

## 28. [Introduction]

# From *Mindstorms*

### *Children, Computers, and Powerful Ideas*

While mainframe computers were hidden away and attended by a priestly class of experts, more populist ideas of the computer were also being hatched. These entered the mainstream in what became, in the 1980s, the home computer era—an era not only heralded but negotiated by the acceptance, mainly by children, of computer-driven home video game consoles. But before that, a few had seen the computer not just as a commercial appliance that could find a wide market for gaming and productivity purposes but also as a means of learning. While computer operators were still solemnly receiving punched cards, a few—most prominent among them, Seymour Papert—were working to provide environments in which children could learn through the process of programming that most esoteric device.

Papert, a student of Jean Piaget's in the early 1960s, developed a philosophy of education based on Piaget's work and called "constructionism." It differs from constructivism (another educational philosophy based on Piaget) in that it "looks more closely . . . at the idea of mental construction," as Papert writes in *The Children's Machine* (143). The computer is seen by Papert as a powerful tool for supporting children's learning—for learning in a self-directed, self-motivated way, in the course of programming. Rather than seeing the computer as a mechanism for instilling knowledge and skills via workbook-like exercises, Papert's programming language, LOGO, allows children to take control of the computer, learning about mathematics through the experience of mathematical concepts. Work on developing LOGO began in 1967 in the same building that housed MIT's Artificial Intelligence Laboratory and Project MAC, which would become the Laboratory for Computer Science.

While Papert clearly believes in the power of the computer to facilitate constructionist educational experiences, computing is a means to an end for him and other constructionist educators. Constructionist education has certainly been an influence on new media, an influence that can be seen most directly on Web sites like *MaMaMedia* (founded by one of Papert's students) and on toy store shelves (the Lego Mindstorms robot kits, based on development done at MIT, are named after Papert's book). Papert was absolutely correct in his prediction in the following selection that "before the end of the century, people will buy children toys with as much computer power as the great IBM computers currently selling for millions of dollars." A computer is not required to allow children to learn in a constructionist way, however—just as software alone will not bring constructionist education to a traditional classroom. Constructionism, as potentially revolutionary to educational philosophy as the computer chip is to information technology, does not come packaged with computing. As yet, it has not upset traditional education to the extent that computing has influenced daily life.

"What follows is a hypothetical conversation between two children who are working and playing with the computer," Papert writes in introducing the illustrated section of *Mindstorms* that is reproduced here, one which succinctly reveals the approach of constructionism. "These and other experiments can happen every day—and they do." With continued development of programming languages and learning environments for children (now a major part of the MIT Media Lab, the University of Maryland's Human-Computer Interaction Lab, and a number of other leading research centers), these sorts of conversations continue to happen today, in a growing number of contexts.

—NM

See Ted Nelson's "No More Teachers' Dirty Looks" (◇21) for a 1970 exploration of the educational computing issue, one written prior to the appearance of the first personal computer. While Nelson is similarly opposed to programming children via computer-aided instruction, he discusses ways of learning that employ the computer as a media system rather than a programming environment.

Papert credited artificial intelligence pioneer Marvin Minsky as being "the most important person in my intellectual life" in the years leading up to *Mindstorms*. Papert also thanked Alan Kay, noting that the only research group outside MIT seriously working with computers for children was Kay's at Xerox PARC. See Kay and Goldberg's article (◇26), which describes Smalltalk, another programming language used in education. Another person whose influence Papert noted was MIT social scientist Sherry Turkle, who has worked extensively on children's perceptions of computers; see selections from her book in ◇34.

## Further Reading

Boden, Margaret A. *Jean Piaget*. New York: Viking, 1979.

Druin, Allison, ed. *The Design of Children's Technology*. San Francisco: Morgan Kaufmann, 1999.

Papert, Seymour. *The Children's Machine: Rethinking School in the Age of the Computer*. New York: Basic Books, 1993.

Papert, Seymour. *The Connected Family: Bridging the Digital Generation Gap*. Atlanta: Longfellow Press, 1996.

## Original Publication

"Computers and Computer Cultures," 19-37 and "Turtle Geometry: A Mathematics Made for Learning," 78-93. *Mindstorms*, New York: Basic Books, 1980.

## From *Mindstorms* *Children, Computers,* *and Powerful Ideas*

Seymour Papert

### Computers and Computer Cultures

In most contemporary educational situations where children come into contact with computers the computer is used to put children through their paces, to provide exercises of an appropriate level of difficulty, to provide feedback, and to dispense information. The computer programming the child. In the LOGO environment the relationship is reversed: The child, even at preschool ages, is in control: The child programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think. The experience can be heady: Thinking about thinking turns the child into an epistemologist, an experience not even shared by most adults.

This powerful image of child as epistemologist caught my imagination while I was working with Piaget. In 1964, after five years at Piaget's Center for Genetic Epistemology in Geneva, I came away impressed by his way of looking at children as the active builders of their own intellectual structures. But to say that intellectual structures are built by the learner rather than taught by a teacher does not mean that they are built from nothing. On the contrary: Like other builders, children appropriate to their own use materials they find about them, most saliently the models and metaphors suggested by the surrounding culture.

Piaget writes about the order in which the child develops different intellectual abilities. I give more weight than he does to the influence of the materials a particular culture provides in determining that order. For example, our culture is very rich in materials useful for the child's construction of certain components of numerical and logical thinking. Children learn to count; they learn that the result of counting is independent of order and special arrangement; they extend this "conservation" to thinking about the properties of liquids as they are poured and of solids which change their shape. Children develop these components of thinking preconsciously and "spontaneously," that is to say without deliberate teaching. Other components of knowledge, such as the skills involved in doing permutations and combinations, develop more slowly, or do not develop at all without formal schooling. Taken as a whole this book is an argument that in many important cases this developmental difference can be attributed to our culture's relative poverty in materials from which the apparently "more advanced" intellectual structures can be built. This argument will be very different from cultural interpretations of Piaget that look for differences between city children in Europe or the United States and tribal children in African jungles. When I speak here of "our" culture I mean something less parochial. I am not trying to contrast New York with Chad. I am interested in the difference between precomputer cultures (whether in American cities or African tribes) and the "computer cultures" that may develop everywhere in the next decades.

I have already indicated one reason for my belief that the computer presence might have more fundamental effects on intellectual development than did other new technologies, including television and even printing. The metaphor of computer as mathematics-speaking entity puts the learner in a qualitatively new kind of relationship to an important domain of knowledge. Even the best of educational television is limited to offering quantitative improvements in the kinds of learning that existed without it. "Sesame Street" might

offer better and more engaging explanations than a child can get from some parents or nursery school teachers, but the child is still in the position of listening to explanations. By contrast, when a child learns to program, the process of learning is transformed. It becomes more active and self-directed. In particular, the knowledge is acquired for a recognizable personal purpose. The child does something with it. The new knowledge is a source of power and is experienced as such from the moment it begins to form in the child's mind.

