they would like.

Papert's gears also lead us to another aspect of constructionism that is distinct from Piaget's constructivism (though in line with Vygotsky and other educational theorists): that of passion. Papert explains that children learning in an embodied way will maintain a love and excitement for learning, and that this *affective* component is another cornerstone of constructionism not present in Piaget's constructivism. "*I fell in love with the gears,*" Papert explained in *Mindstorms*, emphasis his. "This is something that cannot be reduced to purely 'cognitive' terms. Something very personal happened" [81]. Starting at a young age is another factor Papert says is important for establishing a "positive affective tone" for learning. Completely discounting the potential influence his parents and social environment had on his interest, he states,

> First, I remember that no one told me to learn about differential gears. Second, I remember that there was *feeling, love,* as well as understanding in my relationship with gears. Third, I remember that my first encounter with them was in my second year [two years old]. [81]

Echoing educational theorists Dewey, Montessori, Vygotsky, and Piaget, Papert moreover advocates for embracing 'wrong' answers and iteratively revising. He calls this process 'debugging' [81], a term co-opted from programming [58], and explains that an openness to debugging encourages children not to internalize feelings of failure when they have a 'wrong' answer, but to see it as an integral part of the learning process [87]. He connects debugging to learning mathematics, which he argues could be learned and loved by everybody if only our

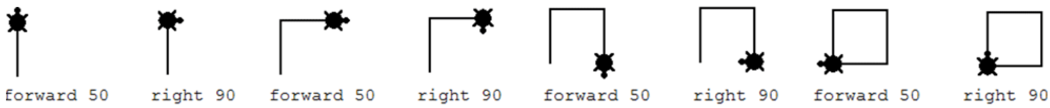

Figure 1. An example of a Logo program, drawing a square (with ‘pen down’ already called).

math-phobic culture provided better tools for assimilating and growing with mathematical ideas rather than blocks for rejecting them such as the common refrain of “I’m just not a math person” [83].

2.1.1 Constructionism’s Cracks

These aspects of constructionism—learning by actively constructing one’s own understanding in an embodied way with objects-to-think-with, all while emphasizing passion and embracing mistakes as part of the learning process—seem straightforward and appealing, even if their essence appears in a number of other learner-centered theories. There is something similarly appealing in the way that Papert blends mathematics and art with his Logo turtles, the enthusiasm he clearly shows for learning, and the profound respect he has for children.

But there are also aspects of his approach that seem more problematic. For instance, one straw man takes form in the otherwise laudable discussion of the value of debugging: the portrayal of school and ‘curriculum’ as inherently evil. Though he explains debugging in detail, Papert does not unpack the complex social processes that can make kids resistant to getting things ‘wrong’; instead, he unilaterally blames school culture for causing the shame they might experience when it happens [81,83]. In *Mindstorms* and even more ardently in his 1993 book *The Children’s Machine*, Papert equates *all* school with the worst kind of disembodied rote learning. Instead of a flawed but aspirational set of institutions that have responded in various ways to a legion of often-conflicting social needs over the decades [76,108,128], ‘School’ in Papert’s writing draws on a common U.S. mythology of ‘school-as-factory’ [7]: a monolith of creativity-squashing drill-and-test, unintuitive facts, and above all “dishonest” “double-talk” [81] which he calls “Instructionism.” Instructionism “rejects the ‘false theories’ of children,” Papert asserts without evidence, “thereby rejecting the way children actually learn.” Teachers, he continues, “distort Piaget’s message by seeing his contribution as revealing that children hold false beliefs, which they, the educators, must overcome. … Children are being force-fed ‘correct’ theories before they are ready to invent them” [81].

Papert’s dislike for the institution of school is so great that he sees nothing redeemable about it. “Computers of the near future will be the private property of individuals, and this will gradually return to the individual the power to determine patterns of education,” he states in *Mindstorms.* “Education will become more of a private act, and people with good ideas … will be able to offer them in an open marketplace directly to consumers” [81]. This and other arguments for privatization (e.g. [83:8–13,86,114]) bear striking similarities to today’s push to de-fund and dismantle public schooling. “Schools as we know them today will have no place in the future,” Papert asserts, and computers in particular will help them “wither away” [81]. With computers, Papert says, the ideals of progressive education can finally be “democratized” [83:14–15]. But how less privileged students might benefit from this privatized education is unaddressed, aside from predictions that computers will be as cheap and ubiquitous as pencils [42,85]—predictions that are still far from being realized (e.g. [8]). Thus, while Papert claims to support diversity [35,114], he does not address how his individualistic approach to learning might enact one of the oldest promises (however unfulfilled it might be) of public education: that of school as a social leveler.

While Papert is unequivocal about his disdain for school, his view of teachers is inconsistent. Throughout his writing he says that children learn “spontaneously” and “without being taught” [81], without curriculum and outside of the classroom, at least as it is currently formulated. At the same time, he provides advice for teachers to incorporate Logo and turtles *into* classroom lessons. He describes adults as hopelessly mired in petty social norms and forever imposing what they have decided are the 'right' ways of thinking on children [81], but then describes how he uses turtles to introduce children to concepts *he* considers important to learn, such as recursion and combinatorics. He even claimed to be doing ethnography as a replacement for other kinds of evaluations [81], but counter to the basic tenets of ethnographic research [19,48,98], his examples fail to reflexively consider the influence of his own presence on learning, instead attributing it to the presence of computers. In short, he implicitly supports and practices *scaffolding*—the work that teachers do to bridge the mental models of learners with new ideas—even while deriding teachers trying to do the same. In a review for *The Children's Machine*, one researcher laments these attacks "because they do not show respect or sympathy for the many teachers who have tried or are now trying to use Logo to implement parts of Seymour Papert's historic vision" [76].

2.1.2 The Missing 'Social'

Papert's disdain for school and his conflicting vision of the role of teachers point to a deeper problem in constructionism: it tends to present learning (especially learning motivation) as largely and implicitly *individualistic* when it is actually a deeply *social* process—and, in particular, that teachers, parents, and other adults play crucial roles. While Papert occasionally discusses the value of learning from *peers*—which Scratch, the Maker Movement, and other informal learning approaches also valorize—he often portrays adults as a *hindrance* to learning and the learning process itself as ultimately individualistic in both practice and motivation. This is illustrated by one of Papert's favorite metaphors for learning mathematics: that of learning a language. The only actors present in his parables of language learning are the learner and the words they are absorbing. Absent is any indication that a strong motivator for learning a language is to be able to communicate needs and desires with other people—absent, in fact, are the other people speaking these words. Also absent is any recognition that as miraculous as children's language learning can seem, it is nonetheless scaffolded by adults who model language and offer definitions and corrections that structure children's rapidly expanding vocabularies. Instead, Papert claims that a child plunked down in France for a year "spontaneously" learns to speak French, rather than being actively (if informally) *taught* it [81]. (This has also been a critique of Papert's mentor Piaget: he first developed his theories based on observations of his own three children, yet did not acknowledge their social influences, including their upper-middle-class Swiss upbringing.) Even when the social inevitably creeps into Papert's narrative—particularly when he discusses 'culture' —the language he uses to describe it is often abstract, even epidemiological. His metaphors of 'germs' and 'seeds' of mathematical learning within a 'landscape' of ideas [81] reduce the social world to a toolbox where children might encounter gadgets that help them learn mathematics and logic, stripping away the complex social motivations and interactions that constitute culture.

