

In This Issue

By Seymour Papert	1
Where's the Program?.....	3
Putting Ideas into Action	4
Book Review	5
Logo Tool Box	5
Logo Users Groups	6

In this second issue of *Logo Update* Seymour Papert joins the debate about whether or not children should program. The controversy is probably familiar to you, but I think you'll find Papert's perspective different and refreshing.

And while you are thinking about the value of programming, you might want to reconsider what programming is. Eadie Adamson provides a travel guide to MicroWorlds Logo in "Where's the Program?"

In *Putting Ideas into Action* I give an overview of the venerable St. Paul Logo Project.

Carol Sperry's *Book Review* is a critique of a new collection of Logo research studies.

Articles in *Logo Update* are brief, but each one includes pointers to additional sources of information.

This issue also includes *The Logo Tool Box*, and news about Logo Users Groups, conferences, new products, and materials.

Among the new items available from the Logo Foundation is the second edition of *Mindstorms* with new forwards by Carol Sperry and by John Sculley. In his new introduction, Papert reflects on "bugs" in the 1980 classic, both his own and those of his readers.

Use the response form on page 7 to order *Mindstorms* and other books and materials, and to request the full list of Logo Foundation publications.

If you received this copy of *Logo Update* in the mail, you are on our mailing list and will continue to receive future issues. If you picked it up at a conference or got it from a

(Continued on page 2)

By

Seymour Papert

Is Programming a Good Activity for Children?

I just can't stand reading those articles in the literature for and against programming that leave you not knowing what kind of experience, if any, the author has had with this activity. Personally, I'd feel really uncomfortable about arguing that programming (or anything else) is "good for" children if I never did it myself, or hated doing it, or had any doubts about its value for me. So I thought I would write this column about the kinds of programs I have made recently.

I begin with a very simple program that came about while I was writing this column.

Most of the writing was done when I was alone in a log house deep in the woods in Maine. It had been so beautiful that I stretched out my stay longer than I ought to have and created a situation of having to leave at a fixed time. The snag was that I needed some sleep, but had no alarm clock. I know from experience that I can sleep easily through the single beep of the alarm clock feature I have on my Mac. So I had to make a better one. . . . Piece of cake! It took about half a minute to set up a Logo instruction in MicroWorlds that would keep on repeating a recorded message until I stopped it. (**Wait 4 * 36000 forever [message]**) Hardly a program, but I couldn't resist adding a frill that made it one by any criterion: At the set time it would start with a message gentle in decibel level and in content and move progressively to messages that were more insistent in both dimensions. Making it took a few minutes, plus five more to get rid of a silly bug. As I was doing this I realized that with another 10 or 20 minutes I could transform a LEGO-Logo construct I had built some time back for another purpose into a water pourer. Although this mechanical frill would absolutely ensure wake-up, I decided that sound was enough and lucked out: Here I am; awake and ready to go.

In this job I used two bits of knowledge that go beyond what a third grader with Logo experience could be expected to know. One is technical: There are 36,000 tenths of a second in an hour. The other is much more important. I fear that even in some of the very best school Logo environments the students don't actually know that what makes sense of Logo, of computers, and of programming in general is putting it to use when one really needs it in real-life situations.

Real life uses can be very varied. I have written programs as gifts for little children – sometimes to amuse, sometimes to open horizons for thinking, sometimes as a basis for relationship, most often for all these together. I have even written programs to amuse a cat. Last week I wrote one to help figure out where to plant bushes so as to make sure that a shack would be invisible at all points on a complexly shaped shore-line, and then used Logo graphics to present the underlying principles of geometric perspective without "trig" or any other "fancy" math.

These examples sound too serious. Some of the programs I do are pure play. . . . which doesn't mean they don't serve a purpose. Think of doodling. When people sit in a meeting drawing patterns on paper this doesn't mean they are not listening – doodling is more like an accompaniment to the intellectual music. I've often written little Logo programs in the same

(Continued on page 2)



Logo Foundation

250 West 57th Street
 New York, NY 10107-2603
 Telephone: 212 765-4918
 FAX: 212 765-4789
 email: michaelt@media.mit.edu

Board of Directors
 Seymour Papert, Chair
 Tessa R. Harvey
 Michael Tempel

The Logo Foundation is a nonprofit educational organization incorporated in New York State.

Logo Update is published three times yearly by the Logo Foundation.
 Subscription is free.

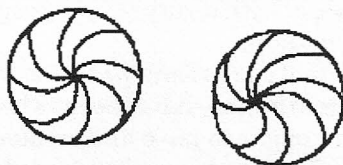
© 1993 Logo Foundation

You may copy and distribute this document for educational purposes provided that you do not charge for such copies and that this copyright notice is reproduced in full.

