

From: Turkle, S., 1984(2004), *The Second Self: Computers and the Human Spirit*, MIT Press.

Introduction to the MIT Press Edition (2004)

When writing this book about computers and people, I immersed myself in a world that was altogether strange to me. Trained as a humanist, I took a job at MIT in the late 1970s. There I was surrounded by people who spoke about the mind in an unfamiliar language of bits and bytes, registers and compilers. Many of them had strong, even passionate relationships with digital machines. I had students and colleagues who claimed that building and programming computers was the most powerful intellectual and emotional experience of their lives, an experience that changed the way they thought about the world, about their relationships with others, and, most strikingly, about themselves. I first heard such extravagant sentiments expressed by computer professionals, but in the course of my six-year study I came across them in personal computer clubs and grade school classrooms. "When you program a computer, there is a little piece of your mind and now it's a little piece of the computer's mind," said Deborah, a sixth-grade student in an elementary school that had recently introduced computer programming into its curriculum. Her comment stayed with me and inspired my title.

The Second Self documents a moment in history when people from all walks of life (not just computer scientists and artificial intelligence researchers) were first confronted with machines whose behavior and mode of operation invited psychological interpretation and that, at the same time, incited them to think differently about *human* thought, memory, and understanding. In consequence, they came to see both their minds and computational machines as strangely unfamiliar or "uncanny" in the sense that Sigmund Freud had defined it. For Freud, the uncanny (*das Unheimliche*) was that which is "known of old and long familiar" seen anew, as strangely unfamiliar.¹

Psychoanalysis shares with computation a subversive vocation: each in its own way defamiliarizes the mind. In the Cartesian tradition, the mind is

taken to have immediate and privileged knowledge of itself. There is nothing in nature that each of us, theoretically speaking, is in a better position to comprehend than our own mind. Psychoanalysis called this transparency of mind into question. It asserted that our conscious thoughts and actions, our deepest feelings and our strongest moral convictions, are shaped by powerful psychical forces of which we are not normally aware. It pointed to serious, previously unrecognized obstacles to self-knowledge. According to psychoanalysis, the mind—known of old and thought to be quite familiar—was actually unexplored territory, an internal but expansive *terra incognita*.

The computer, too, called longstanding assumptions about self-understanding into question. From the earliest days, computer science borrowed terms from everyday psychology to describe the operations of computing machines just as psychology borrowed language from computer science to describe the mind. Most strikingly, it was common to speak of a computer's "memory" at a time when behaviorism was insisting that all one could study in people was the behavior of "remembering." Computers helped to relegitimate the notion of memory within academic psychology, and with the introduction of computers into mainstream culture in the late 1970s and early 1980s, people in their everyday conversation began to describe human mental activity in computational terms. ("Excuse me, I need to clear my buffer; I won't be happy until I debug this problem.")

While I was writing *The Second Self*, I introduced the idea of slips of the tongue in my MIT classroom. At that time, one of my students recast the idea of Freudian slips as "information processing errors." We had read a text in which Freud described the chairman of a parliamentary session opening the session by declaring the meeting closed.² My student, who thought of her mind as a computer, saw the substitution of "closed" for "open" not as a way to understand the chairman's possible ambivalence, but as a "bit being dropped," perhaps due to a power surge. With the transition from a psychoanalytic to a computational metaphor of mind, an explanation in terms of meaning had shifted to an explanation in terms of mechanism. With this, came an attendant question: If mind is program, where is free will?

By the mid-1980s, by the way it posed such questions, the computer had become an evocative object, an object that provoked self-reflection. Philosophical questions that had been traditionally confined to seminar rooms were concretized in discussions about what computers could do. Computers brought philosophy down to earth. Even children playing with the first generation of computer toys and games were asking new questions about the machine's "life" and "mind" and then, by extension, wondering what was special about their own.

Twenty years later, the computer would seem secure in its role as an evocative object for thinking about human identity. Cognitive science has developed far more sophisticated computational models of mental processes than were dreamt of two decades ago, and the Internet has opened up new paths for the exploration of self and sociability. However, with time grows a sense of familiarity. What was once exotic begins to seem “natural.”