The 'social' is also missing in Papert's claim that programmable computers and Logo turtles as "objects-to-think-with" have universal appeal. This rests on the assumption that given unlimited access to a computer with Logo, many children—or at least those whom Papert calls "intellectually interesting"—will use Logo extensively [81,83]. "I have seen hundreds of elementary school children learn very easily to program," he asserts without elaboration in *Mindstorms* [81]—and he further asserts that this programmability is what makes computers so compelling (rather than, say, their novelty, prohibition, status among adults, or other factors). "[A] computer as mathematics-speaking entity puts the learning in a qualitatively new kind of relationship to an important domain of knowledge," he explains. "When a child learns to program, the process of learning is transformed. It becomes more active and self-directed" [81].

The reason that writing small programs in "turtle talk" is much more meaningful to children than existing mathematics curricula, Papert asserts, is because children can "*identify* with the turtle and are thus able to bring their knowledge about their bodies and how they move into the work of learning formal geometry" [81]. But will all children really care about moving a turtle around? Will they really find Logo 'fun' and will making designs with a turtle on a computer really give them a "recognizable personal purpose" any more than current classroom activities [81]? Do children find typing commands for a turtle—much less computer programming in general—really as 'concrete' as throwing a ball [81]? Or could it have been that just *using* a computer was a relative novelty at the time (something that is much less true in our media-rich computing environment today)?

Finally, Papert openly admits that his own childhood provided both inspiration for constructionism and fodder for his descriptions. "I shall in fact concentrate on those ways of thinking that I know best. I begin by looking at what I know about my own development," he says in the first chapter of *Mindstorms* [81]. Some in the technology and design worlds have lauded this, even though it flies in the face of one of the first lessons in human-centered technology design: to 'know your users' rather than making assumptions about them based on your own—possibly quite idiosyncratic—experiences. Throughout his writings, Papert does not critically consider whether his interpretation of his own childhood really can be universalized. He moreover ignores the very different conditions in which different childhoods play out, and even while advocating for 'debugging' he takes technically-inclined childhood creativity and desire to explore a technical world as innate [7,10,78] rather than socially and historically contingent learned behaviors.

2.2 Constructionism in Use: A Brief History of Logo

Thus far we have focused on Papert's writing; how was constructionism implemented in his projects? To find out, we examine the most famous constructionist-based contribution to educational computing by far, the Logo programming language and its "turtle" interface. Logo, developed by Papert in collaboration with Cynthia Solomon and others, took the technology world by storm in the years following Papert's publication of *Mindstorms* in 1980. Alongside the book, Papert's MIT Logo Group, its spin-off corporation Logo Computer Systems Inc., and other companies released versions of Logo for various platforms including the Commodore 64 (sold for $59.95 in 1984), the Atari (for $100), and the Apple II and IIe (for $89.95-$400, depending on features) [80]. These prices were not cheap, and they did not include the cost of the hardware needed to run Logo. As such, despite Papert's rhetoric of reaching all children with Logo, the software was only accessible to institutions, wealthy enthusiasts, and other higher-end consumers.

Papert's timing could not have been better. *Mindstorms* was published and Logo was released just as personal computers started to gain popularity. Alongside them, mass-market computer enthusiast magazines, the main source of information on the new consumer devices, proliferated—and the first wave of social anxiety about teaching all children to program also took hold. Many of these computer enthusiast magazines—*Byte, Family Computing, Compute!, Microcomputing, Antic, RUN, Ahoy, Atari Connection, The Rainbow Magazine, SoftSide*, and more—published positive reviews of *Mindstorms*, interviews with Papert, and descriptions of how to use Logo at home (e.g. [1,18,20,28,30,40,50,51,60,94,97,100,109,118–123,129]). Some of Papert's former students and others in the computing community also published books on Logo. From these influences, Logo gained a large following among computer enthusiasts across the United States.

Logo was also taken up in schools in the early- to mid-1980s, amid a push to teach programming at younger ages and reports that Logo seemed more fun and accessible than its main competitor, BASIC [14,29,47,80,99,113,114]. Despite the ambivalent messages about teachers in *Mindstorms*, teachers around the country enthusiastically brought Logo into their classrooms or computer labs. A few of these programs were sponsored or directly overseen by Papert and his MIT Logo Group, including the Brookline public schools in Boston, Hennigan elementary school in Boston, the Lamplighter School in Dallas, and several New York City public schools [66].

2.2.1 Logo Abroad

Papert also set his sights abroad (for "tribal children in African jungles," as he put it in *Mindstorms* [81]), partnering with fellow MIT professor Nicholas Negroponte and the French government to co-found the World Center for Computation and the Human Resources in Paris. The group installed Logo classrooms in Paris and in Dakar, Senegal in 1982, a plan Negroponte said was "as audacious as … putting men on the moon," though another source said that the two might have joined because of Reagan's deep cuts to research funding in the U.S. [37]. A February 1984 interview with Papert in *Family Computing Magazine* refers to this project when describing Papert as "attempting to cultivate a widespread 'computer culture' especially in Third World societies" [114]. Decades later, the One Laptop per Child project claimed this as a precursor. "In a French government-sponsored pilot project," its *Vision: Progress* page states, "Papert and Negroponte distribute Apple][microcomputers to school children in a suburb of Dakar, Senegal. The experience confirms one of Papert's central assumptions: children in remote, rural, and poor regions of the world take to computers as easily and naturally as children anywhere" [79].

Papert's vision for the project was to bring constructionism to children around the world, but he hoped that this encounter could also help their families overcome more local problems as well. Perhaps "farmers in a small Third World community could use computers to keep better tabs on the availability and distribution of supplies," a 1983 article about the program in the *MIT Technology Review* mused. "Moreover, the very act of 'exploring' with the computer could engender a flexibility of thought that would, in turn, allow the farmers to find novel ways of combining the efficiency of modern agricultural practices with their traditional cultures" [35]. Papert and Negroponte installed computer classrooms overseas several more times, including Costa Rica in 1986 [74] and Cambodia in 1999 [56], before announcing One Laptop per Child in 2005 [7].

2.2.2 Evaluating Logo

According to Papert and others in his group, the results of these Logo-enabled computer classrooms across the U.S. and abroad were nothing short of miraculous. Their working papers and publications on Logo are peppered with anecdotes of formerly average, reticent, or even combative students blossoming with Logo as an object-to-think-with, learning to love the computer and learning (e.g. [81]). However, these did not methodically assess educational outcomes or Papert's claims that Logo helped students "learn how to learn." These were not even accounts that could be independently verified, as details like classrooms and schools were generally not mentioned. Computing magazines that were otherwise excited about Logo at times voiced frustrations from those in the community about this lack of research. John Victor, an educational software developer with a background in educational psychology, complained in the April 1984 issue of *ANTIC Magazine* that the whole computing industry took Papert's word about Logo's power without "a single study showing that Logo is a better teaching language than BASIC." He goes on to outline a few straightforward studies Papert's group could carry out. Instead, he says,

> They just gave Logo to a bunch of kids, and after a while they asked them how they liked it. 'Gee, it's terrific!' That's not scientific. … Logo is used in a lot of schools. They have a big base on which to draw research information, and they're just not doing it. [99:43]

Though Papert's group was not doing systematic assessments of Logo, others were. The education research community was also excited and intrigued by the promises Papert made about Logo, but unlike Papert wanted something more substantial than glowing anecdotes to evaluate the program. Through the 1970s and early 1980s, a group of researchers at Edinburgh University ran classroom experiments with Logo. They found that when they worked directly with a small number of students, they came away with positive stories like Papert's. This observation was formalized by researchers at Kent State University in Ohio, who found that the *meta*-cognitive abilities of nine six-year-olds who used Logo with the researchers for six weeks improved over a small group of nine six-year-olds who used conventional computer-aided instruction [22]. Descriptive studies from other researchers were also modestly positive [136].