I have spoken of mathematics being learned in a new way. But much more is affected than mathematics. One can get an idea of the extent of what is changed by examining another of Piaget's ideas. Piaget distinguishes between "concrete" thinking and "formal" thinking. Concrete thinking is already well on its way by the time the child enters the first grade at age 6 and is consolidated in the following several years. Formal thinking does not develop until the child is almost twice as old, that is to say at age 12, give or take a year or two, and some researchers have even suggested that many people never achieve fully formal thinking. I do not fully accept Piaget's distinction, but I am sure that it is close enough to reality to help us make sense of the idea that the consequences for intellectual development of one innovation could be qualitatively greater than the cumulative quantitative effects of a thousand others. Stated most simply, my conjecture is that the computer can concretize (and personalize) the formal. Seen in this light, it is not just another powerful educational tool. It is unique in providing us with the means for addressing what Piaget and many others see as the obstacle which is overcome in the passage from child to adult thinking. I believe that it can allow us to shift the boundary separating concrete and formal. Knowledge that was accessible only through formal processes can now be approached concretely. And the real magic comes from the fact that this knowledge includes those elements one needs to become a formal thinker.

This description of the role of the computer is rather abstract. I shall concretize it, anticipating discussions which occur in later chapters of this book, by looking at the effect of working with computers on two kinds of thinking Piaget associates with the formal stage of intellectual development: combinatorial thinking, where one has to reason in terms of the set of all possible states of a system, and self-referential thinking about thinking itself.

In a typical experiment in combinatorial thinking, children are asked to form all the possible combinations (or "families") of beads of assorted colors. It really is quite remarkable that most children are unable to do this systematically and accurately until they are in the fifth or sixth grades. Why should this be? Why does this task seem to be so much more difficult than the intellectual feats accomplished by seven and eight year old children? Is its logical structure essentially more complex? Can it possibly require a neurological mechanism that does not mature until the approach of puberty? I think that a more likely explanation is provided by looking at the nature of the culture. The task of making the families of beads can be looked at as constructing and executing a program, a very common sort of program, in which two loops are nested: Fix a first color and run through all the possible second colors, then repeat until all possible first colors have been run through. For someone who is thoroughly used to computers and programming there is nothing "formal" or abstract about this task. For a child in a computer culture it would be as concrete as matching up knives and forks at the dinner table. Even the common "bug" of including some families twice (for example, red-blue and blue-red) would be well-known. Our culture is rich in pairs, couples, and one-to-one correspondences of all sorts, and it is rich in language for talking about such things. This richness provides both the incentive and a supply of models and tools for children to build ways to think about such issues as whether three large pieces of candy are more or less than four much smaller pieces. For such problems our children acquire an excellent intuitive sense of quantity. But our culture is relatively poor in models of systematic procedures. Until recently there was not even a name in popular language for programming, let alone for the ideas needed to do so successfully. There is no word for "nested loops" and no word for the double-counting bug. Indeed, there are no words for the powerful ideas computerists refer to as "bug" and "debugging."

Without the incentive or the materials to build powerful, concrete ways to think about problems involving systematicity, children are forced to approach such problems in a groping, abstract fashion. Thus cultural factors that are common to both the American city and the African village can explain the difference in age at which children build their intuitive knowledge of quantity and of systematicity.

While still working in Geneva I had become sensitive to the way in which materials from the then very young

computer cultures were allowing psychologists to develop new ways to think about thinking.<sup>1</sup> In fact, my entry into the world of computers was motivated largely by the idea that children could also benefit, perhaps even more than the psychologists, from the way in which computer models seemed able to give concrete form to areas of knowledge that had previously appeared so intangible and abstract.

I began to see how children who had learned to program computers could use very concrete computer models to think about thinking and to learn about learning and in doing so, enhance their powers as psychologists and as epistemologists. For example, many children are held back in their learning because they have a model of learning in which you have either “got it” or “got it wrong.” But when you learn to program a computer you almost never get it right the first time. Learning to be a master programmer is learning to become highly skilled at isolating and correcting “bugs,” the parts that keep the program from working. The question to ask about the program is not whether it is right or wrong, but if it is fixable. If this way of looking at intellectual products were generalized to how the larger culture thinks about knowledge and its acquisition, we all might be less intimidated by our fears of “being wrong.” This potential influence of the computer on changing our notion of a black and white version of our successes and failures is an example of using the computer as an “object-to-think-with.” It is obviously not necessary to work with computers in order to acquire good strategies for learning. Surely “debugging” strategies were developed by successful learners long before computers existed. But thinking about learning by analogy with developing a program is a powerful and accessible way to get started on becoming more articulate about one’s debugging strategies and more deliberate about improving them.

My discussion of a computer culture and its impact on thinking presupposes a massive penetration of powerful computers into people’s lives. That this will happen there can be no doubt. The calculator, the electronic game, and the digital watch were brought to us by a technical revolution that rapidly lowered prices for electronics in a period when all others were rising with inflation. That same technological revolution, brought about by the integrated circuit, is now bringing us the personal computer. Large computers used to cost millions of dollars because they were assembled out of millions of physically distinct parts. In the new technology a

complex circuit is not assembled but made as a whole, solid entity—hence the term “integrated circuit.” The effect of integrated circuit technology on cost can be understood by comparing it to printing. The main expenditure in making a book occurs long before the press begins to roll. It goes into writing, editing, and typesetting. Other costs occur after the printing: binding, distributing, and marketing. The actual cost per copy for printing itself is negligible. And the same is true for a powerful as for a trivial book. So, too, most of the cost of an integrated circuit goes into a preparatory process; the actual cost of making an individual circuit becomes negligible, provided enough are sold to spread the costs of development. The consequences of this technology for the cost of computation are dramatic. Computers that would have cost hundreds of thousands in the 1960s and tens of thousands in the early 1970s can now be made for less than a dollar. The only limiting factor is whether the particular circuit can fit onto what corresponds to a “page”—that is to say the “silicon chips” on which the circuits are etched.

But each year in a regular and predictable fashion the art of etching circuits on silicon chips is becoming more refined. More and more complex circuitry can be squeezed onto a chip, and the computer power that can be produced for less than a dollar increases. I predict that long before the end of the century, people will buy children toys with as much computer power as the great IBM computers currently selling for millions of dollars. And as for computers to be used as such, the main cost of these machines will be the peripheral devices, such as the keyboard. Even if these do not fall in price, it is likely that a supercomputer will be equivalent in price to a typewriter and a television set.

There really is no disagreement among experts that the cost of computers will fall to a level where they will enter everyday life in vast numbers. Some will be there as computers proper, that is to say, programmable machines. Others might appear as games of ever-increasing complexity and in automated supermarkets where the shelves, maybe even the cans, will talk. One really can afford to let one’s imagination run wild. There is no doubt that the material surface of life will become very different for everyone, perhaps most of all for children. But there has been significant difference of opinion about the effects this computer presence will produce. I would distinguish my thinking from two trends of thinking which I refer to here as the “skeptical” and the “critical.”