Today, we take for granted our lives with computation (our personal computers, personal digital assistants, our cellphones that serve as organizers and cameras) and within computation (our computer games, e-mail, instant messaging, and online communities). And we show increasing nonchalance about the idea of computation within ourselves. In the medical arena, cochlear implants are a current reality, and we look forward to computational implants that might help with epilepsy, Parkinson’s, and Alzheimer’s. In 1984 the notion of mind as program was controversial. These days, the use of computational metaphors to speak about the mind has become banal.

In the early days of the computer presence in the wider, nontechnical culture, the time frame of *The Second Self*, it was commonplace to describe the computer as “just a tool,” in a way that dismissed its effects on child development and on our emotional lives. In *The Second Self* I was writing *against* the common view that the computer was “just a tool,” arguing for us to look beyond all the things the computer does *for* us (for example, help with word processing and spreadsheets) to what using it does *to* us as people. I was helped in this task by the very newness of the computer. Most people could remember when it hadn’t been around. In the twenty years that followed, the situation became more complex. The trend was for new computational objects—personal digital assistants (PDAs), cellphones, laptops—to become even more intimate partners to their users, more like thought-prosthetics than simple tools. The subjective side to computer technology became more apparent, even as the ubiquity of these objects began to dull our sensitivity to their effects.

In this case there are virtues in learning to see the commonplace as unfamiliar. Psychoanalysis relies on the analytic experience, the “talking cure,” to defamiliarize the mind to itself and thus reveal what would otherwise be hidden in the light. Anthropologists similarly address the question of how to see one’s own culture in sharper relief by spending time in another. They refer to this displacement as *dépaysement*, quite literally, de-countrifying. *Dépaysement* need not involve travel. What matters is immersing oneself in something foreign so that upon returning home the familiar has become strange—and can be seen with fresh eyes.

It is my hope that the republication of *The Second Self* will afford its readers a chance to engage in an intellectual *dépaysement*: not only to (re)experience the now almost-foreign computer culture of the late 1970s and early 1980s but to view our contemporary computer culture from a new perspective.* Many of my readers may have forgotten (and younger readers never knew) what it was like to experience the personal computer as a problematic object, one that defied easy categorization and troubled the mind.

In the 1980s many parents were concerned about putting children and computers together; there seemed to be something unnatural about the combination. Today, when school systems can afford them, computers are taken to be a basic classroom tool: PowerPoint presentation software is routinely taught to third graders with the approval of most parents. When children were introduced to video games in the 1980s, there was serious discussion of banning them using the same statutes that outlawed addictive substances such as heroin and marijuana. These days, video games have become a staple of home entertainment, and Internet-based multiplayer games are a routine pastime for hundreds of thousands. The 2002 launch of *The Sims Online*, an Internet-based multiplayer game, made the cover of *Newsweek* magazine in the anticipation that some day millions of people would live parallel lives in virtual communities.

In general, we have come to accept current, specific applications of computer technology as inevitable. We lose sight of the fact that things were once different and might have developed along other paths if different decisions had been taken by manufacturers and consumers, by educators and governments. Yet, if we hope to construct the richest lives possible with this technology, we must not lose our sense of its many potentials and not see its current direction as inevitable or determined.

Looking back at the recent history of the computer culture should make it easier to look critically at past decisions, sharpen the terms in which decisions yet to be made are framed, and deepen our conversations about who we are becoming in our increasing intimacy with our machines.

The View from Twenty Years

In the twenty years since the first publication of this book, computation has become more complex, but fundamental aspects of how people relate

* To this end, I have made only minor revisions—removing errors and making clarifications—to the 1984 text and notes. Substantive additions, both in the text and in notes, are indicated by the use of *italics*.

to the seductions of interactive media have stayed constant. In this sense, *The Second Self* remains a primer in the psychology of people's relationships with computers. Computational objects, poised between the world of the animate and inanimate, are experienced as both part of the self and of the external world. This is as true today as it was for those early adopters of computer technology I studied in the late 1970s and early 1980s. The remark about programming that inspired my title (thirteen-year-old Deborah saying, "There is a little piece of your mind and now it's a little piece of the computer's mind . . .") has profound analogies with a recent comment by a woman who spoke of her personal digital assistant and said: "When my Palm crashed, it was like a death. It had my life on it . . . I thought I had lost my mind."