However, Tim O'Shea, one of the researchers in Edinburgh, found that these positive results disappeared if the researchers themselves were not present and actively helping children learn. Once the researcher was gone and a teacher took over, in other words, there were no measurable positive outcomes from using Logo over other kinds of learning. "Even imparting the simpler constructs of Logo in more conventional settings turned out to be very demanding," O'Shea concluded [76]. Moreover, he reported, his group and others had been unable to find any consistent evidence of "skill transfer and other general cognitive benefits" that Logo promised [16,76,77,110], a finding corroborated by other Logo researchers [21,23,136]. In fact, some researchers found that *prior* abilities in mathematics or spatial reasoning predicted facility with computer programming—whether with Logo or other languages—rather than programming improving these abilities, pointing the arrow of causality the other way [21,132,136]. As O'Shea later wrote, education researchers "became more skeptical about the extent to which human skills and knowledge could be symbolically represented in the style associated with contemporary MIT work on AI" [76].

Another group of researchers conducted a series of experiments with the New York City public schools that were using Logo, measuring a number of cognitive and meta-cognitive effects in both Logo and non-Logo users. In study after study, they found little or no differences between the two groups [62,89,91–93]. One of the researchers, Roy Pea, took issue with Papert's claims that children can learn Logo without a curriculum (especially when Papert's books prioritized what concepts children should learn and gave examples of how to learn them), and agreed with other researchers that Logo showed potential only when it was well-scaffolded. Pea concluded, "Someone has provided guidance, support, ideas for how [Logo] could be used. Any projected path toward greater competency that another person helps arrange can be thought of as a curriculum," and promising otherwise set educators up for disappointment and the project up for failure [88].

Meanwhile, the Logo classrooms in Paris and Senegal fell apart. "By the end of the Center's first year, Papert had quit," a 1984 *Datamation Magazine* article reported [63,116]. The remnants of the project were "a battlefield, scarred by clashes of management style, personality, and political conviction. It never really recovered. The new French government has done the Center a favor in closing it down." A 1983 article in the *MIT Technology Review* reported that the project was hopelessly utopian and failed to account for issues like widespread illiteracy, especially as Logo depended on written English. It also questioned the project's motives as colonialist:

> Perhaps the most damaging question raised about [the Center] concerns the conviction that helping Third World countries acquire computer technologies would be beneficial. To many critics, such a goal is an artifact of colonialism, imposing Western values and definitions of progress on other cultures for less than altruistic reasons. [35]

As these results percolated through the enthusiast community and as the initial allure of Logo started to wear thin with the struggles of day-to-day use, others stepped forward to voice frustrations. "Five years ago I predicted the demise of BASIC and its eventual displacement by Logo as a programming language for neophytes," one frequent columnist in *Compute! Magazine* wrote in 1986. "As I look back on the past five years, I see that my own vision was clouded by my enthusiasm and that what I saw was largely a dream, not an accurate reflection of the world of educational computing." He described how it was easy to master Logo's basic commands but nearly impossible for new programmers to jump from there to recursion or other more advanced features. Those who made the jump found Logo—an interpreted language, rather than compiled—hopelessly slow and a hog of precious (and very expensive) memory [123].

2.2.3 Papert's Response

Papert's MIT Logo Group, of course, saw these critiques. But instead of adjusting the group's vision or approach to accommodate them, Papert dug in his heels. His books and papers disparaged educational experiments, and the researchers carrying them out. Even in the preface of *Mindstorms*, Papert stated, "If any 'scientific' educational psychologist had tried to 'measure' the effects of this encounter, he would probably have failed. It had profound consequences but, I conjecture, only very many years later. A 'pre- and post-' test at age two would have missed them" [81]. He again stated that educational experiments were worthless in measuring the true effects of Logo in his 1987 essay "Computer Criticism vs. Technocentric Thinking" and said that questions like "does Logo work?" were too "technocentric": so fixated on the technology that they failed to account for the many paths education could take in children's lives [82]. In several subsequent publications he lumped evaluators with their "white coats" in with all that is wrong with School, and his 1987 essay made *ad hominem* attacks on other researchers, including Pea, doing studies on Logo [90].

In dismissing the value of educational research more broadly, these responses sidestepped one of the most important critiques of Logo that it was primarily the researchers' presence and scaffolding that accounted for children's positive experiences with the program. They also avoided addressing Papert's own 'technocentric' thinking in *Mindstorms* in assuming that computers would hold universal appeal for children—an assumption that still undergirds constructionist-inspired projects today. Finally, they inverted critiques that Papert did not account for the importance of scaffolding in learning by accusing other researchers of failing to account for undefined "cultural" factors that made some classrooms successful and others not [82]. Papert entrenched these positions in his books *The Children's Machine: Rethinking School in the Age of the Computer* [83] and *The Connected Family: Bridging the Digital Generation Gap* [84].

Between these critical evaluations and these failures to honestly address them, Papert's direct influence in education started to fade. While some researchers continued to do research on Logo, they incorporated more active instruction than Papert's formulation for constructionist learning specified (e.g. [23,24,44,75,101,133,135,136]). Others in educational research, including Pea and O'Shea, turned away from Logo research entirely. Later studies found that "minimally-guided instruction" and "pure discovery learning"—cornerstones of Papert's claims that children could learn with Logo "without being taught"—did not work more generally [59,69].

3 CONSTRUCTIONISM'S ORIGINS IN HACKER CULTURE

However, Papert's stature in the *technology* world did not fade—it grew. Papert's writings, like Logo's computational "objects-to-think-with," still enchant many who recognize themselves in his descriptions. Even in the face of this evidence that they do not deliver, his ideas have remained captivating to many in technology design. For years technologists, hobbyists, and some enthusiastic teachers convened at Constructionism conferences. Many projects cite him as an influence, including Scratch [102,103], Lego Mindstorms [64], FabLab makerspaces [130], One Laptop per Child [3–5,7,10,56,111,131], and various ed-tech startups and turtle-focused projects. These projects also largely embrace—whether implicitly or explicitly—Papert's views on individualized learning, his disdain for institutionalized education, and his assumptions that computers and other technical devices will naturally captivate children. We can also see the ascendancy of individual 'users' and computers in the technology world more broadly.

So why does constructionism still resonate? The answer involves how those in the technology world view their own school experiences, what kinds of learning they consider most valuable, and their own identities as technologists [7,10]. And this takes us back to 1964, when Papert joined MIT to conduct artificial intelligence (AI) research with AI pioneer Marvin Minsky, and first encountered MIT's nascent 'hacker' culture.

3.1 MIT's Hacker Culture and the Hacker Ethic

Papert's encounter with this hacker group at MIT would set the course for the rest of his life's work. In *The Children's Machine,* Papert explains that one of the main reasons he decided to join Minsky at MIT was a particular group of people in Minsky's lab who obsessed over his mainframe computers. This group had "a wonderful sense of playfulness that I had experienced there on brief visits," Papert explained. "When I finally arrived, all this came together in all-night sessions around a PDP-1 computer that had been given to Minsky. It was pure play" [83:33].

This group has also been described by journalist Steven Levy in his book *Hackers: Heroes of the Computer Revolution*, a book that captured and amplified the zeitgeist of the early hobbyist programming world [65,117]. With a rollicking gonzo-journalistic style, Levy described how this group of men embraced MIT's longstanding culture of elaborate 'hacks' (pranks) in playing with the university's mainframe computers in the 1960s. Extending this terminology, this group started calling themselves 'hackers,' developing a strong idiosyncratic ethos around this identity through the 1960s and 1970s. To them, being a 'hacker' meant being a technical tinkerer driven by an obsession to playfully and deeply engage with computers. They wrote games and other software for fun, shared code, and admired software 'hacks' that were particularly clever. Papert himself appears in Levy's story, albeit only in a parenthetical aside; Papert's collaborator Marvin Minsky, a prominent researcher in the field of artificial intelligence in the 1960s, plays a larger role by providing the hacker group access to his lab computers and sometimes joining their 'hacks.'