Skeptics do not expect the computer presence to make much difference in how people learn and think. I have formulated a number of possible explanations for why they think as they do. In some cases I think the skeptics might conceive of education and the effect of computers on it too narrowly. Instead of considering general cultural effects, they focus attention on the use of the computer as a device for programmed instruction. Skeptics then conclude that while the computer might produce some improvements in school learning, it is not likely to lead to fundamental change. In a sense, too, I think the skeptical view derives from a failure to appreciate just how much Piagetian learning takes place as a child grows up. If a person conceives of children's intellectual development (or, for that matter, moral or social development) as deriving chiefly from deliberate teaching, then such a person would be likely to underestimate the potential effect that a massive presence of computers and other interactive objects might have on children.

The critics,<sup>2</sup> on the other hand, do think that the computer presence will make a difference and are apprehensive. For example, they fear that more communication via computers might lead to less human association and result in social fragmentation. As knowing how to use a computer becomes increasingly necessary to effective social and economic participation, the position of the underprivileged could worsen, and the computer could exacerbate existing class distinctions. As to the political effect computers will have, the critics' concerns resonate with Orwellian images of a 1984 where home computers will form part of a complex system of surveillance and thought control. Critics also draw attention to potential mental health hazards of computer penetration. Some of these hazards are magnified forms of problems already worrying many observers of contemporary life; others are problems of an essentially new kind. A typical example of the former kind is that our grave ignorance of the psychological impact of television becomes even more serious when we contemplate an epoch of super TV. The holding power and the psychological impact of the television show could be increased by the computer in at least two ways. The content might be varied to suit the tastes of each individual viewer, and the show might become interactive, drawing the "viewer" into the action. Such things belong to the future, but people who are worried about the impact of the computer on people already cite cases of students spending sleepless nights riveted to the computer terminal, coming to neglect

both studies and social contact. Some parents have been reminded of these stories when they observe a special quality of fascination in their own children's reaction to playing with the still rudimentary electronic games.

In the category of problems that are new rather than aggravated versions of old ones, critics have pointed to the influence of the allegedly mechanized thought processes of computers on how people think. Marshall McLuhan's dictum that "the medium is the message" might apply here: If the medium is an interactive system that takes in words and speaks back like a person, it is easy to get the message that machines are like people and that people are like machines. What this might do to the development of values and self-image in growing children is hard to assess. But it is not hard to see reasons for worry.

Despite these concerns I am essentially optimistic—some might say utopian—about the effect of computers on society. I do not dismiss the arguments of the critics. On the contrary, I too see the computer presence as a potent influence on the human mind. I am very much aware of the holding power of an interactive computer and of how taking the computer as a model can influence the way we think about ourselves. In fact the work on LOGO to which I have devoted much of the past ten years consists precisely of developing such forces in positive directions. For example, the critic is horrified at the thought of a child hypnotically held by a futuristic, computerized super-pinball machine. In the LOGO work we have invented versions of such machines in which powerful ideas from physics or mathematics or linguistics are embedded in a way that permits the player to learn them in a natural fashion, analogous to how a child learns to speak. The computer's "holding power," so feared by critics, becomes a useful educational tool. Or take another, more profound example. The critic is afraid that children will adopt the computer as model and eventually come to "think mechanically" themselves. Following the opposite tack, I have invented ways to take educational advantage of the opportunities to master the art of *deliberately* thinking like a computer, according, for example, to the stereotype of a computer program that proceeds in a step-by-step, literal, mechanical fashion. There are situations where this style of thinking is appropriate and useful. Some children's difficulties in learning formal subjects such as grammar or mathematics derive from their inability to see the point of such a style.

A second educational advantage is indirect but ultimately more important. By deliberately learning to imitate mechanical thinking, the learner becomes able to articulate what mechanical thinking is and what it is not. The exercise can lead to greater confidence about the ability to choose a cognitive style that suits the problem. Analysis of “mechanical thinking” and how it is different from other kinds and practice with problem analysis can result in a new degree of intellectual sophistication. By providing a very concrete, down-to-earth model of a particular style of thinking, work with the computer can make it easier to understand that there is such a thing as a “style of thinking.” And giving children the opportunity to choose one style or another provides an opportunity to develop the skill necessary to choose between styles. Thus instead of inducing mechanical thinking, contact with computers could turn out to be the best conceivable antidote to it. And for me what is most important in this is that through these experiences these children would be serving their apprenticeships as epistemologists, that is to say learning to think articulately about thinking.

The intellectual environments offered to children by today’s cultures are poor in opportunities to bring their thinking about thinking into the open, to learn to talk about it and to test their ideas by externalizing them. Access to computers can dramatically change this situation. Even the simplest Turtle work can open new opportunities for sharpening one’s thinking about thinking: Programming the Turtle starts by making one reflect on how one does oneself what one would like the Turtle to do. Thus teaching the Turtle to act or to “think” can lead one to reflect on one’s own actions and thinking. And as children move on, they program the computer to make more complex decisions and find themselves engaged in reflecting on more complex aspects of their own thinking.

In short, while the critic and I share the belief that working with computers can have a powerful influence on how people think, I have turned my attention to exploring how this influence could be turned in positive directions.

I see two kinds of counterarguments to my arguments against the critics. The first kind challenges my belief that it is a good thing for children to be epistemologists. Many people will argue that overly analytic, verbalized thinking is counterproductive even if it is deliberately chosen. The second kind of objection challenges my suggestion that com-

puters are likely to lead to more reflective self-conscious thinking. Many people will argue that work with computers usually has the opposite effect. These two kinds of objections call for different kinds of analysis and cannot be discussed simultaneously. The first kind raises technical questions about the psychology of learning. The second kind of objection is most directly answered by saying that there is absolutely no inevitability that computers will have the effects I hope to see. Not all computer systems do. Most in use today do not. In LOGO environments I have seen children engaged in animated conversations about their own personal knowledge as they try to capture it in a program to make a Turtle carry out an action that they themselves know very well how to do. But of course the physical presence of a computer is not enough to insure that such conversations will come about. Far from it. In thousands of schools and in tens of thousands of private homes children are right now living through very different computer experiences. In most cases the computer is being used either as a versatile video game or as a “teaching machine” programmed to put children through their paces in arithmetic or spelling. And even when children are taught by a parent, a peer, or a professional teacher to write simple programs in a language like BASIC, this activity is not accompanied at all by the kind of epistemological reflection that we see in the LOGO environments. So I share a skepticism with the critics about what is being done with computation now. But I am interested in stimulating a major change in how things can be. The bottom line for such changes is political. What is happening now is an empirical question. What can happen is a technical question. But what will happen is a political question, depending on social choices.

The central open questions about the effect of computers on children in the 1980s are these: Which people will be attracted to the world of computers, what talents will they bring, and what tastes and ideologies will they impose on the growing computer culture? I have described children in LOGO environments engaged in self-referential discussions about their own thinking. This could happen because the LOGO language and the Turtle were designed by people who enjoy such discussion and worked hard to design a medium that would encourage it. Other designers of computer systems have different tastes and different ideas about what kinds of activities are suitable for children. Which design will

prevail, and in what sub-culture, will not be decided by a simple bureaucratic decision made, for example, in a government Department of Education or by a committee of experts. Trends in computer style will emerge from a complex web of decisions by Foundations with resources to support one or another design, by corporations who may see a market, by schools, by individuals who will decide to make their career in the new field of activity, and by children who will have their own say in what they pick up and what they make of it. People often ask whether in the future children will program computers or become absorbed in pre-programmed activities. The answer must be that some children will do the one, some the other, some both and some neither. But which children, and most importantly, which social classes of children, will fall into each category will be influenced by the kind of computer activities and the kind of environments created around them.

