CONVERGENCE

Convergence: The Journal of Research into New Media Technologies

Summer 2003, Volume 9, Number 2

Special Issue on New Media Technologies for Learning

Guest edited by Amy Bruckman

Published by the
University of Luton Press

UNIVERSITY OF LUTON
p
press

Contents

Editorial by Amy Bruckman 5

Debates

Seymour Papert
What Are You Talking About? A Dialogue with Seymour Papert 10

Cathleen Norris and Elliot Soloway
A School Computing Infrastructure by 2008 13

Sherry Turkle
From Powerful Ideas to PowerPoint 19

Articles

Michael Eisenberg
Mindstuff: Educational Technology Beyond the Computer 29

Marina Umaschi Bers
Kaleidostories: Sharing Stories Across the World in a Constructionist Virtual Community for Learning 54

Diana Joseph and Deñise C. Nacu
Designing Interesting Learning Environments When the Medium Isn't Enough 84

Reviews

Michelle M. Kazmer
Andrew Zucker, Robert Kozma, Camille Marder and Louise Yarnall, *The Virtual High School: Teaching Generation V* 118

Melanie Misanchuk and Sasha Barab
K. Ann Renniger and Wesley Shumar (eds), *Building Virtual Communities: Learning and Change in Cyberspace* 120

Notes on Contributors **127**

From Powerful Ideas to PowerPoint

Sherry Turkle

When computer literacy was almost synonymous with programming, the programming-in-education advocates were divided into two camps: those who supported BASIC as the language of choice and those who supported the Logo language. BASIC, developed at Dartmouth and with a long history of use in educational settings, had a head start, a strong 'installed base' of users, master teachers, and teaching materials. Logo, developed in Cambridge-based research centres and the Massachusetts Institute of Technology, had been built from the start with child users in mind. It presented itself as the language that could best use programming to transmit programming's most powerful ideas.

Logo emphasised procedural thinking; its proponents shunned BASIC's GOTO command which enabled the programmer to reach line 23 of a program and have the program go to line 6, then perhaps 10 and then 8 at which point the program could return to line 24. The proponents of BASIC insisted that you could teach elegant as well as 'spaghetti' BASIC and they argued that the Logo camp was confusing the issue about the importance of programming in the curriculum because it was pushing an arcane language instead of a widespread and accepted one. It was taking computers in education out of the mainstream and into the exotic. In Logo's defence, Seymour Papert of MIT, argued that the QWERTY keyboard, universal to all American typewriters (and now computers), had been designed to hold back the speed of typing because the early typewriters would get their metal tines entangled if the keys were pressed in too rapid succession. The exotic keyboard might be just what our new computers needed.

Earlier this year, I spoke with Dr. Arthur Luehrmann, one of the developers of BASIC at Dartmouth and one of the most articulate defenders of BASIC as an educational tool. Dr. Luehrmann and I discussed the use of simulation games and PowerPoint in American classrooms and he commented that: 'While the Logo people were fighting with the BASIC people, we missed the real story that was unfolding. We all should have been fighting with the Microsoft people. The computer was closing down. Keeping it open was the real story.'

* * *

In the 1980s, as personal computers moved into schools, there was strong disagreement about what constituted computer literacy. One important issue was whether children should learn to program. Some of those who opposed this idea claimed that software would soon be 'black boxed', making programming skills unnecessary for the effective use of the computer as a tool. But for many advocates of making programming part of the standard curriculum, the case did not rest on instrumental necessity. They challenged the view of the computer as 'just a tool'. The computer was a carrier of culture and programming an element of a new cultural fluency. In this view, one that I shared based on my own work as an ethnographer and clinician, programming could change children's ways of thinking, including their ways of thinking about themselves.[1]

For some advocates of computers in education, programming encouraged children to think like epistemologists and psychologists because it raised questions about procedural thinking and encouraged reflection on one's own style of learning.[2] And some believed that access to the 'innards' of the computer had a positive political valence.[3] Universal access to programming skills seemed like a way to attack the digital divide not just in terms of who owned or used computers but who knew how to control them. In my own research I found that for some people, understanding how a computer worked supported the belief that you could understand how other things worked as well – in the social as well as in the technical world. The transparent understanding of a computer could become a metaphor for political empowerment.

Today, these images of what programming could mean in educational and political culture seem distant, almost exotic. These days programming is no longer much taught in standard classrooms, relegated for the most part to special after-school computer clubs. These days, educators most often think of computer literacy as the ability to use the computer as an information appliance for such purposes as word processing, running simulations, accessing educational CD-ROMs navigating the internet, and using presentation software such as PowerPoint. Together these applications constitute an aesthetic for educational computing. What dominates is simulation and presentation as its own powerful idea.