Along with a countercultural 'hacker' culture built around hardware tinkering in California, scholars have shown that this early hacker culture at MIT went on to establish the ethos that still undergirds technology development communities today. This ethos spread out of MIT, mingled with distinct but related ethos that developed among other hacker groups [25,26], and ultimately helped to drive the industry's famous utopianism [26,117,126]. Levy codified this ethos as the 'hacker ethic' in his 1984 book, which was in turn embraced by many members of the community as an accurate reflection of their ethos. He split this 'ethic' into six tenets (detailed below), all of which related to either the heady power that unrestricted access to programmable computers and software had for the members of this group, or the group's deep disdain for authority figures and methods for establishing authority other than programming skill.

3.2 Constructionism and the Hacker Ethic

Papert readily admits that the time he spent with these hackers left an indelible mark: he describes his time with them as "playing like a child and experiencing a volcanic explosion of creativity" [83:33]. "I had my first experience of the excitement and the holding power that keeps people working all night with their computers," he explains in *The Children's Machine*. "I realized that children might be able to enjoy the same advantages—a thought that changed my life" [83:13].

In some cases, this influence was direct and specific. For instance, in many of Papert's texts he describes how to make the Logo turtle draw a circle by going forward a little, turning a little, and repeating this many times until one is back where one started. This is one of his primary examples of 'embodied' learning—he argues that to make a circle with one's feet on the floor one could do the same thing [81,83]. But while Papert implicitly takes credit for this, the idea for this algorithm was Minsky's. By Levy's account, it began as a graphical mistake Minsky made on his lab's PDP-1 computer. Minsky, recognizing the underlying elegance of it, dubbed it the 'Circle Algorithm,' and then used it to develop a clever display 'hack' that the group immortalized as the 'Minskytron' [65:3]. Minsky's further experimentation making spirals and other curving designs with the 'Minskytron' bear strong resemblance to many of the designs Papert wrote about making with the Logo turtle. It could have even inspired the turtle itself.

But there were more indirect, but more fundamental, influences of MIT's hacker group on Papert. The ethos of this group bears remarkable resemblance to the ideals of constructionism—

so much so that those in the technology world who identify with this ethos can easily extend this to identify with constructionism as well. The following sections will use the six tenets of the 'hacker ethic' that Levy identified, thematically grouped in twos, to briefly illustrate these parallels. In the process, we will see that the elements unique to constructionism, as opposed to borrowed from Piaget's constructivism or other learning theories, largely owe their existence to MIT's hacker culture—and in turn speak strongly to others who share this ethos.

3.2.1 Access to computers and anything which might teach you something about the way the world works should be unlimited and total. Always yield to the Hands-On Imperative! / All information should be free.

The first two tenets of the 'hacker ethic,' focused on having complete freedom to explore "the way the world works" in a "hands-on" way, dovetail with constructionism's goals of making learning embodied with unlimited access to "objects-to-think-with." They also fit with the "low floor" and "no ceiling" characteristics of the best objects-to-think-with, as well as with the imperative to "debug" one's thinking (along with one's programs)—Papert even borrowed the term from computing culture. Moreover, these tenets' focus on access to *computers* as a means to this hands-on learning echoes Papert's claim that the ultimate object-to-think-with is a computer.

In *Mindstorms*, Papert says, "My vision of a new kind of learning environment demands free contact between children and computers" [81]—children should be able to access all aspects of the computer and have the freedom to dive as deeply as they wish into its inner workings. With unlimited access to a computer, Papert asserts that children can also be empowered to think "like a computer," learning the "language" of the computer. In this way, as Papert often repeated, the "child programs the computer, not the other way around" [81], echoing the feeling of power and freedom that Levy describes hackers feeling in their intense interactions with computers.

Much of early hacker culture at MIT focused on open-source software, a legacy that has lived on in the Free Software Foundation, the Electronic Frontier Foundation, the Wikimedia Foundation, and the many other open-source projects across the technology world. While Logo was *not* initially open-source, Papert said in an interview that he liked open-source software "in principle" but needed money to continue Logo development [114]. Logo was later open-sourced, as were subsequent constructionist projects like One Laptop per Child's 'Sugar' [56] and Scratch [103].

3.2.2 Mistrust authority, promote decentralization. / Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, sex, or position.

In the strongly anti-authority tenets three and four, where hackers "mistrust authority" and judge one another "by their hacking, not bogus criteria," we see the seeds of Papert's deep and at times inexplicable disdain for school as an institution, as well as for many teachers, educational researchers, and other authority figures in children's lives. In *Mindstorms*, Papert describes the classroom as "an artificial and inefficient learning environment that society has been forced to invent because its informal environments fail" [81]. In *The Children's Machine,* Papert has even stronger words. "School has an inherent tendency to infantilize children," he says,

> by placing them in a position of having to do as they are told, to occupy themselves with work dictated by someone else and that, moreover, has no intrinsic value—school-work is done only because the designer of a curriculum decided that doing the work would shape the doer in a desirable form. [83:24]

We saw above in section 2.2.1 that Papert rejected school and was pessimistic about reforming the institution. "Fortunately," Papert counters, "there is a weak link in the vicious cycle": computers [83:37]. Computers would, Papert said in *Mindstorms,*

> … so modify the learning environment outside the classrooms that much if not all the knowledge schools presently try to teach with such pain and expense and such limited success will be learned, as the child learns to talk, painlessly, successfully, and without organized instruction. [81]

Papert later elaborated on this. Using computers for learning "enables us to not put children through that traumatic and dangerous and precarious process of schooling," he said.

> Nothing is more ridiculous than the idea that this technology can be used to *improve* school. It's going to *displace* school and the way we have understood school. ... What's wrong with school is absolutely fundamental. ... School means a place where children are segregated from society and segregated among themselves by age and put through a curriculum. ... I'm saying that it is inconceivable that school as we've known it will continue. Inconceivable. And the reason why it's inconceivable is that little glimmer with my grandson who is used to finding knowledge when he wants to and can get it when he needs it [on a computer], and can get in touch with other people and teachers, not because they are appointed by the state, but because he can contact them in some network somewhere. [86]

There are two elements here that are worth unpacking: first, the intransigent loathsomeness of school, and second, the power of computers as a method for overthrowing the institution. Like Papert, Levy's hackers decried traditional schooling and found great solace in computers. Even in the midst of rampant fears of computer-enabled militarized societies in the 1960s (e.g. [67,126]), computers provided his odd characters a common topic to bond over. Levy describes how it did not matter if you were a professor or "just some guy off the street"—"these young adults who were once outcasts found the computer a fantastic equalizer" [65:2]. Moreover, these 'hackers' described learning about computers not as preparation for being 'suits' in the military-industrial complex (even if the mainframes they used were funded by the U.S. Defense Department), but in terms of passion and freedom, contrasted to the boring, stifling, and unfulfilling classroom. This was an important stance to take, even though not all in the community actually rejected school: to be a student or professor at MIT, many had, in fact, excelled. But the anti-establishment *stance* was what was important here, and it continued to be important in the software development world as well as in American culture, making Papert's anti-school messages still resonant.

These tenets can also help us understand Papert's conflicting view of teachers. Papert concedes that some adults are able to oppose the effects of Instructionism and remain what he called 'yearners,' or independent free-thinkers, instead of turning into 'schoolers' who were hopelessly dependent on the thoughts and opinions of others [83:1]. In these 'yearners' we can recognize MIT's hackers' understanding of themselves as having resisted the confining, stifling effects of 'Instructionism' and remained passionate learners—who were moreover deeply opposed to institutionalized authority. Even when such yearners were "driven out in frustration" [83:37] from schools, as Papert claimed without basis, this resistance found a natural home in computer worlds.