Is Programming a Good Activity for Children?—Cont. from pg. 1

spirit – sometimes even at a meeting, but more often, as I tried to warm up to writing this column. As a matter of fact I tell in *The Children's Machine* how the idea of the turtle came from a doodling experience.

One of the most interesting programs I wrote recently was a little job for a friend, an assistant to a prominent sculptor, who asked me if I knew any software he could use to make figures like the following with different numbers and forms of the curvy spokes and different positions in the circle for their meeting point.



Maybe there is. But it was more fun for him and for me (and probably quicker) to make some Logo tools to let him play with the figures.

The curvy spoke program touches on the reason that is most often invoked to explain why programming isn't "necessary" for nonprofessionals. They can just buy ready-made software tools. But do these always exist?

Perhaps there is a tool suitable for my friend's purpose, and of course the only reason I had to make my Logo alarm clock was that the designers of the one in my Mac weren't thinking. . . . given time the world of software makers would wake up to the likely uses of alarm clocks.

My next two examples show programming that is less likely to be made irrelevant by software tool makers. The first of the two is an exceedingly simple case where I used the computer as a medium in the way I use pencil drawings when I see a new wildflower.

Last winter I saw my first total eclipse of the moon and was lucky enough to be able to observe it through the window next to the computer I most often use at home. My first thought was to take advantage of the computer to keep notes on the progression of the eclipse, which, as I learned for the first time from

watching it, takes more than an hour. So I fired up MicroSoft Word. I soon realized I'd like to have graphics as well so I switched to Logo. As the eclipse progressed I began to ask myself questions. The form of the eclipse suggests a circle – the full Moon – moving through a second circle – the shadow of the Earth. What is the ratio of their diameters? I was able to do some quick experiments which enabled me to find the answer and then to predict pretty accurately when the eclipse would end. None of this required any formal mathematics beyond junior high school level and in fact led me to some new general insights about geometry as well as to specific ones about the moon, the sun, and the earth.

The most "technical" programming I did this summer was the result of conversations at NECC with Michael Tempel and Brian Silverman about a kind of project in exploratory math that they have developed for use in workshops with teachers. Consider the following rule for transforming numbers:

If the number is even divide it by 2. If the number is odd multiply it by three and add 1

Look what happens if you apply this rule to a number and recursively to the result. Choose 7 as the starting number:

7 → 22 → 11 → 34 → 17 → 52
 → 26 → 13 → 40 → 20 → 10
 → 5 → 16 → 8 → 4 → 2 → 1

Choose another number and I bet that it will also end in 1. I say "I bet" because very determined attempts to find one that does not have all failed, but as far as I know there is no proof that there isn't some enormous number that will prove to be the first exception.

What Michael and Brian have discovered is that Logo is a wonderful tool for exploring a problem such as this. Many teachers who would find the problem too tedious to explore empirically without good computational tools and too technical to explore analytically, find it

In This Issue—Cont. from pg. 1

friend, you may use the response form on page 7 to request a subscription. Do you know someone who received the first issue of Logo Update last spring and is wondering where the second issue is? That person probably didn't ask for it.

There is no charge for Logo Update, but you must request a subscription by using the response form, or simply by calling, writing, or sending an electronic mail message with your name and address.

I also encourage you to make copies of Logo Update to distribute in courses, workshops, and to give to colleagues. (Please comply with the conditions of the copyright notice in the box above.)

Finally, thanks to all of you who made donations to the Logo Foundation. An annual contribution of \$25 covers the cost of your copies of Logo Update and helps to support other free services provided by the Logo Foundation.

I hope you find this issue of Logo Update to be informative and thought provoking.

Michael Tempel

tantalizing when presented in a good Logo environment. For example, think about how you might pursue this problem in the range of numbers in the millions or billions without a computer!

But the antiprogrammers will say: Exactly, you need computer tools but not programs. A few incidents from my own and other people's explorations of this problem would be enough to remove any tendency I ever had to agree with this.

First of all, some of the greatest excitement came from setting up the computer to do a long and tedious exploration overnight. There is something very special about coming in in the morning to see how it cooked overnight.* In my playing with the $3n + 1$ problem, one time in four or five the tense expectation was rewarded with an interesting mathematical nugget – a little discovery, something to think about.

Making the computer do the work is going beyond tools – it involves programming in an essential way. But even if one could imagine a “tool” to do that without “programming,” another incident rejoins the alarm clock and the moon situation in reminding us that the makers of software just can't anticipate all needs.