One step beyond the PDA that has one's "life on it" is the development of "wearable computing." More recently, people who may refer to themselves as "cyborgs" wear their computers: the central processing unit and radio transmitter in one pocket, a tiny keyboard in another, their eyeglasses serving as screens. The designers of such systems talk about new possibilities for information access: one can be online all the time, for example, in conversation with a faculty colleague while at the same time reading that colleague's most recent papers. The cyborgs, however, testify to effects of the technology on a very different register: they say that wearable computers change their sense of self. For one, "I become my computer. It's not just that I remember people or know more about them. I feel invincible, sociable, better prepared. I am naked without it. With it, I'm a better person." Over the past twenty years, there have been several revolutions in computer hardware and software, but the projection of self onto computational media is as consistent as it is dramatic. In 1984, referring to that projection by calling computers a "second self" was provocative. Today, it does not go far enough. To be provocative, one is tempted to speak not merely of a second self but of a new generation of self, itself.

Yet there are some things that have not been carried forward. Schoolchildren learning to program under innovative educational initiatives of twenty years ago, such as Deborah, are among this book's central actors. So, too, is a vibrant culture of personal computer owners who built and bought home computers for the joy of understanding how they worked. But in today's cultural mainstream, these actors are no longer with us. The socially shared activity of computer programming and hardware tinkering has been displaced by playing games, participation in online chat and blogs, and using applications software out of the box.

What are the differences between programming a computer that quite literally “re-minds” you of your mind—what Deborah was able to accomplish even with her very primitive programming skills—and the experience of externalizing your schedule or sensibility on the PDAs, computer desktops, or Web sites, where “second selves” are constructed in contemporary personal computing? Deborah was programming in the Logo programming language, which enabled her to “drive” a screen cursor known as a *turtle*. The turtle left traces of its path on the screen. Children were taught to give explicit commands to the turtle (such as FORWARD 100; RIGHT TURN 90) that caused it to trace geometric patterns that children could then capture in a program. Deborah used her programming skills to create a “microworld” on the computer, a rule-driven universe of her own design.

Deborah restricted the commands she could give to the turtle: she would allow herself only one turning command—a right turn of thirty degrees. Once she had her rule, she got down to serious work, an explosion of creativity. Most important was how her rule made her feel. Away from the computer, she felt out of control. She was struggling with overeating and the temptations of smoking. While on the computer she felt herself in a situation simple enough for her to feel in control yet varied enough for creative exploration. In chapter 4, when I tell Deborah’s story, I stress that her need was for a world apart in which she could build a new set of distinctions that she could then transfer to thinking about herself. The computer provided this world and gave her categories more useful than “I am good” or “I am bad.” With the thirty-degrees world she had a new way to think about her problems. She was able to go beyond thinking of herself as bad, to thinking, “I am in trouble because I have no rules. I am not in control. And I should be. I can be.” The computer, quite literally, became Deborah’s object-to-think-with for thinking about herself.

The experience of authorship in programming gave children like Deborah a sense of control that enabled them to construct microworlds that were exquisitely tuned to their own developmental needs. Programming provided a medium for projection—in *The Second Self* I refer to it as a “Rorschach effect.” But unlike the Rorschach inkblots, programming also provided a means for people to work through personal issues, as Deborah had been able to do with her thirty-degrees world.

These days, if a child such as Deborah tried to work out her need for structure with a computer, she would more likely turn to the activity of building personal avatars in virtual space or joining a team of online adventurers in a rule-based multiplayer universe—in the past ten years the

most popular of these have included *EverQuest*, *Ultima II*, and *Asheron's Call*. In these contexts, she would have a large canvas for identity play. She could choose a new name—say, “Rule_Girl”—and develop a play pattern that made her feel safe, perhaps by joining a player class that could only function in highly constrained ways. In a medieval online game, she might not play a magician but a serf or a knight who operated under an elaborate code. There would be rich possibilities for experimenting with identity; she would be playing with other people, *yet the game would be of someone else's creation*. When Deborah created the thirty-degrees world she had the sense not only that it was all hers, but that she understood how it worked. From her perspective, if not perhaps from a computer scientist's, Deborah's microworld was both self-authored and transparent.