3.2.3 You can create art and beauty on a computer. / Computers can change your life for the better.

The last two tenets of the hacker ethic are where the idealism that hackers feel around computers is most clearly articulated, and where the co-mingling of constructionism and the hacker ethic come together to create a vision for the future. These also connect with the aspects of constructionism that set it apart from other learning theories: its belief that computers have universal appeal, which we also discussed above; its focus on following one's passion in learning; its blend of math and art through drawing with a turtle; and its conviction that through understanding programming concepts one will understand the nature of learning. As we saw above, the alternative to school for Papert was a computer. Like those who helped shape the tenets of the 'hacker ethic,' Papert firmly believed in the power of computers to captivate and enthrall. Part of this captivation was the magic and power of understanding programming concepts enough to command the machine. But unlike Levy's hackers, who seemed to revel in their idiosyncrasy knowing that their passion for computers was *not* shared by most, Papert believed that the computer was the "Proteus of machines": that given unlimited access early enough, *all* children would love it, and that this love would change their lives for the better. In *Mindstorms,* Papert says,

> I am essentially optimistic—some might say utopian—about the effect of computers on society. … I too see the computer presence as a potent influence on the human mind. [81]

Papert's conviction that understanding programming concepts would unlock the keys to understanding one's own learning processes originates in the field into which he was hired at MIT: machine learning. In the 1960s and 1970s, machine learning and MIT were both at the epicenter of the diverse and exciting field of cybernetics [36]. Attracting some of the biggest names in science to wide-ranging conversations, cybernetics was in some ways a reaction to behaviorism, but it also perpetuated some of the utopian thinking about machines that first developed around World War II [38,126,127] and it continued to animate various research initiatives funded by the Advanced Research Projects Agency (or 'ARPA,' with 'Defense' added to the front in 1972)—including Minsky's lab, into which Papert had been hired. Papert himself described cybernetics rapturously in *Mindstorms* as a potent framework for understanding learning by thinking about brains like we think about computers [81:169–171]. This was not unusual: at the time, there was widespread belief that human brains were particularly sophisticated computers; Minsky famously called brains 'meat machines.' While the problems with this equivalence eventually contributed to the collapse of cybernetics and to the 'artificial intelligence winter' during which the field languished, underfunded, for decades, fragments of the belief persisted [6]. Papert's persistent popularity attests to this, as well as the renewed conviction that coding should be a fundamental skill that all children should master—a conviction echoed today by not only Scratch but the Maker Movement, Hour of Code, and other movements in CSCL and beyond.

4 CONCLUSION

"And so it now is up to all of us to carry forth Seymour's legacy—indeed an honorable calling for the IDC community," Paulo Blikstein stated in the description of the 2017 Interaction Design and Children conference, which had the theme "Logo: The Next 50 Years." He continued, "It is the legacy of creating for children environments where each will find the gears of their childhood." Blikstein's message resonates not only with those who experienced the Papert-heavy elements of the MIT Media Lab curriculum; we have seen that the ideals that constructionism was built on are ideals that still motivate the technology world more broadly.

In particular, we have seen a strong alignment between constructionism and the 'hacker ethic,' and the deep convictions that gave rise to this ethos among the hacker community at MIT. Among other effects, this alignment likely meant that even in the face of contradictory evidence, Papert would not revise his theories. To do so would have entailed giving up an ethos that undergirded his professional, if not his personal, identity [7]. These parallels have also helped constructionism keep a strong presence in the technology world, where the 'hacker ethic' still resonates, even when some of its lessons fly in the face of best practices in human-computer interaction. Thus, constructionism is still celebrated at IDC, in design schools, and across CSCW and HCI—a logic further entrenched by decades of technology design centered on individualized 'users.'

This paper has shown that the same forces that shaped constructionism also shape the ways that we tend to view computers and the nature of collaboration—and that this 'hacker ethic' still has purchase even when evidence points the other way. In so doing, it shows the importance of tracing the ideological histories of contemporary technology projects: these linkages take us back to previous waves of utopian thought that may seem quaint, but have often been translated into contemporary discourse with only cosmetic changes. It also offers an antidote to the allure of constructionism specifically, drawing back the curtain and discussing not only what is wrong with the learning theory, but why it continues to be captivating despite this. One contribution here, then, is not merely a critical history of constructionism, but an examination of some of the underlying forces that make some technology projects "charismatic" [4,7] even when they fail to deliver.

Moreover, the ubiquity of these individualized, computer-centric perspectives can make it more difficult to envision alternatives for computer-supported collaboration—ones that encode pro-social behavior, mechanisms for countering power differences, or modes of social reciprocity, for instance—or even decenter the computer as a solution for social problems altogether. While the form of such systems is best left to future work, there are some good starting points for considering alternatives in the growing body of critical work on the technology industry. This scholarship details some of the consequences of the anti-authority, every-person-for-themselves, wild-west-like ethos of early computing communities like MIT's hackers (though their utopianism has also been productive in some respects; see e.g. [7,26,126]). That the very logic of the 'personal computer' was one of individual empowerment rather than ceding agency to the machine [126] might have allowed us to more easily acquiesce to ceding our agency to algorithmic regulation [6,45,70].

For example, social media and other technology companies have upheld computers as a *solution* to social problems (e.g. [27]), even as many have abdicated responsibility for actively cultivating positive social spaces online by framing themselves as 'platforms.' We are increasingly suffering the consequences of this as these spaces are implicated in hate crimes, antisocial behavior, and even damage to democratic institutions and inclusivity [12,39,46,61,73,95,115]. Similarly, a plethora of technology-focused educational companies propose to replace public education with individualized learning, and privately-owned technology-centric charter schools have taken steps toward weakening this public institution and removing pathways of public accountability in education. Alongside this, middle-class parents fret about 'screen time' in a world where technology use is seen as antithetical to family togetherness and pro-social behavior (e.g. [2,9]). Even the industry's tepid and individualistic reaction to institutionalized racism and sexism—where women are encouraged to simply "lean in" and hateful behavior is framed as "a few bad apples" or even "a problem of both sides"—indicates the depth of this ideological frame.

A focus on transparency as a possible solution for some of these social problems within the technology world ignores the problematic interaction between transparency and power (e.g. [12,124]). This 'solution' is moreover individualistic and renders us all less receptive to the potentials for collective action to foment change—something toward which technology cultures within Silicon Valley in particular have long been hostile [17]. Still, I find hope in the increasing strength of social movements built on principles of collective action (including Occupy Wall Street, Black Lives Matter, Stand With Standing Rock, and Me Too/Time's Up) which lean on technology platforms to amplify their messages—even if these platforms fail to protect them [125].

While these broader trends are not necessarily traceable directly to Papert and constructionism, they are all influenced by the same ideological forebears that made the values of early hacker culture so resonant, and could all benefit from some "troubling" [49] in order to plumb the smooth veneer of their utopian mythologies [4,7,34,72] and understand better how they play out in "the mangle of practice" [96], both historically and contemporaneously. These ideologies tend to favor limited, and limiting, perspectives on not only learning, but computer use more generally, and this constraint continues to inform how we build and evaluate technologies for collaboration and more. While constructionism's 'rugged individualism,' often defined outside or even in opposition to institutions and social relations, is not unique to either the learning theory or to hacker culture, both do valorize it. Indeed, it reflects a primacy of individual over collective experiences that has defined the American cultural landscape for at least a century [7,10]. When American norms have played an outsized role in informing technology design, this atomization becomes at least legible (if not actually taken up) across the world—a dominance worth questioning. This paper examines some of the ideological foundations that brought us here as a first step in countering them.