Exploring large numbers took me into higher and higher ranges. . . . millions, trillions . . . and eventually beyond the size of number that could be handled by whatever tools I might be using – as it happens, Logo. At this point I'd have been flummoxed had I been relying on a non-programmable tool. As it happened, I could simply write a Logo program that gave me the greater precision of arithmetic I wanted! ▲

* Mitch Resnick gives some excellent examples of how this kind of excitement can be used in school settings. See *Logo Overnight*, 1993, Logo Foundation.

A companion piece to this article, *Is Programming a Good Activity for Children? — Code and Comments*, is currently being prepared by the author. Check the appropriate box on the Response Form on page 7 if you want to receive a copy.

Where's the PROGRAM?

by Eadie Adamson

All versions of Logo provide ways for you to look at programs. If you need to know how something works, want to debug it, or simply want to look at programming style, you can **print out procedures (pops)**, or **poall** to see variables and procedures. In some Logos **edit all** puts all procedures and variables in the editor for you to examine and modify. With LogoWriter you “flip” the page. Someone knowledgeable can look at the code and deduce what a program does.

With MicroWorlds Logo these strategies will severely limit investigating or debugging a program. What we used to think of as “the program” now doesn't tell the whole story. There's a lot more going on than what may be revealed by a peek at the Procedures Page. The Procedures Page may even be blank! The program may be attached to turtles, buttons, and other objects.

Where is “the program”? Do we need a new definition of the word program? Here are some places to look as you find your way around a MicroWorlds project:

Procedures Page

You write procedures here as you would in a Logo editor or on the Flip Side of the LogoWriter page.

Turtles

Instructions may be attached directly to turtles. Then clicking on a turtle with the mouse runs the instruction. These instructions may be Logo primitives and/or procedures you write on the Procedures Page. In the project below, a turtle named “bird” carries an instruction to alternate between shape 11, a wings-up bird and shape 12, a wings-down bird, while moving forward a bit.

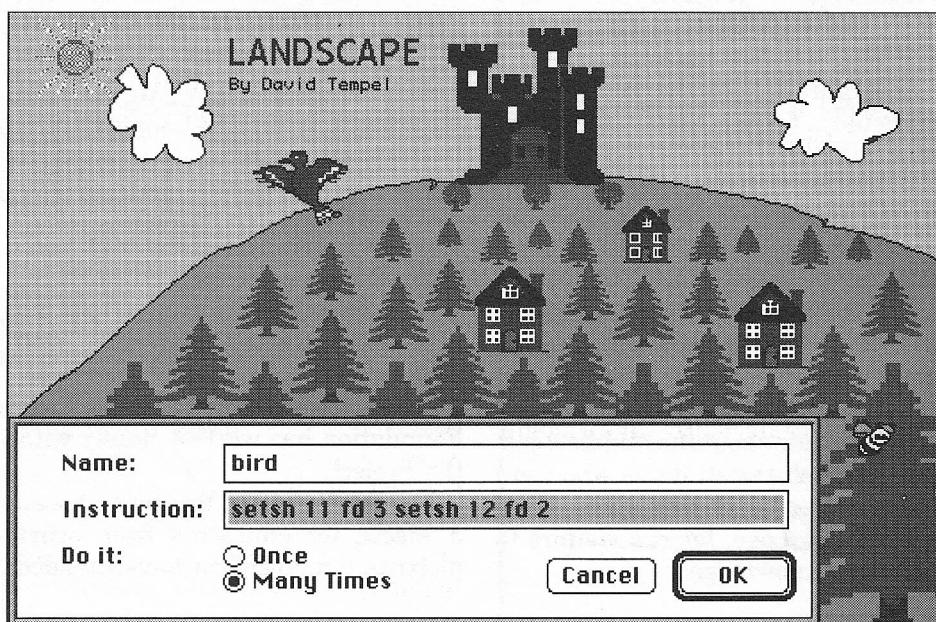
Colors

Colors may be programmed so that when a turtle touches an area of a certain color an instruction list is run only “once” or “each time” the turtle goes over that color. The choice is yours. You may also program a color so that an instruction list is run when you click the mouse in an area of that color.

Text Boxes and Sliders

Text boxes and sliders contain information that may be used by a program. The name of a text box reports the text that's in it. The name of a slider reports its value. In MicroWorlds Logo you may find yourself using text boxes and sliders where you used to use variables.