As an example, we consider an activity which may not occur to most people when they think of computers and children: the use of a computer as a writing instrument. For me, writing means making a rough draft and refining it over a considerable period of time. My image of myself as a writer includes the expectation of an "unacceptable" first draft that will develop with successive editing into presentable form. But I would not be able to afford this image if I were a third grader. The physical act of writing would be slow and laborious. I would have no secretary. For most children rewriting a text is so laborious that the first draft is the final copy, and the skill of rereading with a critical eye is never acquired. This changes dramatically when children have access to computers capable of manipulating text. The first draft is composed at the keyboard. Corrections are made easily. The current copy is always neat and tidy. I have seen a child move from total rejection of writing to an intense involvement (accompanied by rapid improvement of quality) within a few weeks of beginning to write with a computer. Even more dramatic changes are seen when the child has physical handicaps that make writing by hand more than usually difficult or even impossible.

This use of computers is rapidly becoming adopted wherever adults write for a living. Most newspapers now provide their staff with "word processing" computer systems. Many writers who work at home are acquiring their own computers, and the computer terminal is steadily displacing the typewriter as the secretary's basic tool. The image of

children using the computer as a writing instrument is a particularly good example of my general thesis that what is good for professionals is good for children. But this image of how the computer might contribute to children's mastery of language is dramatically opposed to the one that is taking root in most elementary schools. There the computer is seen as a teaching instrument. It gives children practice in distinguishing between verbs and nouns, in spelling, and in answering multiple-choice questions about the meaning of pieces of text. As I see it, this difference is not a matter of a small and technical choice between two teaching strategies. It reflects a fundamental difference in educational philosophies. More to the point, it reflects a difference in views on the nature of childhood. I believe that the computer as writing instrument offers children an opportunity to become more like adults, indeed like advanced professionals, in their relationship to their intellectual products and to themselves. In doing so, it comes into head-on collision with the many aspects of school whose effect, if not whose intention, is to "infantilize" the child.

Word processors *can* make a child's experience of writing more like that of a real writer. But this can be undermined if the adults surrounding that child fail to appreciate what it is like to be a writer. For example, it is only too easy to imagine adults, including teachers, expressing the view that editing and re-editing a text is a waste of time ("Why don't you get on to something new?" or "You aren't making it any better, why don't you fix your spelling?").

As with writing, so with music-making, games of skill, complex graphics, whatever: The computer is not a culture unto itself but it can serve to advance very different cultural and philosophical outlooks. For example, one could think of the Turtle as a device to teach elements of the traditional curriculum, such as notions of angle, shape, and coordinate systems. And in fact, most teachers who consult me about its use are, quite understandably, trying to use it in this way. Their questions are about classroom organization, scheduling problems, pedagogical issues raised by the Turtle's introduction, and especially, about how it relates conceptually to the rest of the curriculum. Of course the Turtle can help in the teaching of traditional curriculum, but I have thought of it as a vehicle for Piagetian learning, which to me is learning without curriculum.

There are those who think about creating a "Piagetian curriculum" or "Piagetian teaching methods." But to my

mind these phrases and the activities they represent are contradictions in terms. I see Piaget as the theorist of learning without curriculum and the theorist of the kind of learning that happens without deliberate teaching. To turn him into the theorist of a new curriculum is to stand him on his head.

But “teaching without curriculum” does not mean spontaneous, free-form classrooms or simply “leaving the child alone.” It means supporting children as they build their own intellectual structures with materials drawn from the surrounding culture. In this model, educational intervention means changing the culture, planting new constructive elements in it and eliminating noxious ones. This is a more ambitious undertaking than introducing a curriculum change, but one which is feasible under conditions now emerging.

Suppose that thirty years ago an educator had decided that the way to solve the problem of mathematics education was to arrange for a significant fraction of the population to become fluent in (and enthusiastic about) a new mathematical language. The idea might have been good in principle, but in practice it would have been absurd. No one had the power to implement it. Now things are different. Many millions of people are learning programming languages for reasons that have nothing to do with the education of children. Therefore, it becomes a practical proposition to influence the form of the languages they learn and the likelihood that their children will pick up these languages.

The educator must be an anthropologist. The educator as anthropologist must work to understand which cultural materials are relevant to intellectual development. Then, he or she needs to understand which trends are taking place in the culture. Meaningful intervention must take the form of working with these trends. In my role of educator as anthropologist I see new needs being generated by the penetration of the computer into personal lives. People who have computers at home or who use them at work will want to be able to talk about them to their children. They will want to be able to teach their children to use the machines. Thus there could be a cultural demand for something like Turtle graphics in a way there never was, and perhaps never could be, a cultural demand for the New Math.

Throughout the course of this chapter I have been talking about the ways in which choices made by educators, foundations, governments, and private individuals can affect the potentially revolutionary changes in how children learn.

But making good choices is not always easy, in part because past choices can often haunt us. There is a tendency for the first usable, but still primitive, product of a new technology to dig itself in. I have called this phenomenon the QWERTY phenomenon.

The top row of alphabetic keys of the standard typewriter reads QWERTY. For me this symbolizes the way in which technology can all too often serve not as a force for progress but for keeping things stuck. The QWERTY arrangement has no rational explanation, only a historical one. It was introduced in response to a problem in the early days of the typewriter: The keys used to jam. The idea was to minimize the collision problem by separating those keys that followed one another frequently. Just a few years later, general improvements in the technology removed the jamming problem, but QWERTY stuck. Once adopted, it resulted in many millions of typewriters and a method (indeed a full-blown curriculum) for learning typing. The social cost of change (for example, putting the most used keys *together* on the keyboard) mounted with the vested interest created by the fact that so many fingers now knew how to follow the QWERTY keyboard. QWERTY has stayed on despite the existence of other, more “rational” systems. On the other hand, if you talk to people about the QWERTY arrangement they will justify it by “objective” criteria. They will tell you that it “optimizes this” or it “minimizes that.” Although these justifications have no rational foundation, they illustrate a process, a social process, of myth construction that allows us to build a justification for primitivity into any system. And I think that we are well on the road to doing exactly the same thing with the computer. We are in the process of digging ourselves into an anachronism by preserving practices that have no rational basis beyond their historical roots in an earlier period of technological and theoretical development.