ACKNOWLEDGMENTS

Generous feedback from colleagues including Daniela Rosner, Joseph Klett, Matt Rafalow, Alicia Blum-Ross, Amber Levinson, Antero Garcia, Laura Portwood-Stacer, Jenna Burrell, Benjamin “Mako” Hill, and CSCW reviewers have been instrumental in helping me strengthen this paper. I am also indebted to Nabeel Siddiqui for providing scans of articles that mentioned Papert and Logo in 1980s popular computing magazines.

REFERENCES

[1] Christine Adamec. 1984. Computer Tutor. *RUN Magazine*, 34–36.

[2] Morgan G. Ames. 2013. Managing mobile multitasking: the culture of iPhones on stanford campus. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work and Social Computing*. 1487–1498. https://doi.org/10.1145/2441776.2441945

[3] Morgan G. Ames. 2014. Translating Magic: The Charisma of One Laptop Per Child’s XO Laptop in Paraguay. In *Beyond Imported Magic: Essays on Science, Technology, and Society in Latin America*, Eden Medina, Ivan da Costa Marques and Christina Holmes (eds.). MIT Press, 207–224.

[4] Morgan G. Ames. 2015. Charismatic Technology. *Proceedings of the 5th Decennial AARHUS Conference*: 109–120. https://doi.org/10.1080/19447014508661941

[5] Morgan G. Ames. 2016. Learning consumption: Media, literacy, and the legacy of One Laptop per Child. *The Information Society* 32, 2: 85–97. https://doi.org/10.1080/01972243.2016.1130497

[6] Morgan G. Ames. 2018. Deconstructing the algorithmic sublime. June: 1–4. https://doi.org/10.1177/2053951718779194

[7] Morgan G. Ames. 2019. *The Charisma Machine: The Life, Death, and Legacy of One Laptop per Child*. MIT Press.

[8] Morgan G. Ames and Jenna Burrell. 2017. “Connected learning” and the equity agenda: A microsociology of Minecraft play. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work and Social Computing*. https://doi.org/10.1145/2998181.2998318

[9] Morgan G. Ames, Janet Go, Joseph’Jofish’ Kaye, and Mirjana Spasojevic. 2011. Understanding technology choices and values through social class. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work and Social Computing*. 55–64.

[10] Morgan G. Ames and Daniela K. Rosner. 2014. From Drills to Laptops: Designing Modern Childhood Imaginaries. *Information, Communication & Society* 17, 3: 357–370. https://doi.org/10.1080/1369118X.2013.873067

[11] Morgan G. Ames, Daniela K. Rosner, and Ingrid Erickson. 2015. Worship, Faith, and Evangelism. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 69–81. https://doi.org/10.1145/2675133.2675282

[12] Mike Ananny and Kate Crawford. 2016. Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*: 146144481667664. https://doi.org/10.1177/1461444816676645

[13] Marek Bell, Matthew Chalmers, Louise Barkhuus, Malcolm Hall, Scott Sherwood, Paul Tennent, Barry Brown, Duncan Rowland, and Steve Benford. 2006. Interweaving mobile games with everyday life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/1124772.1124835

[14] John Blackford. 1983. Computers in School: New Approaches. *Compute! Magazine*, 42–48.

[15] Lindsay Blackwell, Emma Gardiner, and Sarita Schoenebeck. 2016. Managing Expectations: Technology Tensions among Parents and Teens. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work and Social Computing*, 1388–1399. https://doi.org/10.1145/2818048.2819928

[16] J. B. H. Du Boulay and Tim O’Shea. 1976. *How to work the Logo Machine*.

[17] David E Broockman, Greg F Ferenstein, and Neil A Malhotra. 2017. Wealthy Elites’ Policy Preferences and Economic Inequality: The Case of Technology Entrepreneurs.

[18] Chris Brown. 1981. Harvard Micro Conference: Education for Publishers and Teachers. *80 Microcomputing*, 54–56.

[19] Michael Burawoy. 1998. The Extended Case Method. *Sociological Theory* 16, 1: 4–33. https://doi.org/10.1111/0735-2751.00040

[20] Bill Camarda. 1984. Computers could save our public schools, says education expert kenneth komoski. *Family Computing*, 54–56.

[21] Douglas Clements. 1985. Research on Logo in Education: Is the Turtle Slow but Steady, or Not Even in the Race? *Computers in the Schools* 2, 2–3: 55–71. https://doi.org/10.1300/J025v02n02_07

[22] Douglas H. Clements and Dominic F. Gullo. 1984. Effects of computer programming on young children’s cognition. *Journal of Educational Psychology* 76, 6: 1051–1058. https://doi.org/10.1037/0022-0663.76.6.1051

[23] Douglas H. Clements and Julie Sarama. 1997. Research on Logo. *Computers in the Schools* 14, 1–2: 9–46. https://doi.org/10.1300/J025v14n01_02

[24] Douglas H Clements and Bonnie K Nastasi. 1999. Metacognition, learning, and educational computer environments. *Information Technology in Childhood Education Annual*: 3–36.

[25] E. Gabriella Coleman and Alex Golub. 2008. Hacker practice. *Anthropological Theory* 8, 3: 255–277. https://doi.org/10.1177/1463499608093814

[26] Gabriella Coleman. 2013. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton University Press.