Transparency and Opacity

For me, among so many changes to the landscape of twenty years ago, the shift in expectations about technological transparency stands out as particularly striking. Early personal computers, like their mini and mainframe cousins, used operating systems and programming languages that gave users a feeling of contact with the “bare machine.” I wrote *The Second Self* on an Apple II computer that had, quite literally, been torn bare. Its cover had been removed and its operating system replaced with another called CP/M. In order to communicate with my computer, for example, to ask it to summon the word processing program called Scribble, I had to give it specific symbolic commands that I understood as my means of addressing the machine “below.” And once I was dealing with Scribble, I was still in a world of commands, this time to format my text. For example, to indicate that I wanted a flush left heading, “Transparency and Opacity,” printed in bold face, I would type “@left[@b(Transparency and Opacity)].” Every command I issued was a line of text, a neat string of symbols—requirements that kept me in touch with the idea that I was directly addressing a machine, speaking to it in its language. I felt that I had to use symbols and a formal language of nested delimiters (parentheses and brackets) because my machine needed to reduce my commands to something that could be translated into electrical impulses. The fact that my Apple II's printed circuits were physically exposed only reinforced this notion.

Although I did not build my own personal computer from a kit or learn to program in assembly language as did many of the early home computer enthusiasts I interviewed, my experience with CP/M and my naked Apple II provided a reference point for my understanding the aesthetic of

technological transparency that I met in the early personal computer culture. Such transparency was described by one enthusiast as “the pleasure of understanding a complex system down to its simplest level.” This was a culture committed to developing a relationship with the computer as a rule-based, understandable machine. It was a culture in many ways reminiscent of that around early automobiles, a world in which most drivers understood the workings of the internal combustion engine, or at least how to fix it in a pinch.

There were, of course, competing aesthetics among the subcultures of computing in the early 1980s. While the personal computer hobbyists were committed to the view that “the machine only does what you tell it to, nothing more, nothing less,” many artificial intelligence researchers were committed to an aesthetic of emergence, where the machine would quite precisely do more than you could ever specify. For them, the beauty of the computer was that programmed agents within a computer system, operating with simple rules, could, through their interaction, create unexpected, “emergent” behavior. Another challenge to the aesthetic of rule-based transparency came from engineers and designers who believed that communication with computers should not rely on commands but on a more fluid and gestural language. To these designers, there was no need for a user to ever address a machine’s underlying mechanism. In their view, computer users should be liberated from having to think about the machine at all. In the late 1970s and early 1980s, during visits to Xerox PARC, a research laboratory in Palo Alto, I was first introduced to computers that offered the new gestural and “conversational” style. Instead of commanding my word processing program through typed commands, I was given a pointing device and shown how to gesture to a screen icon. At Xerox PARC, I met my first computer mouse and saw my first graphical user interface. Both of these technologies became part of the public face of computing during the year this book was first published, 1984, the year of the Macintosh.

The aesthetic of rule-based communication with a bare machine that I had met in the early days of personal computing would not survive the computer becoming a consumer object. However, through the 1980s, one could find its legacy, its style of command, in the DOS operating language for the IBM personal computer and again in the early Windows operating system, built on top of DOS, which enabled a user to “reach back” into DOS and recapture the feeling of directly addressing a machine. In contrast, the Macintosh, like the computers I had met at Xerox PARC, introduced a way of thinking that put a premium on the manipulation of

a surface simulation. Macintosh users worked with a new understanding of the word transparency, indeed, one that turned common usage on its head. If one used the DOS operating system, things felt transparent when computer use felt analogous to working on a traditional mechanical device, like a car. Specific instructions to a computer enabled one to “open the hood” and “poke around” its inner workings. But when Macintosh users spoke about transparency, they were referring to an ability to make things work *without* going below a screen surface filled with attractive icons and interactive dialogue boxes. Indeed, these screen objects suggested that communicating with a computer could be less like commanding a machine and more like having a conversation with a person. In only a few years, the “Macintosh meaning” of the word transparency had become a new lingua franca. By the mid-1990s, when people said that something was transparent, they meant that they could immediately make it work, not that they knew how it worked.

