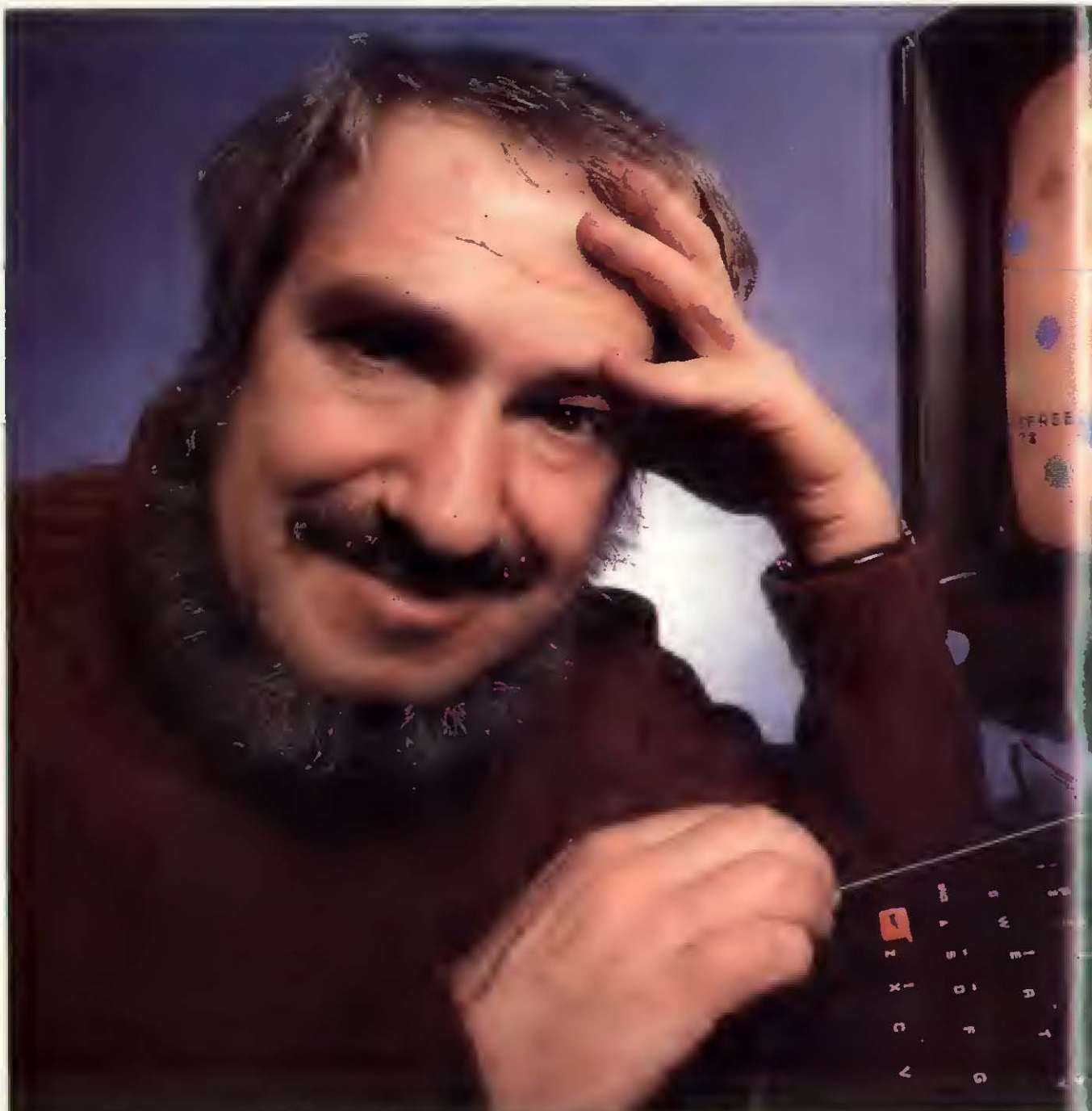


SEYMOUR PAPERT—*technological seer, outspoken social and educational critic, self-styled computer poet —exerts a profound influence in the microcomputer field. One journalist recently called him "the spiritual father of the Logo movement," and referred to his book,* Mindstorms *(Basic Books, New York, 1980), as "the movement's bible." He is best known for inventing Logo, a versatile computer language that provides a friendly introduction to programming, a serious tool for advanced programmers, and a medium for educational discovery.*

Papert says his goal was to create a language "simple enough so a five-year-old could write a program in the first few minutes of contact with a computer, and sophisticated enough so a computer scientist would find the system congenial and rich."