[27] Josh Constine. 2017. Facebook changes mission statement to "bring the world closer together." *TechCrunch*.
[28] Fred D'Ignazio. 1985. "Easy-Play" Computer Peripherals For The Family. *Compute!'s Gazette*.
[29] Fred D'Ignazio. 1985. Is Programming Dead? *Compute!'s Gazette*, 128.
[30] Fred D'Ignazio. 1986. Vancouver 1986: Distance Is A State Of Mind. *Compute!'s Gazette*, 83.
[31] Sayamindu Dasgupta, William Hale, Andrés Monroy-Hernández, and Benjamin Mako Hill. 2016. Remixing as a Pathway to Computational Thinking. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work and Social Computing*. 1436–1447. https://doi.org/10.1145/2818048.2819984
[32] Sayamindu Dasgupta and Benjamin Mako Hill. 2017. Scratch Community Blocks. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*: 3620–3631. https://doi.org/10.1145/3025453.3025847
[33] Mark Dorling and Dave White. 2015. Scratch: A Way to Logo and Python. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*: 191–196. https://doi.org/10.1145/2676723.2677256
[34] Paul Dourish and Genevieve Bell. 2011. *Divining a Digital Future*. MIT Press.
[35] James Dray, Joseph Menosky. 1983. Computers and a New World Order. *MIT Technology Review* 13, May/June: 12–16.
[36] Jean-Pierre Dupuy. 2009. On the Origins of Cognitive Science: The Mechanization of the Mind. MIT Press.
[37] William Echikson. 1982. Microcomputer Center in Paris. *Christian Science Monitor*: 1–4.
[38] Paul N. Edwards. 1996. The Closed World: Computers and the Politics of Discourse in Cold War America. MIT Press.
[39] Gayle Falkenthal. 2011. Internet trolls, anonymity and The First Amendment. *Washington Times*.
[40] Peter J. Favaro. Taught to the Tune of A. *SoftSide*, 14–20.
[41] Deborah A. Fields, Katarina Pantic, and Yasmin B. Kafai. 2015. "I have a tutorial for this": The language of online peer support in the Scratch programming community. *Proceedings of the 14th International Conference on Interaction Design and Children - IDC '15*, November 2016: 229–238. https://doi.org/10.1145/2771839.2771863
[42] Edward B. Fiske and Seymour Papert. 1985. Education Interview: Seymour Papert on Computers. *The New York Times*.
[43] Louise P. Flannery, Brian Silverman, Elizabeth R. Kazakoff, Marina Umaschi Bers, Paula Bontá, and Mitchel Resnick. 2013. Designing ScratchJr. In *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13*, 1–10. https://doi.org/10.1145/2485760.2485785
[44] Catherine Wilson Gillespie and Sally Beisser. 2001. Developmentally appropriate LOGO computer programming with young children. *Information Technology in Childhood Education* 12: 229–245.
[45] Tarleton Gillespie. 2012. The Relevance of Algorithms.
[46] Tarleton Gillespie. 2013. The Politics of "Platforms." *A Companion to New Media Dynamics*: 407–416. https://doi.org/10.1002/9781118321607.ch28
[47] Carlos Vidal Greth. 1983. Learning to Learn with the Father of LOGO. *Atari Connection*, 23–36.
[48] Donna Haraway. 1988. Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective. *Feminist Studies* 14, 3: 575. https://doi.org/10.2307/3178066
[49] Donna Haraway. 2016. Staying with the Trouble: Making Kin in the Chthulucene. Duke University Press.
[50] John and Mary Harrison. 1983. Computers in Education: Benefit or Bombshell? *ANTIC, The ATARI Resource*, 33–36.
[51] Richard Herring. Educational software: a guide for parents (part IV). *Ahoy! Magazine*, 63–67.
[52] Juan Pablo Hourcade, Daiana Beitler, Fernando Cormenzana, and Pablo Flores. 2007. Reflections on a Pilot OLPC Experience in Uruguay. *CHI 2008 Workshop on HCI for Community and International Development*: 2.
[53] Erkki Huhtamo and Jussi Parikka. 2011. *Media Archaeology*. University of California Press.
[54] Hilary Hutchinson, Heiko Hansen, Nicolas Roussel, Björn Eiderbäck, Wendy Mackay, Bosse Westerlund, Benjamin B Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, and Helen Evans. 2003. Technology probes. In *Proceedings of the Conference on Human Factors in Computing Systems*, 17–24. https://doi.org/10.1145/642611.642616
[55] Mizuko Ito et al. 2008. Living and Learning with New Media: Summary of Findings from the Digital Youth Project.
[56] Charles Kane, Walter Bender, Jody Cornish, and Neal Donahue. 2012. *Learning to Change the World: The Social Impact of One Laptop Per Child*. Palgrave Macmillan.
[57] Eva-Sophie Katterfeldt, Nadine Dittert, and Heidi Schelhowe. 2015. Designing digital fabrication learning environments for Bildung: Implications from ten years of physical computing workshops. *International Journal of Child-Computer Interaction* 5: 3–10. https://doi.org/10.1016/j.ijcci.2015.08.001
[58] Peggy Aldrich Kidwell. 1998. Stalking the Elusive Computer Bug. *Ieee Annals Of The History Of Computing* 20, 4: 5–9.
[59] Paul A. Kirschner, John Sweller, and Richard E. Clark. 2006. Why Minimal Guidance During Instruction Does Not Work. *Educational Psychologist* 41, March 2015: 87–98. https://doi.org/10.1207/s15326985ep4102
[60] Glenn Kleiman. 1982. Teaching Johnny To Program. *Compute! Magazine*, 88–91.
[61] Panos Kompatsiaris and Yiannis Mylonas. 2014. The rise of nazism and the web: Social media as platforms of racist discourses in the context of the Greek Economic crisis. *Social Media, Politics and the State: Protests, Revolutions, Riots, Crime and Policing in the Age of Facebook, Twitter and YouTube*: 109–130. https://doi.org/10.4324/9781315764832
[62] D. Midian Kurland and Roy D. Pea. 1985. Children's Mental Models of Recursive Logo Programs. *Journal of Educational Computing Research* 1, 2: 235–243. https://doi.org/10.2190/JV9Y-5PD0-MX22-9J4Y
[63] Robert W. Lawler and Masoud Yazdani. 1987. Artificial Intelligence and Education, Volume One: Learning Environments and Tutoring Systems. Intellect Ltd.
[64] LEGO staff. 2017. LEGO Mindstorms History. *lego.com*. Retrieved from https://www.lego.com/en-us/mindstorms/history
[65] Steven Levy. 1984. Hackers — Heroes of the computer revolution. Dell Publishing.
[66] Logo Foundation staff. What is LOGO? History. *The MIT Media Lab website*. Retrieved from http://el.media.mit.edu/logo-foundation/what_is_logo/history.html

[67] Steven Lubar. 1992. "Do Not Fold, Spindle or Mutilate": A Cultural History of the Punch Card. *The Journal of American Culture* 15, 4: 43–55. https://doi.org/10.1111/j.1542-734X.1992.1504_43.x

[68] J. Nathan Matias, Sayamindu Dasgupta, and Benjamin Mako Hill. 2016. Skill Progression in Scratch Revisited. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*: 1486–1490. https://doi.org/10.1145/2858036.2858349

[69] Richard E. Mayer. 2004. Should There Be a Three-Strikes Rule Against Pure Discovery Learning? *American Psychologist* 59, 1: 14–19. https://doi.org/10.1037/0003-066X.59.1.14

[70] Brent Daniel Mittelstadt, Patrick Allo, Mariarosaria Taddeo, Sandra Wachter, and Luciano Floridi. 2016. The ethics of algorithms: Mapping the debate. *Big Data & Society* 3, 2: 205395171667967. https://doi.org/10.1177/2053951716679679

[71] Wendy Moncur. 2007. Providing affective information to family and friends based on social networks. In *CHI '07 extended abstracts on Human factors in computing systems*. https://doi.org/10.1145/1240866.1240983

[72] Vincent Mosco. 2005. The Digital Sublime: Myth, Power, and Cyberspace. The MIT Press.

[73] Angela Nagle. 2017. *Kill All Normies*. Zero Books.

[74] Nicholas Negroponte. 2006. The $100 Laptop. In *Globalization and Education*, Marcelo Sánchez Sorondo, Edmond Malinvaud and Pierre Léna (eds.). Walter de Gruyter, 19–23.

[75] Richard Noss and Celia Hoyles. 1996. *Windows on Mathematical Meanings*. Kluwer Academic Publishers.

[76] Tim O'Shea and Timothy Koschmann. 1997. Review, The Children's Machine: Rethinking School in the Age of the Computer (book). *Journal of Learning Sciences* 6, 4: 401–415.

[77] Tim O'Shea and John Self. 1983. *Learning and teaching with computers*. Harvester.

[78] Amy F. Ogata. 2013. *Designing the Creative Child*. University of Minnesota Press.

[79] OLPC staff. 2012. OLPC's Vision. *OLPC Website*.

[80] Mindy Pantiel and Becky Petersen. 1984. Learning Logo is a Family Affair. *Family Computing*, 64–69.

[81] Seymour Papert. 1980. Mindstorms: Children, Computers, and Powerful Ideas. Basic Books.

[82] Seymour Papert. 1987. Computer Criticism vs. Technocentric Thinking. *Educational Researcher* 16, 1: 22–30. https://doi.org/10.2307/1174251

[83] Seymour Papert. 1993. The Children's Machine: Rethinking school in the age of the computer. Basic Books.

[84] Seymour Papert. 1996. The Connected Family: Bridging the digital generation gap. Longstreet Press.

[85] Seymour Papert. 1996. Computers in the Classroom: Agents of Change. *The Washington Post Education Review*.