The use of computers for drill and practice is only one example of the QWERTY phenomenon in the computer domain. Another example occurs even when attempts are made to allow students to learn to program the computer. As we shall see in later chapters, learning to program a computer involves learning a “programming language.” There are many such languages—for example, FORTRAN, PASCAL, BASIC, SMALLTALK, and LISP, and the lesser known language LOGO, which our group has used in most of our experiments with computers and children. A powerful QWERTY phenomenon is to be expected when we

choose the language in which children are to learn to program computers. I shall argue in detail that the issue is consequential. A programming language is like a natural, human language in that it favors certain metaphors, images, and ways of thinking. The language used strongly colors the computer culture. It would seem to follow that educators interested in using computers and sensitive to cultural influences would pay particular attention to the choice of language. But nothing of the sort has happened. On the contrary, educators, too timid in technological matters or too ignorant to attempt to influence the languages offered by computer manufacturers, have accepted certain programming languages in much the same way as they accepted the QWERTY keyboard. An informative example is the way in which the programming language BASIC<sup>3</sup> has established itself as the obvious language to use in teaching American children how to program computers. The relevant technical information is this: A very small computer can be made to understand BASIC, while other languages demand more from the computer. Thus, in the early days when computer power was extremely expensive, there was a genuine technical reason for the use of BASIC, particularly in schools where budgets were always tight. Today, and in fact for several years now, the cost of computer memory has fallen to the point where any remaining economic advantages of using BASIC are insignificant. Yet in most high schools, the language remains almost synonymous with programming, despite the existence of other computer languages that are demonstrably easier to learn and are richer in the intellectual benefits that can come from learning them. The situation is paradoxical. The computer revolution has scarcely begun, but is already breeding its own conservatism. Looking more closely at BASIC provides a window on how a conservative social system appropriates and tries to neutralize a potentially revolutionary instrument.

BASIC is to computation what QWERTY is to typing. Many teachers have learned BASIC, many books have been written about it, many computers have been built in such a way that BASIC is “hardwired” into them. In the case of the typewriter, we noted how people invent “rationalizations” to justify the status quo. In the case of BASIC, the phenomenon has gone much further, to the point where it resembles ideology formation. Complex arguments are invented to justify features of BASIC that were originally included

because the primitive technology demanded them or because alternatives were not well enough known at the time the language was designed.

An example of BASIC ideology is the argument that BASIC is easy to learn because it has a very small vocabulary. The surface validity of the argument is immediately called into question if we apply it to the context of how children learn natural languages. Imagine a suggestion that we invent a special language to help children learn to speak. This language would have a small vocabulary of just fifty words, but fifty words so well chosen that all ideas could be expressed using them. Would this language be easier to learn? Perhaps the vocabulary might be easy to learn, but the use of the vocabulary to express what one wanted to say would be so contorted that only the most motivated and brilliant children would learn to say more than “hi.” This is close to the situation with BASIC. Its small vocabulary can be learned quickly enough. But using it is a different matter. Programs in BASIC acquire so labyrinthine a structure that in fact only the most motivated and brilliant (“mathematical”) children do learn to use it for more than trivial ends.

One might ask why the teachers do not notice the difficulty children have in learning BASIC. The answer is simple: Most teachers do not expect high performance from most students, especially in a domain of work that appears to be as “mathematical” and “formal” as programming. Thus the culture’s general perception of mathematics as inaccessible bolsters the maintenance of BASIC, which in turn confirms these perceptions. Moreover, the teachers are not the only people whose assumptions and prejudices feed into the circuit that perpetuates BASIC. There are also the computerists, the people in the computer world who make decisions about what languages their computers will speak. These people, generally engineers, find BASIC quite easy to learn, partly because they are accustomed to learning such very technical systems and partly because BASIC’s sort of simplicity appeals to their system of values. Thus, a particular subculture, one dominated by computer engineers, is influencing the world of education to favor those school students who are most like that subculture. The process is tacit, unintentional: It has never been publicly articulated, let alone evaluated. In all of these ways, the social embedding of BASIC has far more serious consequences than the “digging in” of QWERTY.

There are many other ways in which the attributes of the subcultures involved with computers are being projected onto the world of education. For example, the idea of the computer as an instrument for drill and practice that appeals to teachers because it resembles traditional teaching methods also appeals to the engineers who design computer systems: Drill and practice applications are predictable, simple to describe, efficient in use of the machine's resources. So the best engineering talent goes into the development of computer systems that are biased to favor this kind of application. The bias operates subtly. The machine designers do not actually decide what will be done in the classrooms. That is done by teachers and occasionally even by carefully controlled comparative research experiments. But there is an irony in these controlled experiments. They are very good at telling whether the small effects seen in best scores are real or due to chance. But they have no way to measure the undoubtedly real (and probably more massive) effects of the biases built into the machines.

We have already noted that the conservative bias being built into the use of computers in education has also been built into other new technologies. The first use of the new technology is quite naturally to do in a slightly different way what had been done before without it. It took years before designers of automobiles accepted the idea that they were cars, not "horseless carriages," and the precursors of modern motion pictures were plays acted as if before a live audience but actually in front of a camera. A whole generation was needed for the new art of motion pictures to emerge as something quite different from a linear mix of theater plus photography. Most of what has been done up to now under the name of "educational technology" or "computers in education" is still at the stage of the linear mix of old instructional methods with new technologies. The topics I shall be discussing are some of the first probings toward a more organic interaction of fundamental educational principles and new methods for translating them into reality.

We are at a point in the history of education when radical change is possible, and the possibility for that change is directly tied to the impact of the computer. Today what is offered in the education "market" is largely determined by what is acceptable to a sluggish and conservative system. But this is where the computer presence is in the process of creating an environment for change. Consider the conditions under which a new educational idea can be put into practice today and in the near future. Let us suppose that today I have an idea about how children could learn mathematics more effectively and more humanely. And let us suppose that I have been able to persuade a million people that the idea is a good one. For many products such a potential market would guarantee success. Yet in the world of education today this would have little clout: A million people across the nation would still mean a minority in every town's school system, so there might be no effective channel for the million voices to be expressed. Thus, not only do good educational ideas sit on the shelves, but the process of invention is itself stymied. This inhibition of invention in turn influences the selection of people who get involved in education. Very few with the imagination, creativity, and drive to make great new inventions enter the field. Most of those who do are soon driven out in frustration. Conservatism in the world of education has become a self-perpetuating *social* phenomenon.

Fortunately, there is a weak link in the vicious circle. Increasingly, the computers of the very near future will be the private property of individuals, and this will gradually return to the individual the power to determine patterns of education. Education will become more of a private act, and people with good ideas, different ideas, exciting ideas will no longer be faced with a dilemma where they either have to "sell" their ideas to a conservative bureaucracy or shelve them. They will be able to offer them in an open marketplace directly to consumers. There will be new opportunities for imagination and originality. There might be a renaissance of thinking about education.

## Notes

1. The program FOLLOW\* is a very simple example of how a powerful cybernetic idea (control by negative feedback) can be used to elucidate a biological or psychological phenomenon. Simple as it is, the example helps bridge the gap between physical models of "causal mechanism" and psychological phenomenon such as "purpose."

Theoretical psychologists have used more complex programs in the same spirit to construct models of practically every known psychological phenomenon. A bold formulation of the spirit of such inquiry is found in Herbert A. Simon, *Sciences of the Artificial* (Cambridge: MIT Press, 1969).