In *The Second Self*, I discussed how different computer languages and architectures suggested different ways of thinking. In the years that followed, this idea, which had once seemed esoteric, played out on the larger cultural stage in the “Macintosh/IBM wars,” or otherwise understood, the “Macintosh/Microsoft wars.” The face-off between the competing operating systems was about more than industrial loyalties or personal style. There was also a conflict of intellectual values. In the late 1980s, provoked by the reception of the Macintosh, my thoughts increasingly turned to how computational objects carried ideas. By 1995, in *Life on the Screen*, I was able to characterize the transition of sensibilities that marked the introduction of the Macintosh by saying that with the Macintosh (and then, by extension, the ubiquitous Macintosh-style Windows interfaces introduced by Microsoft), people had moved away from a reductive and mechanistic view of how to relate to a computer and were “learning to take the machine at (inter)face value.”³

In the 1970s and early 1980s, computers carried a modernist ethos: analyze and you shall know; by the mid-1990s, the complex simulation worlds of opaque computers offered an experience that called these assumptions into question. Culturally, the Macintosh carried the idea that it is more fruitful to explore the world of shifting surfaces than to embark on a search for mechanism, origins, and structure.

The Macintosh way of understanding stood in contrast to the modernist perspective that animated the writings of such thinkers as Freud, Marx, and Darwin, who suggested that understanding proceeds by reducing complex things to simpler elements, by discovering the hidden

mechanisms behind behavior. Analyze and you shall know presented itself as a way of understanding the self and the social world. As a way of thinking, it animated the personal computer owners I write about in *The Second Self*. They were populist computer utopians who saw the computer as providing widespread access to information (previously available only to elites) that would encourage political engagement. Beyond this, they believed that a transparent relationship with computers would be empowering, that once people could own and understand something as complex as a computer, they would demand greater transparency in political decision-making processes.

Certain elements of the future they imagined have been realized in the past twenty years. Online communities bring people together for political purposes. (In the 2003 Democratic Party primary campaign, many first appreciated how the computer could be put to this use through the work of MeetUp.com—a central organizing tool for Howard Dean’s run for the Democratic nomination. In the election that followed, Internet organizing became a political staple.) Blogging (online journalism that provides a new, distributed source of news and commentary) is a potent force. Even online game worlds have provided politically evocative objects. Early controversy about *The Sims Online* did not question the premise of living a virtual life in an online suburbia. Rather, it interrogated the nature of politics in that world. The McDonald’s Corporation purchased virtual real estate in the game, along with the right to sell virtual fast food to Sim citizens. Some objected, and this led to much discussion of how one might counter the corporate move (online picketing? online boycotts?) as well as to strenuous debate about whether game players were citizens or consumers and about the meaning and ultimate effectiveness of virtual protest.

However, as I have noted, the political hopes of the first-generation personal computer users were pinned not only on how the computer presence would democratize access to information, but on how a particular relationship with the computer (a sense of the machine’s transparency) would generalize to a new and more empowered relationship with politics. In the main, these hopes have not been realized. In 2004 the cultural message of digital technology is not about simplicity but complexity, not about transparency but opacity. This transition has played itself out in many arenas of the computer culture. From a societal perspective, one of the most significant of these has been in the role of computers in education.

Through the mid-1980s, when educators wanted to make computation transparent, they taught students about the logical processes carried out inside the computer and instructed them in programming languages. In

his influential *Mindstorms: Children, Computers, and Powerful Ideas*, published in 1980, MIT's Seymour Papert argued that learning about the computer should mean learning about the powerful ideas embodied in it.⁴ This was the culture of computing that Deborah inhabited as she worked in the Logo language, developed by Papert and others. Papert hoped that the process of writing programs would teach children how to "think like a computer" and to understand how simple programs could be used as building blocks for more complex ones.

In *The Second Self* I report on my studies of children learning Logo. Their styles of programming were varied and revealing. The computer, as I have said, served as a Rorschach, and programming was one of the most powerful manifestations of its projective power. Twenty years later, programming is no longer taught much 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. But the question remains whether mastery of these skills should be the goal of computer education. Do they constitute computer literacy?