[86] Seymour Papert and Paulo Freire. The Future of School. Retrieved from http://www.papert.org/articles/freire/freirePart1.html

[87] Seymour Papert and Idit Harel. 1991. Situating Constructionism. *Constructionism*: 1–11. https://doi.org/10.1111/1467-9752.00269

[88] Roy D. Pea. 1984. Symbolic Systems and Thinking Skills: Logo in Context. *Proceedings of LOGO '84, MIT, Cambridge*.

[89] Roy D. Pea. 1987. Cognitive technologies for mathematics education. *Cognitive science and math education*: 89–122.

[90] Roy D. Pea. 1987. The Aims of Software Criticism: Reply to Professor Papert. *Educational Researcher* 16, 5: 4–8. https://doi.org/10.3102/0013189X016005004

[91] Roy D. Pea and D. Midian Kurland. 1984. *Logo Programming and the Development of Planning Skills*. New York, NY.

[92] Roy D. Pea and D. Midian Kurland. 1984. On the cognitive effects of learning computer programming. *New Ideas in Psychology* 2, 2: 137–168. https://doi.org/10.1016/0732-118X(84)90018-7

[93] Roy D. Pea, D. Midian Kurland, and Jan Hawkins. 1985. LOGO and the Development of Thinking Skills. In *Children and Microcomputers: Research on the Newest Medium*, M. Chen and W. Paisley (eds.). Sage, 193–212.

[94] Pamela Petrakos. 1981. Education reviews: Mindstorms. *80 Microcomputing*, 14.

[95] Whitney Phillips. 2016. This Is Why We Can't Have Nice Things: Mapping the Relationship between Online Trolling and Mainstream Culture. MIT Press.

[96] Andrew Pickering. 1993. The Mangle of Practice: Agency and Emergence in the Sociology of Science. *American Journal of Sociology* 99, 3: 559–589.

[97] Michael Plog. 1987. Programming the LOGO Turtle: Studies in Learning Transfer. *Rainbow*, 86–87.

[98] Paul Rabinow. 1977. *Reflections of Fieldwork in Morocco*. University of California Press.

[99] Christopher Rauber. 1984. John Victor, Computer Educator. *ANTIC, The ATARI Resource*.

[100] Sally Reed. 1983. Preschool Computing: What's Too Young? *Family Computing*, 55–59.

[101] Lillian Kay Reeder and James S. Leming. 1994. The effect of logo on the nonverbal reasoning ability of rural and disadvantaged third graders. *Journal of Research on Computing in Education* 26, 4: 558–564. https://doi.org/10.1080/08886504.1994.10782111

[102] Mitchel Resnick. 2012. Explore the Next Generation of Scratch. *Constructionism 2012: Theory Practice and Impact*.

[103] Mitchel Resnick, Brian Silverman, Yasmin B. Kafai, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, and Jay Silver. 2009. Scratch: Programming for everyone. *Communications of the ACM* 52, 11: 60. https://doi.org/10.1145/1592761.1592779

[104] Gabriela T. Richard and Yasmin B. Kafai. 2016. Blind Spots in Youth DIY Programming. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, May: 1473–1485. https://doi.org/10.1145/2858036.2858590

[105] Kathryn E Ringland, Christine T Wolf, Lynn Dombrowski, and Gillian R. Hayes. 2015. Making "Safe." In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15*, 1788–1800. https://doi.org/10.1145/2675133.2675216

[106] Ricarose Roque, Yasmin Kafai, and Deborah Fields. 2012. From tools to communities. *Proceedings of the 11th International Conference on Interaction Design and Children - IDC '12*: 220. https://doi.org/10.1145/2307096.2307130

[107] Daniela K. Rosner and Morgan Ames. 2014. Designing for repair? In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW '14*, 319–331. https://doi.org/10.1145/2531602.2531692
[108] John Rury. 2005. *Education and Social Change*. Lawrence Erlbaum Associates, Mahwah, N.J.
[109] Fran Saito and Bob Albrecht. 1984. Watching Important Program Evaluators - The Children. *The Rainbow*, 60–65.
[110] John Self. 1985. Microcomputers in Education: A Critical Evaluation of Educational Software. Harvester.
[111] Janaki Srinivasan, Megan Finn, and Morgan G. Ames. 2017. Information determinism: The consequences of the faith in information. *Information Society* 33, 1: 13–22. https://doi.org/10.1080/01972243.2016.1248613
[112] Gary S. Stager. 2013. Papert's Prison Fab Lab: Implications for the maker movement and education design. *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13*: 487–490. https://doi.org/10.1145/2485760.2485811
[113] David Stanton. Logo Does Much More than Just Teach - It Turns Children into Teachers who Learn for Themselves. *Compute! Magazine*, 84.
[114] Nick Sullivan. 1984. The Man Behind Logo. *Family Computing*, 70–71.
[115] Nabiha Syed. 2017. Real Talk About Fake News: Towards a Better Theory for Platform Governance. *The Yale Law Journal Forum*: 337–357.
[116] Paul Tate. 1984. The Blossoming of European AI. *Datamation, International Edition*, 85–88.
[117] Douglas Thomas. 2002. *Hacker Culture*.
[118] David D. Thornburg. 1981. Computers and Society. *Compute! Magazine*, 14.
[119] David D. Thornburg. 1982. Stranger Than Fiction ... *Compute! Magazine*, 14–16.
[120] David D. Thornburg. 1982. Friends of the Turtle: LOGO Is Not Just Child's Play. *Compute! Magazine*, 100–107.
[121] David D. Thornburg. 1984. That's not a game, that's a microworld ... *Compute! Magazine*, 154–156.
[122] David D. Thornburg. 1984. Computers and Society. *Compute! Magazine*, 132–133.
[123] David D. Thornburg. 1986. Whatever Happened to Logo? *Compute! Magazine*, 83.
[124] Scott Titsworth. 2018. "Transparency" Is the Mother of Fake News. *The New York Times*.
[125] Zeynep Tufekci. *Twitter and Tear Gas*. Yale University Press.
[126] Fred Turner. 2006. *From Counterculture to Cyberculture*. University of Chicago Press, Chicago.
[127] Fred Turner. 2014. *The Democratic Surround*. University of Chicago Press.
[128] David Tyack and Larry Cuban. 1995. *Tinkering toward Utopia*. Harvard University Press.
[129] G Michael Vose. 1981. Artificial Intelligence at M.I.T. *Microcomputing*, 118.
[130] Julia Walter-Herrmann and Corinne Büching. 2014. *FabLab: Of Machines, Makers and Inventors*. Transcript-Verlag.
[131] Mark Warschauer and Morgan G. Ames. 2010. Can One Laptop per Child Save the World's Poor? *Journal of International Affairs* 64, 1: 33–51.
[132] N. M. Webb, P. Ender, and S. Lewis. 1986. Problem-Solving Strategies and Group Processes in Small Groups Learning Computer Programming. *American Educational Research Journal* 23, 2: 243–261. https://doi.org/10.3102/00028312023002243
[133] Catherine Gillespie. 2004. Seymour Papert's Vision for Early Childhood Education? A Descriptive Study of Head Start and Kindergarten Students in Discovery-based, Logo-rich Classrooms. *Early Childhood Research & Practice* 6, 1: 1–19.
[134] Sarita Yardi and Amy Bruckman. 2012. Income, race, and class. In *Proceedings of the 2012 ACM annual Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2207676.2208716
[135] N. J. Yelland. 1994. A Case Study of Six Children Learning with Logo. *Gender and Education* 6, 1: 19–34. https://doi.org/10.1080/0954025940060102
[136] Nicola Yelland. 1995. Mindstorms or a storm in a teacup? A review of research with Logo. *International Journal of Mathematical Education in Science and Technology* 26, 6: 853–869. https://doi.org/10.1080/0020739950260607

Received April 2018; revised July 2018; accepted September 2018.