2. The critics and skeptics referred to here are distillations from years of public and private debates. These attitudes are widely held, but, unfortunately, seldom published and therefore seldom discussed with any semblance of rigor. One critic who has set a good example by publishing his views is Joseph Weizenbaum in *Computer Power and Human Reason: From Judgement to Calculation* (San Francisco: W.H. Freeman, 1976).

Unfortunately Weizenbaum's book discusses two separate (though related) questions: whether computers harm the way people think and whether computers themselves can think. Most critical reviews of Weizenbaum have focused on the latter question, on which he joins company with Hubert L. Drefus, *What Computers Can't Do: A Critique of Artificial Reason* (New York: Harper & Row, 1972).

A lively description of some of the principal participants in the debate about whether computers can or cannot think is found in Pamela McCorduck, *Machines Who Think* (San Francisco: W.H. Freeman, 1979).

There is little published data on whether computers actually affect how people think. This question is being studied presently by S. Turkle.

3. Many versions of BASIC would allow a program to produce a shape like that made by the LOGO program HOUSE. The simplest example would look something like this:

```
10 PLOT (0,0)
20 PLOT (100,0)
30 PLOT (100,100)
40 PLOT (75,150)
50 PLOT (0,100)
60 PLOT (0,0)
70 END
```

Writing such a program falls short of the LOGO program as a beginning programming experience in many ways. It demands knowledge of cartesian coordinates. This demand would be less serious if the program, once written, could become a powerful tool for other projects. The LOGO programs SQ, TRI, and HOUSE can be used to draw squares, triangles, and houses in any position and orientation on the screen. The BASIC program allows once

particular house to be drawn in one position. In order to make a BASIC program that will draw houses in many positions, it is necessary to use algebraic variables as in PLOT (x, y), PLOT (x + 100, y), and so on. As for defining new commands, such as SQ, TRI, and HOUSE, the commonly used versions of BASIC either do not allow this at all or, at best, allow something akin to it to be achieved through the use of advanced technical programming methods. Advocates of BASIC might reply that: (1) these objections refer only to a beginner's experience and (2) these deficiencies of BASIC could be fixed. The first argument is simply not true: The intellectual and practical primitivity of BASIC extends all along the line up to the most advanced programming. The second misses the point of my complaint. Of course one could turn BASIC into LOGO or SMALLTALK or anything else and still call it "BASIC." My complaint is that what is being foisted on the education world has not been so "fixed." Moreover, doing so would be a little like "remodelling" a wooden house to become a skyscraper.

\* *Touch Sensor Turtle*. The simplest touch sensor program in LOGO is as follows:

TO BOUNCE	Comments
REPEAT	This means repeat all the individual steps
FORWARD 1	The turtle keeps moving
TEST FRONT.TOUCH	It checks whether it has run into something
IFTRUE RIGHT 180	If so, it does an about turn

END

This will make the Turtle turn about when it encounters an object. A more subtle and more instructive program using the Touch Sensor Turtle is as follows:

TO FOLLOW	Comments
REPEAT	
FORWARD 1	
TEST LEFT.TOUCH	Check: Is it touching?
IFTRUE RIGHT 1	It thinks it's too close and turns away
IFFALSE LEFT 1	It thinks it might lose the object so it turns toward

END

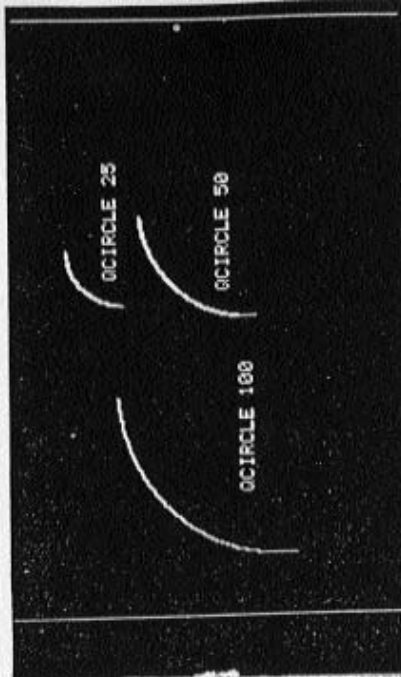
This program will cause the Turtle to circumnavigate an object of any shape, provided that it starts with its left side in contact with the object (and provided that the object and any irregularities in its contour are large compared with the Turtle).

It is a very instructive project for any group of students to develop this (or an equivalent) program from first principles by acting out how they think they would use touch to get around an object and by translating their strategies into Turtle commands.

[MIT PRESS:-----  
[-----DO NOT KEY THIS ARTICLE IN-----  
[This article will be included in facsimilie.]

**FIND RESOURCES**

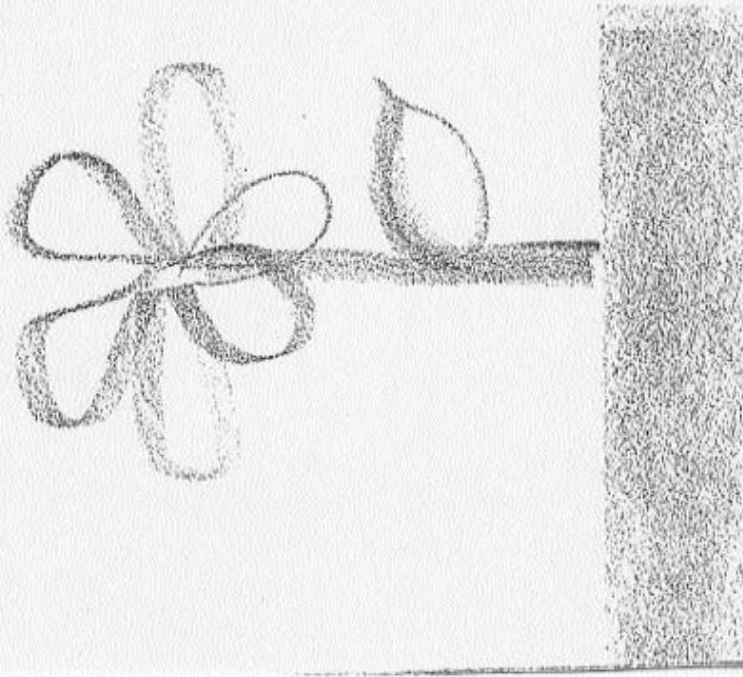
- Do you have any programs we could use?
- Yes, there's that quarter circle thing I made last week
- Show me.



- It draws quarter circles starting wherever the turtle
- It needs an input to tell it how big.

**A PLAN**

— Let's make the computer draw a flower like this.



**FIX THE BUG**

- We have to turn the Turtle between QCIRCLES.
- Try 120°.
- OK, that worked for triangles.
- And let's hide the Turtle by typing HIDE TURTLE.