Remember, with MicroWorlds, there's more there than you think! Where's the program now? ▲



Putting Ideas into Action



The St. Paul Logo Project

by Michael Tempel

The St. Paul Logo Project is different. For the past 12 years, Geraldine Kozberg, founder of the project, has held a series of administrative positions in program and staff development with the St. Paul Public Schools. None of these positions has been "Technology Coordinator." In St. Paul, Logo is not only something to do with a computer — it is one of a number of programs and strategies aimed at school change and improvement.

Logo in the Classroom

St. Paul has avoided School's tendency, described by Seymour Papert in *The Children's Machine*, to confine computers to a special place with a specialist teaching "computer literacy." In St. Paul, Logo is part of the life of classrooms, taught by regular classroom teachers. Even at the secondary level, Logo is used by mathematics, science, English, social studies, and art teachers.

First grade teacher Helen Kraft at Jackson School developed a science project on animals. It was not a "Logo project," but Logo played a part. Along with many other activities, children drew animals and wrote about them using Logo. In the process they also learned some geometry. Through Logo, many children began writing before they learned to read.

At Mann School the school-wide curriculum for the 1989-1990 school year was the Middle Ages. The culminating activity was an "Expo" during which the gym and corridors were filled with exhibits of student work for parents and other visitors to see. Judy Ronnei's fifth-graders built a LEGO castle, complete with computer-controlled drawbridge and jousting knights. A computer was set up with a quiz about the Middle Ages. But it wasn't the students who were being subjected to this quiz. They wrote it, in Logo, for the visitors to test their knowledge.

St. Paul students created a LEGO Logo exhibit at the Minnesota Museum of Science. It included a "house of the future" in which furniture automatically moved out of the way when a vacuum cleaner approached. An "alarm bed" tilted up, sliding its occupant out a window and down onto the street below, ready for the day ahead.

Assessment

When it came to assessing the early results of working with Logo (1983 and 1984), University of Minnesota sociologist Pete Fire Dog found that about half the students using Logo showed academic improvement, according to their teachers. Improvement was seen among students of varying backgrounds. "Logo effects seem to be both wide ranging and substantial, and appear to be available to students from almost any type of learning, social, or motivational background."*

Collaboration

From its beginning in 1982, the St. Paul Logo Project has been a collaborative effort with corporate, university, and foundation support playing a major role. The St. Paul Companies provided financial aid. Engineers from the 3M Company tutored in classrooms. University partners included University of Minnesota, Macalester College, and Hamline University.

Outside consultants have played an important role. Seymour Papert and other members of the MIT Logo Group have taught and lectured in St. Paul. Logo Computer Systems provided software, materials, and professional development services. Since its founding in 1991, the Logo Foundation has worked closely with the Project.

The St. Paul Logo Project has been a Mecca for educators from other districts, many of whom have attended

the annual Logo Summer Institutes that have been the foundation of Logo teacher education.

Over the past 12 years approximately 350 teachers from half of St. Paul's 66 schools have been involved in the project. Currently there are 20 active Logo schools.

The project has changed with the times, adopting new software and hardware. What has not changed is the commitment to Logo's constructionist philosophy, and to educational equity and school improvement. ▲

* Fire Dog, Pete "Exciting Effects of Logo in an Urban Public School System" *Educational Leadership*, September 1985

For more information . . .

The St. Paul Logo Project has published collections of research papers, essays, classroom reports, and lesson plans:

Logo Learning in a Computer Culture (1985)

Logo Learning in a Computer Culture—Book 2 (1988)

The Way It Ought to Be; LEGO Logo Lesson Plans (1988)

Logo Animals
by Helen Kraft (1991)

These documents may be obtained without charge by contacting

Geraldine Kozberg
St. Paul Logo Project
360 Colborne Street
St. Paul, MN 55102
612 228-3631

Book Review

by Carol Sperry



New Paradigms in Classroom Research on Logo Learning

Edited by Daniel Lynn Watt and Molly Lynn Watt, ISTE, 1993 \$19.95

In 1991, a group of teacher-researchers came together for a one-day miniconference on Logo classroom research at the National Educational Computing Conference in Arizona (NECC). Nine of the papers presented at that conference are included in this monograph, the third in a series published by the International Society for Technology in Education.

A section called "Cognitive Outcomes of Logo Learning" comprises five of the nine papers. The remaining four are categorized as "Logo Learning in a Social Context." The work represents a healthy attempt to pair teachers and researchers in collaborative ventures and to try finally to place Logo research in the context of the classroom or learning culture.

I found the lead-off article, "Making a Case for the Learning Culture as the Focus of Classroom Research," by Jim Dunne, a valuable overview and good critique of past research, most of which tries to treat Logo as a stand-alone treatment for what ails the schools, or, as Sylvia Weir states, "... as some patent medicine, good for everything regardless." As Dunne points out, "Without the associated learning environment, Logo is just technical knowledge."