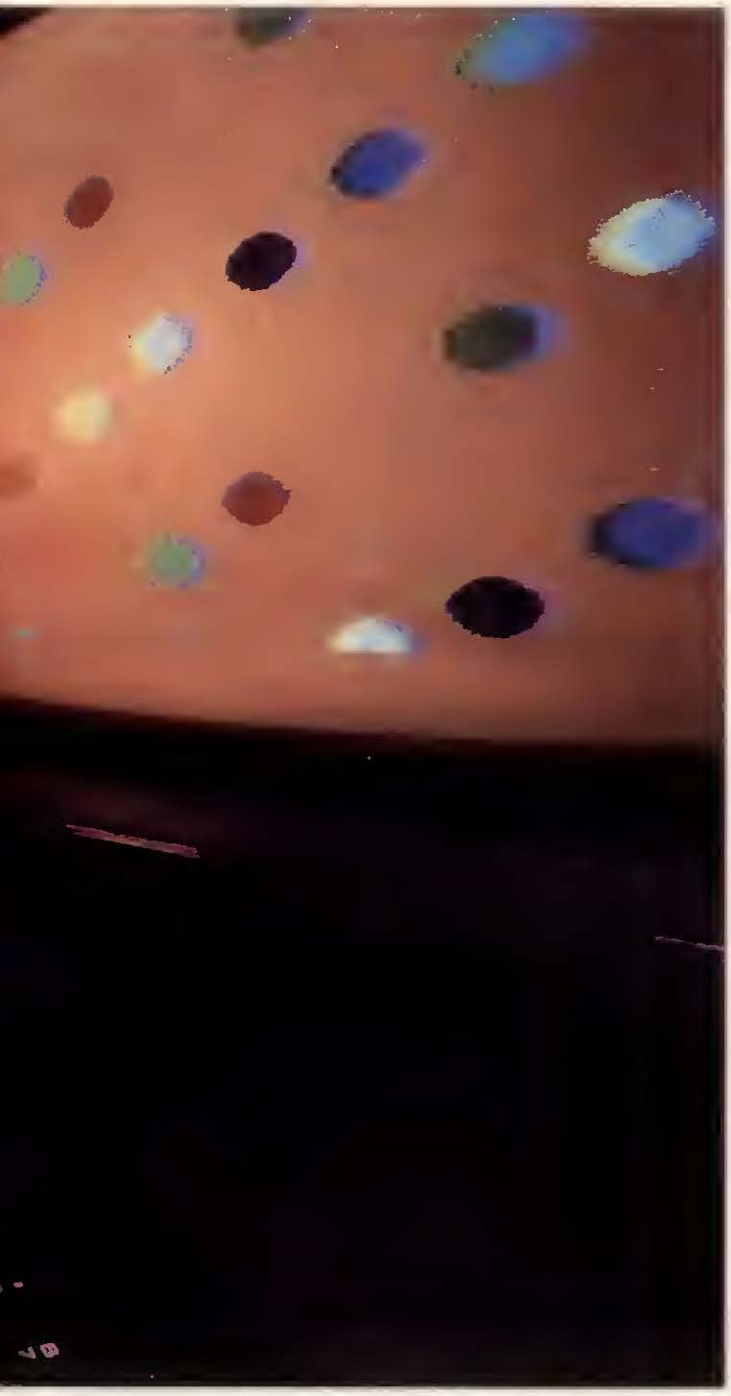
The resulting language might have been created in Papert's own image. It is friendly, accessible and as complex as you want to make it. But, like any language, Logo is difficult to describe in a few words. "Because it can take on a thousand forms and can serve a thousand functions," Papert says, "it can appeal to a thousand tastes."

Papert's unaffected clothing and meditative air bespeak his position as MIT professor and programming guru for the current generation of computer educators. In personal conversation, Papert is reflective and almost otherworldly, but charming. He delivers carefully considered statements in a South African accent, a relic of his Johannesburg upbringing, tempered by years of study at Cambridge and the University of Paris.