**TRY SOMETHING**

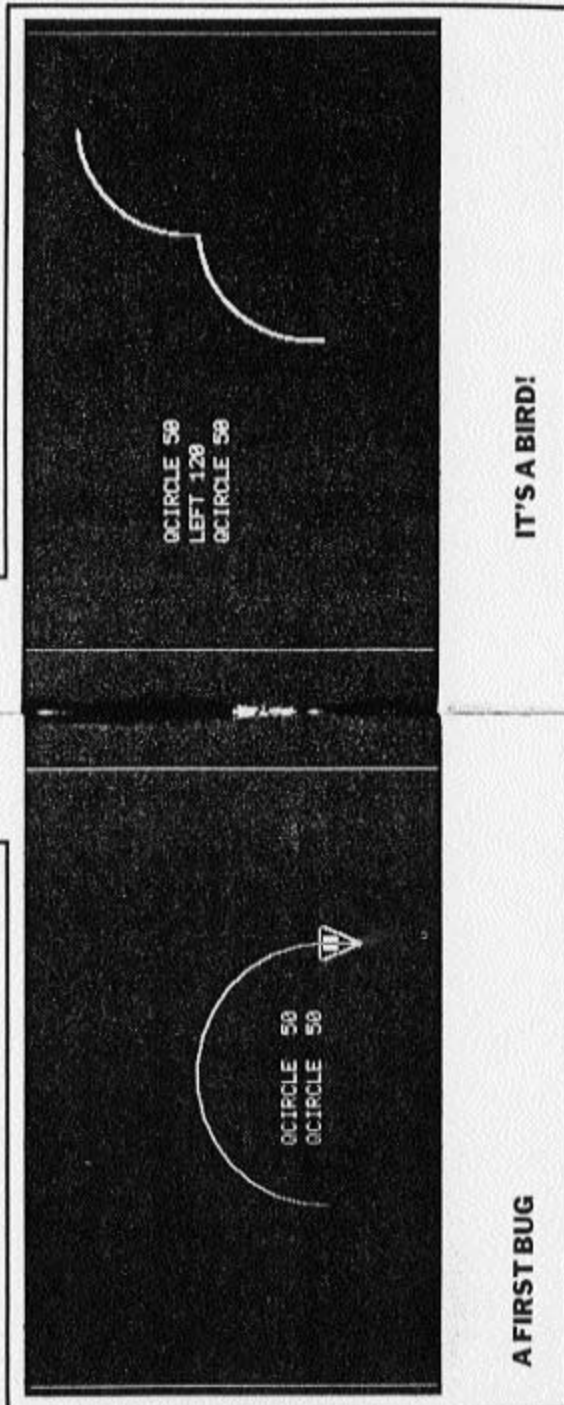
- Let's make a petal by putting two QCIRCLES together.
- OK. What size?
- How about 50?

**IT'S A BIRD!**

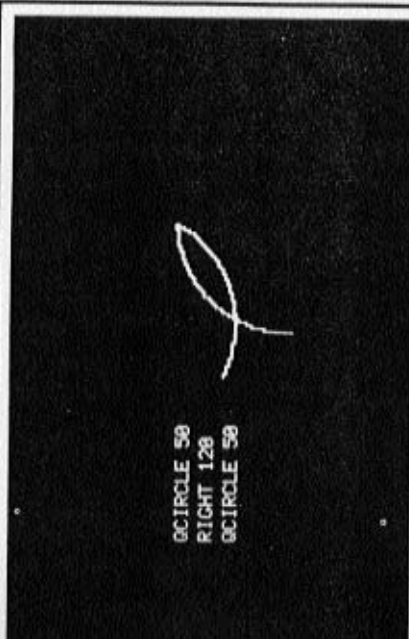
- What's going on?
- Try a right turn.

**A FIRST BUG**

- It didn't work.
- Of course! Two QCIRCLES make a semicircle.



- Why don't we just stick with the bird? We could make a flock.
- You do that. I want my flower.
- We could do the flower, then the flock.



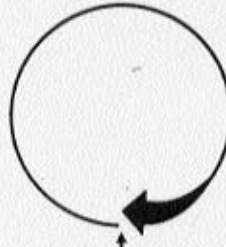
QCIRCLE 50  
RIGHT 120  
QCIRCLE 50

**IT'S A FISH!**

- The right turn is better.
- But we don't know how much to turn.
- We could try some more numbers.
- Or we could try some mathematics.

**MATH TO THE RESCUE**

—Do you know about the Total Turtle Trip Theorem? You think about the Turtle going all around the petal and add up all the turns. 360°.



- All around is 360.
- Each QCIRCLE turns it 90. That makes 180 for two QCIRCLES.
- 360 altogether. Take away 180 for the QCIRCLES. That leaves 180 for the pointy parts. 90 each.
- So we should do RIGHT 90 at each point.
- Let's try.



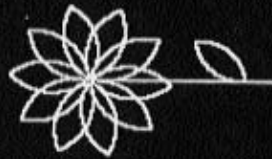
**An image will be  
inserted here**

- If we let PETAL have an input we can make big or small flowers.
- That's easy. Just do TO PETAL SIZE
- QCIRCLE SIZE, and so on.
- But I bet we'd get bugs if we try that. Let's try plain 25 first.
- Then we can make a superprocedure to draw a plant.

**BUILDING UP**

```

TO PLANT
NEWFLOWER
BRCK 50
PETAL
BRCK 50
END
    
```

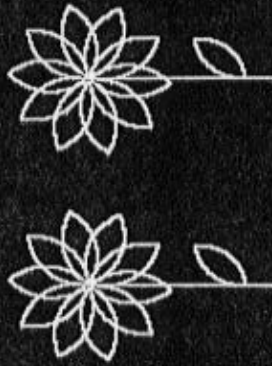


**ENDS BECOME MEANS**

—I have a great procedure for putting several together. It's called SLIDE.  
 You just go, PLANT SLIDE PLANT SLIDE PLANT SLIDE.