The move toward teaching computer utility programs as a "computer education" curriculum was far along by 1996, when I spoke with a group of teachers at a June meeting of MassCUE (Massachusetts Computer Using Educators).⁵ Most of the eighty or so teachers present had been in computer education for over a decade. In the 1980s, many had seen their primary job as teaching the Logo programming language because they believed that it communicated important thinking skills. One teacher described those days: "Logo was not about relating to the hardware of the computer, so it wasn't about how the computer 'worked' in any literal sense, but its claim was that it could teach about procedural thinking. It could teach about transparency at its level." Another added, reflecting on Logo: "The point was not that children needed to understand things about the simplest level of how the hardware worked, but that things needed to be translated down to an appropriate level, I mean, a relevant level." Someone challenged her, asking how she knew what level was relevant. She stumbled, and looked around to her fellow teachers hesitantly, questioningly. A colleague tried to help: "You have to offer children some model of how a computer works because the computer needs to be demystified. Children need to know that it is a mechanism, a mechanism that they control." Here, in the world of computers in education, a fight for the

aesthetic that had animated the early personal computer movement was being played out, two decades later.

By now the conversation was heated. Another teacher argued that presenting children with a view of the computer as “controllable” could itself be misleading. Today’s programs were so complex as to be out of control and students needed to learn this disquieting fact. For her, the reason to teach programming centered on children feeling empowered to embody their imaginations in code. One unhappy seventh-grade teacher concurred, “It’s not my job to instruct children in the use of an appliance and then to leave it at that.” These teachers were struggling toward an argument for a certain kind of “computational exceptionalism.” It takes as a given that people once knew how their cars, televisions, or telephones worked and don’t know this any more, but that in the case of mechanical technology, such losses are acceptable. It insists, however, that ignorance about the fundamentals of computation comes at too high a price. One teacher put it this way: “Children know that the telephone is a mechanism and that they control it. But it’s not enough to have that kind of understanding about the computer. You have to know how a simulation works. You have to know what an algorithm is.”

In the nearly ten years since I recorded these conversations, educational advocates for computational transparency have, in large measure, lost their battle. Educators who want to demystify the computer face a new generation of children that no longer finds enough mystery in the machine to care what an algorithm is. It is a generation that has made a transition from the transparency of algorithm to the opacity of simulation. This generation takes overland journeys along a simulated Oregon Trail and when it plays *The Sims* or *The Sims Online*, it designs houses, personal histories, and social engagements for the virtual citizenry. In *The Second Self*, when I wrote of the “computer as Rorschach,” it was programming that served as the projective screen for personal and cultural differences. These days, computation offers far more immediate projective media: one can create multiple avatars in online communities and play with relationships, quite literally using one’s “second (or third, or fourth, or fifth) self.”

Simulation and Its Discontents

The games of the Sim series, first introduced in the early 1990s, socialized a generation of children into the culture of simulation, and perhaps above all, into its aesthetic of opacity. In *SimCity* you engage in civil engineering and urban policy planning; in *SimLife* you design ecosystems and the

organisms to inhabit them; in *The Sims* you create a family and attempt to steer its members toward social, financial, and emotional well being. *The Sims Online*, the site of the virtual McDonald's franchises, takes the Sims concept onto the Internet where you create online avatars and play the game with thousands of others in a networked virtual community. In each of these simulation games the goal is to make a working system from complex, interrelated parts. In no case does the user design or modify the algorithms that underlie the game. Success comes rather from developing an understanding, through trial and error, of a system designed by others.

I have suggested, in talking about Deborah, that on the level of the individual child, something interesting has been lost in the move away from authorship of the programs that underlie one's own game. On a societal level, there is an analogous loss. The aesthetic of transparency (common to the Logo movement and the early generations of personal computer hobbyists) carried with it a political aesthetic that was tied both to authorship and to knowing how things worked on a level of considerable detail. This is a kind of understanding that is not communicated by playing off-the-shelf simulations.