From transparency to opacity: simulation and its discontents

In 1980, most computer users who spoke of 'transparency' were referring to a transparency analogous to that of traditional machines, an ability to 'open the hood' and 'poke around', at least metaphorically speaking. But with the introduction of the Macintosh in 1984, when computer users spoke about transparency, they were referring to seeing their documents and programs represented by attractive and easy-to-interpret icons. They were referring to an ability to make things work without needing to go below the screen surface. Today, the word 'transparency' has taken on its 'Macintosh meaning' in both computer talk and colloquial language. In a culture of simulation, when people say that something is transparent, they mean that they can see how to make it work, not that they know how it works. Users are presented with a scintillating surface on which to float, skim, and play. There is nowhere visible to dive. The evolving computer (and how it has been presented to children) has served as a carrier object for the idea that a search for depth and mechanism is not the way to understand, that it is more fruitful to explore the world of shifting surfaces than to embark on a search for origins and structure. It is the difference between aspiring to make your own video game where you decide and embody the rules and becoming a master player of someone else's game.

The computing that children are most immersed in has moved from programming and the aesthetic of the algorithm to software that socialises users into the culture of simulation. For example, in *SimCity* you engage in civil engineering and urban policy planning, in *SimLife* you design ecosystems and the organisms to inhabit to them, in *The Sims* you create a family and attempt to steer its members toward social and financial well being. The goal in each case is to make a successful whole from complex, interrelated parts. In no case, does the user design or modify the algorithms that underlie the game. Success at the games comes rather from developing an understanding, through trial and error, of algorithms designed by others.

'Your orgot is being eaten up', says the message on the screen. It is a rainy Sunday afternoon and I am with Tim, 13. We are playing *SimLife*, Tim's favourite computer game. Tim says that he likes *SimLife* because 'even though it's not a video game, you can play it like one'. By this he means that as in a video game, he is able to act on a vague intuitive sense of what will work even when he doesn't have a model of the rules underneath the game's behaviour. ('My trilobites went extinct. They must have run out of algae. I didn't give them algae. I forgot. I think I'll do that now.') For example, when his sea urchins become extinct, I ask him why:

> Tim: I don't know, it's just something that happens.
> ST: Do you know how to find out why it happened?
> Tim: No.
> ST: Do you mind that you can't tell why?
> Tim: No. I don't let things like that bother me. It's not what's important.

'Your orgot is being eaten up', says the message on the screen. 'What's an orgot?' I ask Tim. He doesn't know. 'I just ignore that', he says confidently. 'You don't need to know that kind of stuff to play'. I suppose I look unhappy, haunted by a lifetime habit of not proceeding to step two before I understand step one, because Tim tries to appease me by coming up with a working definition of orgot: 'I think it is sort of like an organism. I never read that, but just from playing, I would say that's what it is.'

A few minutes later the game informs us: 'Your fig orgot moved to another species.' I say nothing, but Tim reads my mind and shows compassion: 'Don't let it bother you if you don't understand. I just say to myself that I probably won't be able to understand the whole game any time soon. So I just play.' I begin to look through dictionaries in which orgot is not listed and finally find a reference to it embedded in the game itself, in a file called READ ME. The text apologises for the fact that orgot has been given several and in some ways contradictory meanings in this version of *SimLife*, but one of them is close to organism. Tim was right – enough.

Tim's approach to *SimLife* is highly functional, but his relationship to *SimLife* – comfort at play, without much understanding of the model that underlies the game – is precisely why educators worry that students may not be learning much when they use 'learning' software. When I interview a tenth-grader named Marcia about *SimCity* where players are put in the role of city mayor with near-total decision-making authority, she boasts of her prowess and reels off her 'Top ten most useful rules of Sim'. Among these, her rule number six grabs my attention: 'Raising taxes always leads to riots.'

Marcia has no language for discriminating between this rule of the game and the rules that operate in a 'real' city. She has never programmed a computer. She has never constructed a simulation. She has no language for asking how one might write the game so that increased taxes led to increased productivity and social harmony. And she certainly does not see herself as someone who could change the rules. Like Tim confronted with the orgot, she does not know how to 'read' a simulation. Marcia is like someone who can pronounce the words in a book but doesn't understand what they mean. She does not know how to measure, criticise, or judge what she is learning.

Marcia may not need to see the registers on her computer or the changing charges on a computer chip, but she needs to see something. She needs to be working with simulations that teach her about the nature of simulation itself, that teach her enough about how to build her own that she becomes a literate 'reader' of the new medium.

We come to written text with centuries-long habits of readership. At the very least, we have learned to begin with the journalist's traditional questions: Who, what, when, where, why, and how? Who wrote these

Presentation
its own powerf
id

words, what is their message, why were they written, and how are they situated in time and place, politically and socially? The dramatic changes in computer education over the past decades leave us with serious questions about how we can teach our children to interrogate simulations in much the same spirit. The challenge is to design worlds that develop habits of critical readership appropriate to a culture of simulation. When Arthur Luehrmann said that the 'real story' was the 'closing down' of the computer he was not simply referring to whether students would keep learning to program in Logo or BASIC. We live in a society where we understand things through computer models. This means that our students need to understand how they are built. Opacity is not the answer.