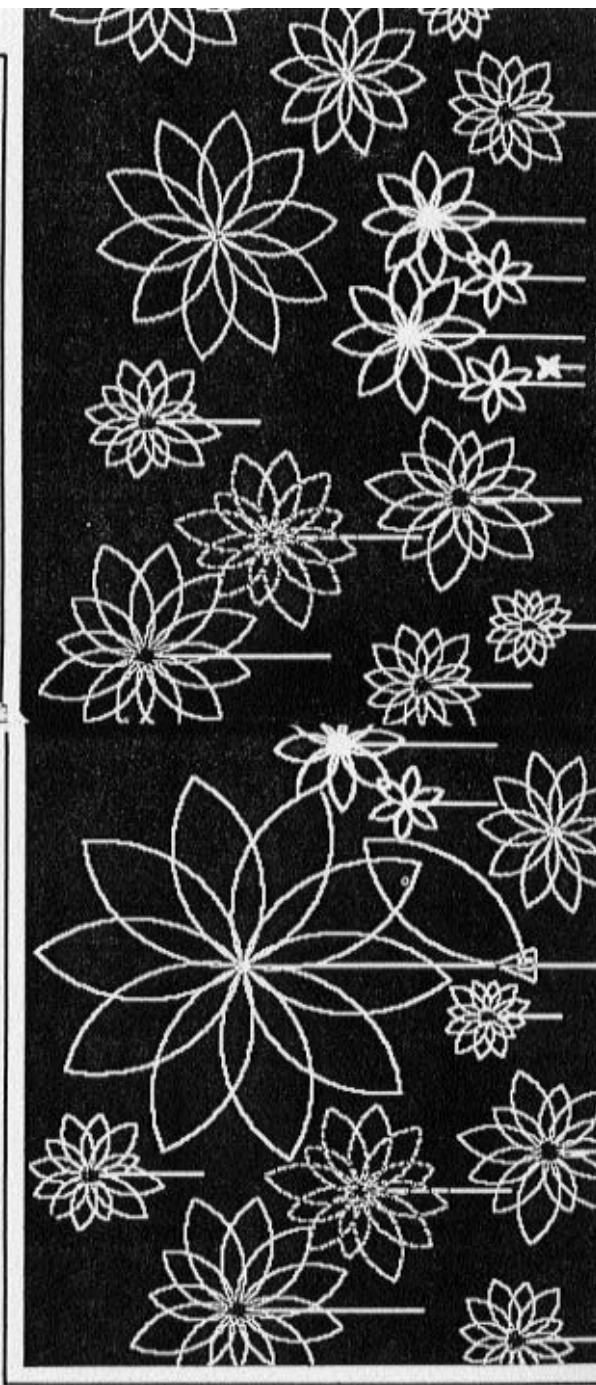
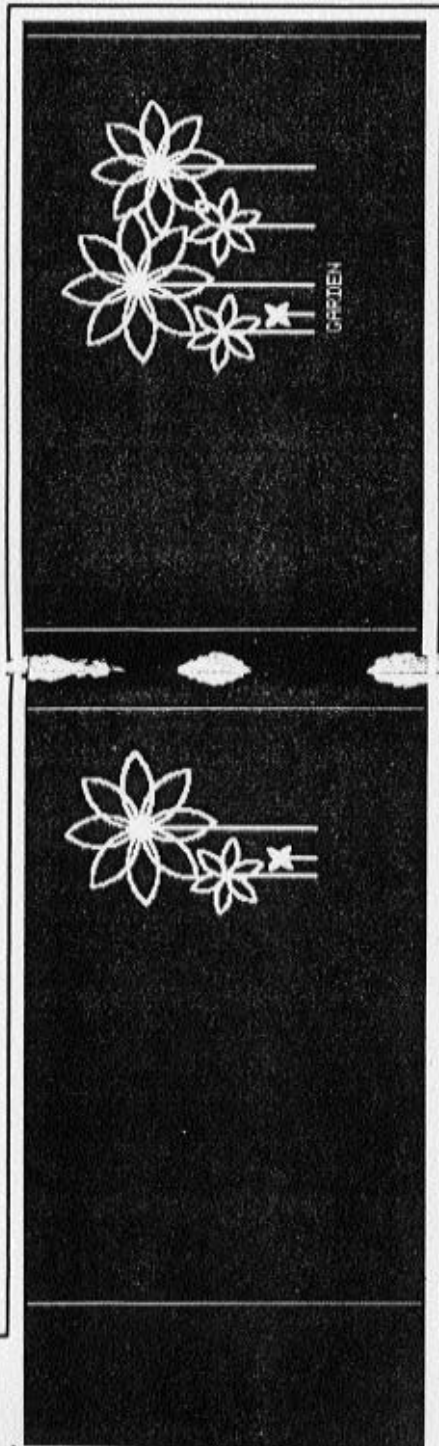


**TRYING THE NEW TOOL**



— My next project is a flock of birds.  
— Maybe we'll put the birds and flowers together.  
— Maybe.

— It would be better with small ones and big ones.  
— So, change the procedure to accept inputs.  
— And if we use RANDOM we can make a garden.



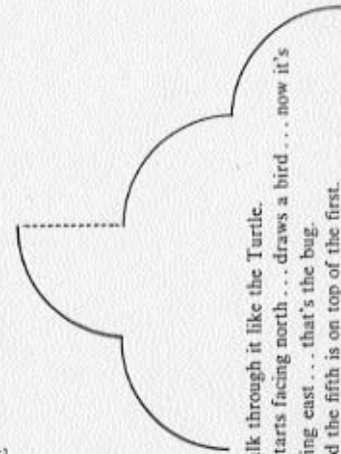
**SERENDIPITY**

- Make a flock by doing BIRD SLIDE BIRD SLIDE.
- I want six birds, and I'm going to use REPEAT.



- That's funny. I wanted 6 birds all the same way up.
- But it's neat. If we debug it, we should keep a copy like this.

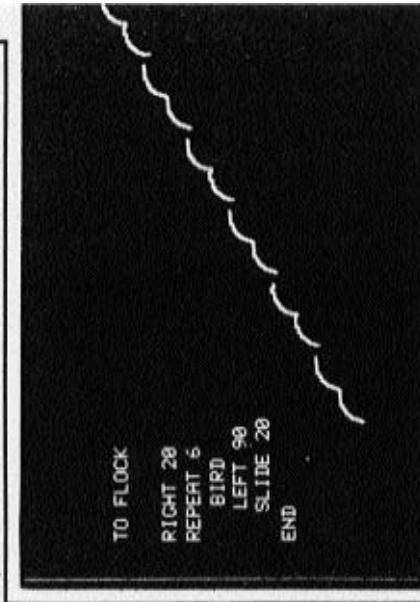
BIRD  
SLIDE  
BIRD



- Walk through it like the Turtle.
- It starts facing north... draws a bird... now it's facing east... that's the bug.
- And the fifth is on top of the first.

**DEBUGGING**

- If you want to fix the bug, bring the Turtle around to face north after doing the bird.
- And let's make them smaller.



— Here's the flock.

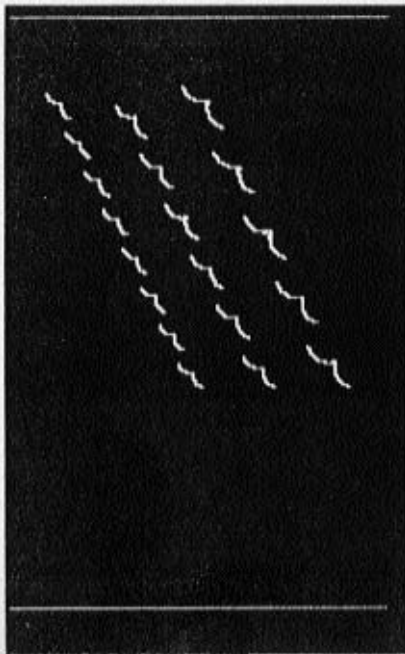
**THE END**

AI

**... A BEGINNING**

The next phase of the project will produce the most spectacular effects as the birds go into motion. But the printed page cannot capture either the product or the process: the serendipitous discoveries, the bugs, and the mathematical insights all require movement to be appreciated. Reflecting on what you are missing leads me to another description of something new the computer offers a child: the opportunity to draw in motion, indeed to doodle and even to scribble with movement as well as with lines. Perhaps they will be learning, as they do so, to think more dynamically.

—It's not finished. Let's give the flock inputs and put several together.



—How can we make them fly?  
—I found something neat. In BIRD use SPIN instead of RIGHT ... it's got bugs, but it is pretty.