Mathematical studies and years of work with renowned child psychologist Jean Piaget in Geneva prepared Papert for his work in the development of educational computing.

"Personal computers will soon become the most important medium for learning," Papert says, "because of the ways they affect how people think and learn." He suggests that computers are like pencils; you need more than one or two per class. He estimates that, at current prices, every student could have a personal computer for approximately five percent of the cost of

PHOTOGRAPHED BY TED HOROWITZ



Learning to LEARN *with the* FATHER OF LOGO

Seymour Papert on Education and Language

by

Carlos Vidal Greth

each student's total education.

"I dream of using this technology not to improve the schools we have always known, but to replace them with something better," Papert says. "It will be like the growth of a new culture in which computers will be so integrated into ways to think about ourselves and the matters we learn that the nature of learning itself will be transformed."

On a recent visit to Atari corporate headquarters in Sunnyvale, California, Logo inventor Seymour Papert shared his views about Logo and its revolutionary implications for our schools and society.

CONNECTION: Did you expect to develop such a large, devoted following when *Mindstorms* was published?

PAPERT: I wrote *Mindstorms* as a criticism of our educational system. I've been incredibly, pleasantly surprised by the number and diversity of people who have felt something personal in the book. People have found in it a validation of themselves in the face of an educational system that is a virtual put down. I wrote what people don't want to admit —even to themselves—about how they feel about our schools.

CONNECTION: In *Mindstorms* you compare Logo environments to Brazilian samba schools (social clubs whose members prepare music, dances, and skits for the annual parade during Carnival in Rio De Janeiro). Why do you make this analogy?

PAPERT: I've done the samba in Brazilian samba schools. Samba schools provide a context where people enjoy themselves while learning, and where experts and beginners interact. This never happens in American schools, where learners are segregated by levels of ability in classrooms. Samba schools are in the culture, a part of people's lives.

CONNECTION: Will there be a role for teachers in the future?

"With Logo, I use computers the way a poet uses words."

PAPERT: Absolutely. When people claim that my vision of education does away with teachers, I get very upset. That turns on its head what I've been trying to say. Schools give teachers very little opportunity to teach. They spend most of their time brainwashing or forcing children to do rote activities nobody believes in. In the kind of learning environment I envision, teachers can really teach.

CONNECTION: Can schools be transformed to accommodate the style of learning that you and others propose?
PAPERT: Twenty years from now, there will be very different learning environments involving computers. Schools will successfully transform themselves into these new environments or they'll die out like the dinosaurs.

CONNECTION: That's a disturbing concept, in part because so many people are involved in education in some way. Is it a case of program or perish?
PAPERT: The role of learning professionals in society is going to grow. Rapid technological changes mean that people will have to or want to do more learning. Educators will have to adapt to new contexts. If they can't teach themselves to teach in new contexts, they are pretty lousy teachers, and should be doing something else.

CONNECTION: In a symposium several years ago, you said that if you had $10 million, you would spend it on creating "conditions for the emergence of computer poets." What are computer poets?
PAPERT: A poet is someone who establishes a relationship with his readers that's outside the logical, discursive relationship we usually classify as science. With Logo, the computer can be used similarly, to touch on the deeper, non-logical dimensions of self and the personal aesthetic. As a programmer, I use the computer in the same way a poet uses words, to touch on intimate and individual aspects of life.

CONNECTION: Is BASIC bad for you?
PAPERT: BASIC is a bad language. BASIC exists for the same reason we use the QWERTY [the top row of alphabetic keys on a standard typewriter] keyboard. It was created for historical reasons. The only reason it continues to exist is that lots of people know it, and lots of programs have been written in it. If BASIC were wiped out by some kind of cosmic chemical, I'm quite sure it wouldn't be reinvented.

Logo will replace BASIC altogether or in a very large proportion of homes. Certainly, Logo will be the language home computer owners use for pleasurable programming. There are very few projects that most home computer owners are likely to want to do that couldn't be done better on Logo than on other available languages.

CONNECTION: Most Logo applications are geared toward youngsters. How can an adult use Logo?