Dunne cuts to the heart of the matter by stating: "Many of the underlying issues, themes, and approaches associated with Logo are closely linked to old struggles on the nature of learning, the purpose of schooling, and the role of teaching. The history of education and the literature of educational change indicate that it is unlikely that an innovation or reform such as Logo will succeed when it conflicts with the prevailing beliefs and organizational structure of schools." We've all seen instances of this – creativity and excitement in the computer lab or around the computer – then back to a dry, traditional classroom atmosphere. But Dunne goes on to give a quite comprehensive and helpful list of what constitutes a good Logo learning culture. Finally, he notes that the current restructuring movement could provide the most fertile ground for Logo-like learning environments and for the creation of implementation models that "can help guarantee the fruitful survival of Logo."

The remaining eight articles are all based in classroom (or in one case, a museum) situations. Each offers insight into issues that surround learning and teaching, the life of teachers, cultures of learning, and the school system. In most cases, both teacher-researchers and university researchers are quite candid in their reporting of the ups and downs of the process. Consequently, researchers can learn a lot about their own processes and possibly avoid some pitfalls. It is refreshing to see work done by teachers, themselves, to answer their own real questions, rather than reports abstracted from a situation by researchers alone. Of course, even teachers can fall into the trap of trying to squeeze Logo learning research into a quantitative straight jacket. I think it's fair to say that shadows of an anachronistic educational system do fall here and there on these works: Computer time seems restricted in many of these school situations and there is certainly balkanization of subjects, but these can serve as reminders that the system has a long way to go. ▲

New Paradigms in Classroom Research on Logo Learning is available from International Society for Technology in Education (ISTE)

1787 Agate Street

Eugene, OR 97403-1923

Telephone: 800 336-5191 or 503 346-4414

FAX: 503 346-5890

email: ISTE@Oregon.uoregon.edu

Logo Tool Box

Have you ever written an interactive program that starts out something like this?

```
to talk
ct
print [What's your name?]
name readlist "response
...etc.
end
```

If this is in LogoWriter, you've set yourself up for a possible disaster. If the flip side of the page happens to be showing when you type **talk**, the text that is cleared by the first line of the program (**ct** stands for **clear text**) is the program itself! The **undo** command will restore text after **ct**, but only before another text-related command is given. In the procedure above, once the line

```
print [What's your name?]
```

is run, it's too late to use **undo**. Here's a one-line fix you can include at the beginning of any program that clears text:

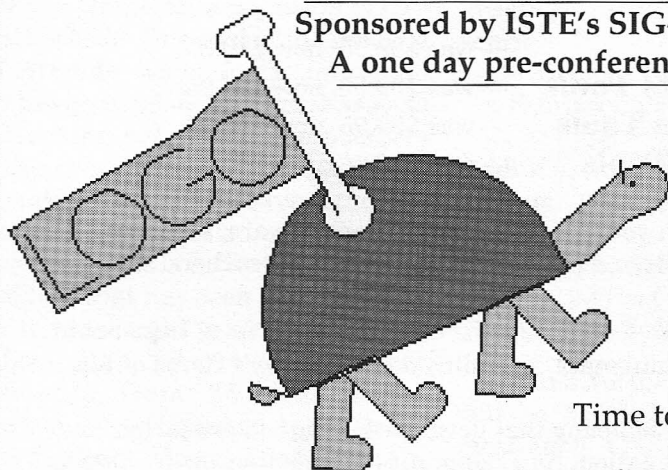
```
to talk
if not front? [flip]
ct
print [What's your name?]
name readlist "response
...etc.
end
```

Tom Trocco, who teaches at St. Hilda's and St. Hugh's School in New York City, advises his students never to use **ct**. He gives them an **erase front** procedure to use instead.

```
to erf
if not front? [flip]
ct
end
```

Save Sunday, June 12, 1994 for the Logosium (Logo Symposium)

Sponsored by ISTE's SIG-Logo and the Logo Foundation
A one day pre-conference at NECC '94, Boston, Massachusetts



Mini hands-on sessions

Jitterbug Logo Style — it's procedural!

Students, student projects, and video tapes

Time to schmooz with Logo users from around the world

Interactive, informal group discussions on Logo topics – assessment, NCTM standards, action research, new environments, beginning and advanced topics

Look for more information in the next issue of *Logo Update*.

Logo Foundation

250 West 57th Street • New York, NY 10107-2603

Nonprofit Org.
U.S. Postage
PAID
New York, NY
Permit No. 6378