Presentation as its own powerful idea

One of the most popular programs in educational circles today is Microsoft's PowerPoint. A Google search linking PowerPoint and classroom came up with over 237,000 items. Twenty-four thousand were specifically about PowerPoint use in Kindergarten. The popularity of the presentation software is not surprising. Children enjoy the experience of fluency at something genuinely 'adult'. They know that their parents really use PowerPoint at work, the very same one that they are using at school. Parents are happy that their children are learning something useful, something that they can use in the 'real world'. Administrators are happy because the students and parents are happy and teachers are happy because the students and the parents and the principal are happy. Additionally, many teachers claim that PowerPoint simplifies assignments and motivates through its use of multimedia. In my own observations of children using PowerPoint I am left with many positive impressions. It helps some students organise their thoughts more effectively. It is an excellent note-taking device. Flashcards can be automated, personalised, annotated, and illustrated. It can accommodate different cognitive styles: those who think in charts, those who prefer pictures, those who learn through narratives. But there is also reason for pause.

When PowerPoint is used in elementary and junior high school classrooms it teaches a great deal that is independent of the content of any student's particular slide show. The software does not encourage students to make an argument. They are encouraged to make a point. PowerPoint encourages presentation not conversation. Students grow unaccustomed to being challenged. Ambiguity is not valued. A strong presentation closes down debate rather than opening it up because it conveys absolute authority. What works so well for silencing opposition in the corporate boardroom works just as well in sixth grade but not to good effect. Teachers take books off reading lists if those books 'don't give good PowerPoint'. The books that get taken off are the books that are complicated, hard to summarise. Poetry 'gives' notoriously poor PowerPoint but poetry units are put on PowerPoint anyway. Clear exposition has long seen itself as dependent on clear outlining. The global reach of presentation software has fetishised the outline. The many styles that can co-mingle in a single PowerPoint presentation suggest that it might enable multiple styles of thought. But the presentation itself (and the finger that imposes the cadence of the slides) implies a unitary authorial voice.

And this is perhaps where the new presentational aesthetic holds the greatest danger. Simulation enables us to experiment with multiple identities, with experiencing different perspectives, genders, and positions in narrative and space.[4] Yet, in practice, and most particularly in educational practice, the presentational aesthetic gives authority to the unity of outlined points and the unified narrative they describe. The critics of programming as an aesthetic were concerned that it would take children away from the complex flow of literary ideas. It is ironic that tens of thousands of websites devoted to using PowerPoint in the teaching of literature model how to discuss poetry through the use of bulletpoints. Advocates of programming wanted it to spark discussions of psychology; the practice of PowerPoint can turn discussions of literature into stark outlines.

Notes

Artifacts carry ideas, aesthetics, and ways of seeing the world. I am concerned that the movement from transparency to opacity in educational technology has compromised children's sense that understanding is accessible and action is possible; we are challenged to build educational microworlds where we rediscover new forms of transparency appropriate to our times and technologies. I am concerned that when presentation becomes its own powerful idea we diminish our appreciation of complexity. Watching sixth and seventh graders using PowerPoint to present final class projects on 'World Religions', 'The Civil War', and 'Favourite Poems' is sobering. There are whooshing sounds, many colours, assertions of fact that screech in from left to right, from upper right screen to centre. Things are very simple. There is little discussion. There is much applause when the special effects are particularly appealing. Only some of this can be 'blamed' on the technology. Some of the classrooms are overcrowded. Some of the teachers are harried. But the PowerPoint doesn't help.

When I first began studying the computer culture, hackers were commonly called 'computer people'. No more. In a certain sense, if we take the computer to be a carrier of a way of knowing, of a way of seeing the world and what is important, we are all computer people now. We spend much of our lives staring at screens. We live much of our lives in artificial worlds. Our educational technology – our simulation worlds and our presentational software – present scintillating surfaces and compelling interactions but when we use them we are too often not teaching about complex relationships or about how to think about battles that don't end in 'infinite justice'. Computation gives us powerful tools to think about multiple realities but we most often use them to think in terms of bullet points instead. The real world is messy and painted in shades of gray. In that world we need to be comfortable with ambivalence and contradiction. We need to be able to put ourselves in the place of others in order to understand their motivation. In my view, no matter what their merits, until we learn to use educational technologies to help us with this stuff, we have to enjoy their pleasures in measured doses.

Notes

[1] I began working with Seymour Papert and the Logo group at the MIT AI Laboratory (which would become the Learning and Epistemology group at the MIT Media Lab) in 1976. I studied school settings in which the Logo language was integrated into the curriculum. My first round of studies of computers in education were reported in *The Second Self: Computers and the Human Spirit* (Cambridge, MA: MIT Press, 2nd revised edition, forthcoming). Later work on education in the culture of simulation was reported *in Life on the Screen: Identity in the Age of the Internet* (New York: Simon and Schuster, 1995).

[2] See Seymour Papert, *Mindstorms: Children, Computers, and Powerful Ideas* (New York: Basic Books, 1980).

[3] Here I note the active educational interests of the early computer hobbyist movement. For a description of the connection they made between programming and activism see Turkle, *The Second Self.*

[4] Indeed, in *Life on the Screen* I have gone so far as to argue that the computer is a carrier of postmodernism, that computation brings postmodernism 'down to earth'.