PAPERT: We've created a wrong impression by de-emphasizing the value of Logo for adults. Logo is a particularly good introduction for adults who want to learn to program. Experience shows that it's easier to convert from Logo to other programming languages that it is to learn those languages directly. Logo is also good for business and record-keeping applications. Logo is being pushed by a number of people as the small business applications language of the future.

CONNECTION: Why are so many people surprised after experiencing Logo to find that computers can be so accessible?
PAPERT: The easy answer is that many people grew up thinking of themselves as nontechnological.

There are deeper explanations having to do with the

"A child might use dynaturtles to create pleasing patterns, but he applies the same laws that govern the paths of planets. Call it participatory physics."

"If BASIC were wiped out, it would not be reinvented."

subconscious. There are many fears coming to light in terms of our relationship to computers. These are fears about whether we are self-determining, free creatures, or whether we are machines. The computer forces us to face some very basic elements of human nature. Any confrontation with the depths of what one is has to be frightening.

CONNECTION: Why are the turtles valuable as learning tools?
PAPERT: Graphics is one of the easiest ways into the world of programming. If you are a beginner, your first program should probably be one that draws something on the video screen. If you're an advanced programmer and want to learn a new style of programming, getting the compurer to draw something for you is a good way to start. Graphics programs let you see immediately what your program is doing.

CONNECTION: How is turtle geometry different from other types of geometry?
PAPERT: It's process-oriented or dynamic geometry. In Cartesian geometry, one thinks of a circle as the set of poinrs that makes a cerrain equation come rrue. That equation defines the points. In turtle geometry, you define a circle by thinking about how the turtle moves on the circle. The turtle is based on knowledge that mosr people have: how to physically move around in the world. The main thing is the process of making the circle rather than the properties of the finished circle.

CONNECTION: What purpose do dynaturtles serve?
PAPERT: Dynaturtles are turtles whose laws of motion are like those of physics. Using these mobile turtles gives the user a chance to learn physics by actually playing with fundamental concepts. A child might use dynaturtles because he likes to create pleasing patterns on the screen. In order to create those pleasing patterns, however, he must apply the same laws that govern the paths of planets circling the sun. You might call it participatory physics.

CONNECTION: How can the animation capabilities of Atari Logo be used?
PAPERT: The animation in Atari Logo concretizes the idea of motion for learners. Motion is tradirionally one of the hardest concepts to learn in science. Paradoxically, knowledge of how to move around in the world is learned very early, but the study of motion in science is delayed until advanced levels of schooling. With Logo, very young children learn to work with concepts of motion, which places it in its rightful fundamental place in early education.

CONNECTION: What is list processing, and why does it make Logo so special?
PAPERT: In Logo, a program is a list of instructions, or a list of lists. Each instruction is a list of Logo words. If you're creating a program to draw a square, for example, you type *FORWARD 100, RIGHT 90* four times. *FORWARD 100* is a list of two words in a special order. The relationship between the program and its instructions is the same as that hetween each instruction and the words that make it up.

When "new math" came out in the 1960s, many people learned for the first time the concept of *set*. A list is like a set, with one addition—it involves process-oriented set theory. The set of objects in a list is organized in a definite way so that you come to each object in a special order.

CONNECTION: What is recursion? Why is it important to Logo?
PAPERT: Recursion is a process that refers to itself. Since it refers to itself, it can refer to itself referring to itself.

I'll make an analogy. One of the most important things about thinking is that you can think about thinking. You can even think about thinking about thinking. Similarly, Logo has programs that act on programs, and programs that act on programs that act on programs.

CONNECTION: Why is recursion powerful?
PAPERT: Because it is infinite. When rhe way a higher level acts upon a lower level is also the way a yet higher level acts on the first level, then there are no limits how far you can expand a program.

In the history of philosophy, the problem of self-reference has always come up. Consider the Greek invention of the paradox of the liar from Crete. For argument's sake, say that Cretans always lie. But if a Cretan says "I'm lying," wouldn't he be telling the truth? It's always puzzling when someone or something refers to itself.

Recursion offers a solution to the puzzle. It gives you a way of thinking about things referring to themselves.