On one level, high school sophomores playing *SimCity* for two hours may learn more about urban planning than they would from a textbook, but on another level, they may not know how to think about what they are doing. They "play" simulations but don't have a clear way to discriminate between the rules of the game and those that operate in a real city. Most have never programmed a computer or constructed their own simulations. They do not have a language for talking about how one might rewrite the rules of their games. So, for example, *SimCity* often gives players the impression that raising taxes will lead to riots. But, of course, there is a way to write the game so that increased taxes lead to an increase in health services, productivity, and social harmony. In my view, citizenship in a culture of simulation requires that you know how to rewrite the rules. You need tools to measure, criticize, and judge every simulation. Today's teenagers are comfortable as inhabitants of simulated worlds, but most often, they are there as consumers rather than as citizens. To achieve full citizenship, our children need to work with simulations that teach about the nature of simulation itself.

The utopian vision of the computer culture that animated many of its 1980s pioneers was that computers would lead to unprecedented opportunities for participation in every area of social and cultural life. The reality of simulation culture as it has developed, whether in games, the

professions, or politics (where simulations are central to planning) is that those who write the simulations get to set the parameters.

In 1995, in *Life on the Screen*, I wrote about my encounter with Tim, a thirteen-year-old whose experience with *SimLife* stands in stark contrast with my encounter with Deborah of the thirty-degrees rule, only a decade before. Deborah worked in a simple system that she built by herself. Tim, who did not know how to program, worked in a complex system built by others. Tim played his simulation software as though it were a video game, moment to moment, with no understanding of the rules. Deborah was nurtured by transparency; Tim's skill set was centered on the artful navigation of opacity. His philosophy of play: "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 it."⁶

Tim's method enabled him to accomplish a great deal in simulation space. His comfort in his virtual world might serve him (not well, but adequately) in the many possible careers that lay before him, careers in architecture, law, business, medicine, or history. In all of these fields, dealing with information increasingly entails the navigation of simulations of other people's creation. However, as I meet professionals in all of these fields who move easily within their computational systems and yet feel constrained by them, trapped by their systems' unseen limitations and unknown assumptions, I feel continued concern. Are the new generations of simulation consumers reminiscent of people who can pronounce the words in a book but don't understand what they mean? 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 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 specific questions may be different, but the intent needs to be the same: to develop habits of readership appropriate to a culture of simulation. These habits of readership are central to computer literacy and social responsibility in the twenty-first century.

Thinking about computer literacy and social responsibility in simulation space cannot be an exercise that takes schoolchildren as its only subjects. Architects design our cities in virtual spaces; biologists study protein molecules that are possible to envisage only as screen objects, medical students learn dissection on virtual cadavers. Every time we log on to our e-mail

accounts we are able to create different user names, different personae, different selves through which to live our lives. *The Second Self* was written at a time when virtuality seemed new. Chapter 2, on the nascent video game culture, pointed toward the seductions of simulation to come. There, we began to see a tension between life in the physical and virtual worlds. These days, that tension defines our cultural situation.

Now, as in the mid-1980s, we stand on the boundary between the physical and virtual. And increasingly, we stand on the boundary between worlds we understand through transparent algorithm and worlds we understand by manipulating opaque simulation. Our current experience of life “betwixt and between” recalls what the anthropologist Victor Turner termed a “liminal moment,” a moment of passage. It is a moment of anxiety, but it is also a moment of invention and creativity.⁷ When Turner spoke of liminality, he understood it as a transitional experience, but for us, living the tension between physical and virtual and between analysis and simulation, seems a permanent state of affairs, our permanent existence on the edge of things.

In the late 1990s I took my daughter, then seven, on a vacation in Italy. We took a boat ride in the postcard-blue Mediterranean. She saw a creature in the water, pointed to it excitedly, and said: “Look Mommy, a jellyfish. It looks so realistic.” When I told this story to a research scientist at the Walt Disney Company, he responded to it by describing the reaction of visitors to Animal Kingdom, Disney’s newest theme park in Orlando, populated by “real”—that is, biological—animals. The first visitors to the park expressed disappointment that the animals were not “realistic” enough. They did not exhibit the lifelike behavior of the more active robotic animals at Disney World, only a few miles away. What is the gold standard here? A life in simulation has left my daughter’s generation suspended in play yet newly alive. Our displacement from the traditions of the physical by the shadow of the virtual has created a new kind of *dépaysement*, providing the opportunity for a clearer view of both registers.