CONNECTION: You said that computers will help bridge the gap between technical-scientific and humanistic cultures, by putting "powerful ideas in computational form, ideas as important to the poet as to the engineer." Will the bridge be built by those on the technical-scientific side?
PAPERT: Undoubtedly that is where the initiative is going to come from, but that's not where it *has* to come from. One of my most important tasks is opening doors for people other rhan those who are taking the lead now. I want to see more people get involved with computers and help shape the culture of the future. The power of computers is presently in the hands of a small minority—largely white, male, and Californian. A tiny subculture is shaping the computer culture for us all. It's scary in terms of developing social inequalities. ■

Carlos Vidal Greth is a Senior Writer in The Atari Products Company.

IN DEFENSE OF BASIC

by Carlos Vidal Greth

SEYMOUR PAPERT'S ENTHUSIASM for his language, Logo, is shared hy many programmers and educators. There is, however, some controversy about specific aspects of Logo and Papert's strongly held beliefs about BASIC. The contrasting views of Arthur Luehrmann, another major influence in educational computing, are presented below.

Luehrmann is co-founder of Bekeley-based Computer Literacy, a three-person partnership designing and writing curriculum materials for beginning computer users. Formerly a director of the Computer Education Project at the Lawrence Hall of Science in Berkeley, Luehrmann is co-author of the book, *Computer Literacy: A Hands-On Approach*.

Luehrmann shared rhe following observarions with the ATARI CONNECTION in an interview at his home in the hills overlooking Berkeley.

"Covered with Warts, But It's There"

Whether you like it or not, millions of home computers speak BASIC. Only a tiny percentage speak other languages. BASIC is covered with warts, but it's there. Current versions of BASIC on microcomputers are overly complex, tasteless, and full of inconsistencies. There's no easy way to define a procedure and use local variables and pass parameters. BASIC doesn't have tight conrrol structures; you have to huild them out of pieces. It docsn't have a loop structure or a multi-statement branch structure.

Despite the language's drawbacks, however, one can learn to read, write, and do structured and top-down programming with garden-variety BASIC.

If the educator teaches a language besides BASIC, will the kid get any practice on it outside of school? Not likely. If schools teach languages that kids only have access to in schools, kids won't learn much.

If BASIC isn't taught well in schools, the children learn street BASIC. Street BASIC is an unstructured, hacker approach where programs are built like houses of cards. You keep adding statements to a wobhly structure—a program—until it collapses into a heap.

Logo—the Pluses and Minuses

Current versions of Logo for schools are pretty good, considering that they're shoehorned into a pretty tight box [a microcomputer]. Logo's primary advantage is that it teaches how to break a problem down into manageable pieces and how to deal with rhe whole problem in an absrract way. This is called the "top-down" method in computer science. The top-down method doesn't come naturally; it needs to be taught. There's nothing magical about any language in that regard. It takes a teacher to do the job.

It's hard to teach structured programming in Logo because there's no conditional loop for repetitive functions, and branch construction is liinited. A language designer should provide a simple method of telling the compurer ro do somerhing repeatedly until some condition hecomes true. That can he done in Logo, but only by a recursive procedure. It's easier to teach beginners repetitive procedures than it is to teach them recursion.

Branches are a problem as well. The data that you type until you press RETURN is the logical line in Logo. That's constraining when the branch structure you need to do something is rather lengthy. It forces programmers ro make two choice blocks into separate procedures and give each one a name.

That decision was made in part to protecr users from errors that can propagate from multi-line statements. If you close off each line by pressing RETURN, when you do error testing you've got a known world where mistakes can be flagged. The unfortunate effect is to make lengthy if/then constructions difficult.

The Transcendent Issues

We shouldn't ger bogged down in details of languages when there are transcendent issues like learning how to solve a problem with the top-down approach and how to use control structures. These are fundamental issues, and it doesn't matter a damn what language you teach them in.

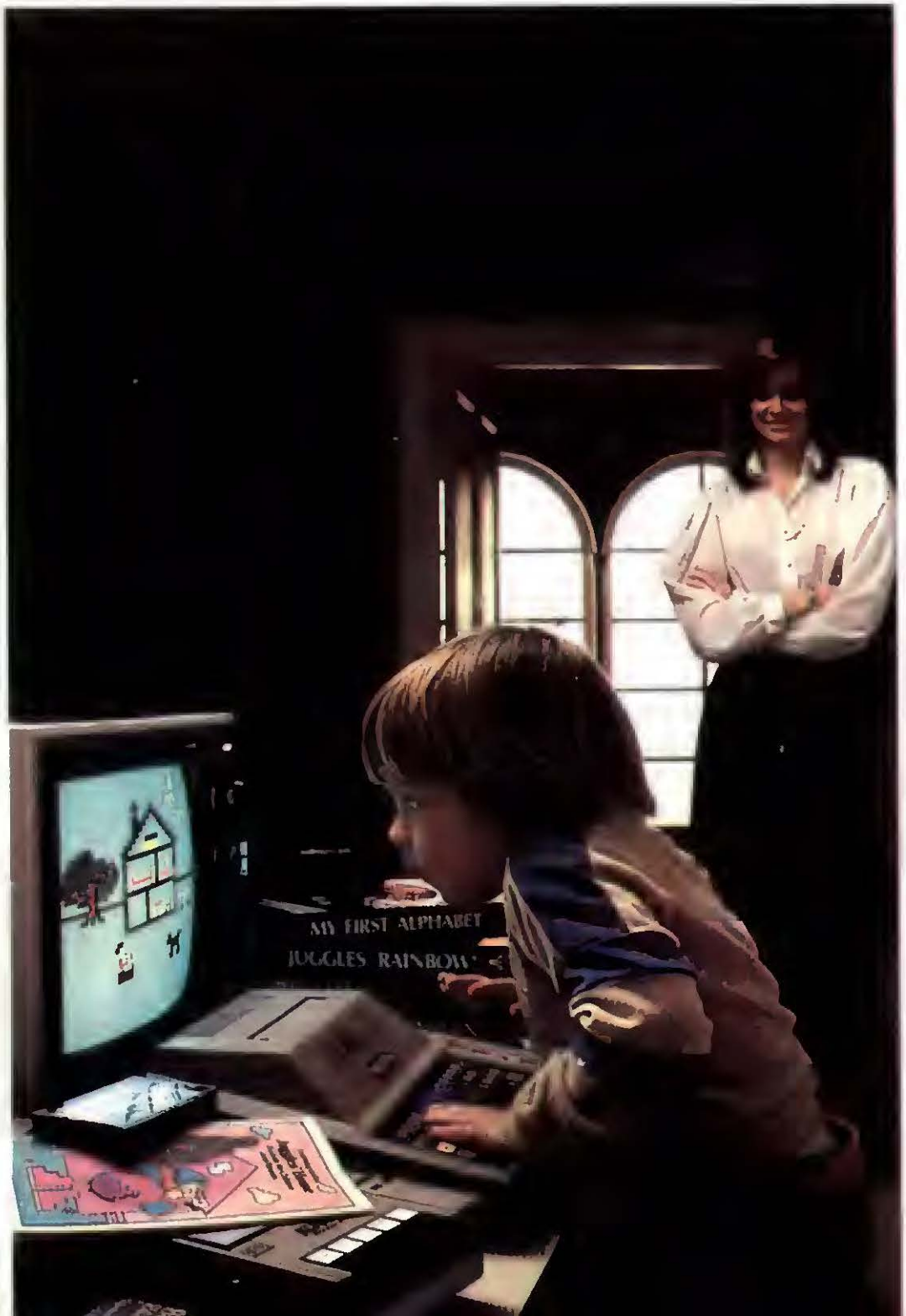
If you know any programming language, and have heen correctly taught about procedures, control structures, dara and variables, and lists, learning a new language should be a piece of cake. ■

PHOTOGRAPHED BY JEFFERY NEWBURY

ATARI CONNECTION
60 E. Plumeria Drive
P.O. Box 50047
San Jose, CA 95150

Address Correction Requested

Gavin thinks he's playing. Gavin's mother knows he's learning.



With colorful graphics, lively melodies and captivating computer animation, Juggles' House* teaches *upper* and *lower*, *inside* and *outside*. Understanding these concepts helps your child learn to recognize shapes and letters – to begin reading.

Games with windows, toy shelves and a clown's house help children understand basic writing and counting concepts. And naturally, Juggles' House comfortably establishes computer awareness.

Give your child Juggles' House for its early learning fundamentals. Or, give your child Juggles' House for the fun of it.

ATARI®

C017668-11