

# Logo Update

The Logo Foundation Newsletter ▲ Volume 3, Number 1 – Fall 1994

## In This Issue

A New Approach.....	1
by Michael Tempel	
Logo Tool Box.....	3
New Ways of Creating Variables .....	6
by Sharnee Chait	
Turtle Math in the Classroom .....	10
by Douglas Clements and Julie Meredith	
Book Review.....	14

There is nothing new in this *Logo Update*. Wait! Don't throw it away just yet. What I mean is that the articles continue themes that began in the last issue. "A New Approach" and the "Logo Tool Box" are about workshops that emerged from discussions that Seymour Papert described in his column, "Advancing Logo." In "New Ways of Creating Variables with MicroWorlds" Sharnee Chait responds to Kathryn Verzoni's description of her students' MicroWorlds projects. Doug Clements and Julie Meredith offer more information about Turtle Math.

In keeping with our theme, Carol Sperry's Book Review is about an old book, but a great one, *Computer Environments for Children*, by Cynthia Soloman.

And it's not too early to start thinking about next summer. Logosium, the one day Logo conference held in Boston last June, will be repeated at NECC '95 in Baltimore. The St. Paul Logo Summer Institutes will be held again for the 14th year. Look at pages 14 and 16 for more information.

Are you on our mailing list? If you received this issue of Logo Update in the mail you *are* on our mailing list and will continue to receive future issues. If you got it from a friend or colleague, or at a conference, you can request a subscription by calling the Logo Foundation or by returning the Logo Foundation Response Form on page 15.

*Michael Tempel*

## A New Approach

by Michael Tempel

It was near the end of a five-day Logo workshop, time for sharing projects on the big screen. There were car races, boat races, and fish races; a rabbit searching for carrots and a turtle trying to find its way home; a maze, a pacman, and a pong game.

I was facilitating this workshop for 25 "Tutors", the people who provide training and support for teachers in 160 Costa Rican elementary schools in the Programa Informática Educativa (PIE), a joint project of the Fundación Omar Dengo (FOD) and the Ministerio de Educación Pública (MEP). During the past year I have spent a good part of my time in Costa Rica as one of a group of international consultants working at the FOD under a grant from the Banco Interamericano Desarrollo\* (BID). Our charge has been to develop Logo curriculum materials, and to define what that means.

The workshop was inspired by discussions Seymour Papert referred to in "Advancing Logo," his column in the spring, 1994 issue of *Logo Update*. The "narrative" style of Logo program that he describes – using Logo to present a report or tell a story – has been developed to a remarkable degree in the Costa Rican schools over the past six years. I attended a Children's Logo Conference last October where students presented a series of such projects to schoolmates, parents, and guests. One major work was a study of the presidents of Costa Rica. This 20 minute extravaganza included biographical and historical information, along with a full color portrait, drawn using the turtle, for each of Costa Rica's 39 presidents. Remarkable projects like this have been presented at national and regional Children's Logo Conferences over the years.

One is so overwhelmed by the beauty and quality of these projects that it is difficult to adopt a critical stance and look for areas of potential growth and improvement. Yet that is precisely what is happening within the PIE. In part there is a wish to encourage children to use a more procedural style of Logo programming. But also, with the PIE now six years old, there is a growing feeling of wanting to broaden and diversify the kinds of activities that students engage in – to develop a "new approach" to Logo programming and projects.

In April we began designing workshops for the Tutors and teachers to be followed by testing of materials and teaching strategies in the schools. The school year in Costa Rica goes from March through December with a two-week break in July. We planned to use this July break for workshops, and then pilot our new approach in selected schools during the second half of the school year.

As part of the workshop planning I wrote a game program and saved versions of it at various stages of development. Along with the game I wrote a collection of Logo tool procedures that could be used as if they were primitives. As I went along, I modified the Logo tool procedures and added new ones. This process helped me to become familiar with possibilities and problems and in the end, I had a collection of tools and sample programs for use in the July workshops. (See the "Logo Tool Box" on page 3.)

We began to use the term "event programming" to refer to a range of Logo projects that include interactive video games, animations, and simulations. An important element in this type of program is the uncertainty about its course and the unpredictability of its outcome. The flow of action is affected by pressing keys and there may be elements of randomness, as well.

I don't mean to suggest that with our "new approach" we were journeying into uncharted territory. Interactive programs and games have long been a

\* Interamerican Development Bank

(continues on the next page)



(continued from page 1)

part of the Logo culture, familiar in Costa Rica as well as in other centers of Logo activity. In fact, some of the Tutors used the workshop time to modify and improve projects they had begun previously. But in Costa Rica, and elsewhere, the narrative style of Logo project has tended to be dominant, at least among LogoWriter users. In our approach to event programming, we drew upon old ideas but added some new dimensions, especially in our way of using Logo tool procedures to support the style of programming we sought to encourage.

During several months in which I was involved in planning and teaching these workshops we had frequent discussions about the many educational issues involved in this new approach. How do we present the tools? Do we introduce them to learners in an orderly sequence, or on an "as needed" basis as projects are developed? What relevance do games have to school subjects and curriculum, and to important domains of human knowledge?

Papert shed light on this last issue in a video tape he prepared for use in the workshop. He showed several examples of Logo programs. In one, a bee flies randomly around the screen. When it encounters a red flower, it stops, but it ignores flowers of other colors. This beginning could be elaborated into a simulation of insect behavior and plant propagation. Later in the video, we see a person walking across the screen and attempting to jump over an obstacle. There are many ways to program the jump. While working on this problem the learner will be engaged in questions about the laws of motion.

Papert was involved in this workshop not only through his video presence, but much more directly, as well. He participated in the planning process during the months leading up to the workshop. During the workshop itself we had numerous email exchanges and a few live telephone calls using a speaker phone so that the entire group could participate in the conversations.

During the second week of July there was another workshop for the

teachers who had been selected to pilot the new ideas and materials with their students. This time two of the Tutors, Efraim Lopez and Julia Rivera, joined me as instructors. As soon as the first workshop ended, the three of us looked closely at people's projects, sorted out what had happened, and developed plans for the following week. Based on our experience we simplified the tools and samples. During the second workshop the projects that people created were similar to those we saw during the first week, but they were more elaborate. This was partly because there were fewer problems with the tools, allowing people to move ahead more easily with their projects. And, since it was the final week of the World Cup competition, the projects included a soccer game.

The months of planning and the two weeks of workshops were an intense experience for me, and a very rewarding one. It has also been a time of exciting change and growth for the Logo project in Costa Rica. In May, a newly elected government took office. President José Figueres and Education Minister Eduardo Doryan

are committed to increasing resources for education. This includes expansion of computers in education projects. The PIE, which now serves 30% of Costa Rica's elementary school children, will grow to reach 50% within the next four years and a new project is being initiated in the secondary schools.

But the FOD has other changes to adjust to. Clotilde Fonseca, Executive Director of the FOD, and Eleanora Badilla, Director of the PIE, have both taken positions in the new government. The expansion of the PIE is proceeding with the guidance of its new director, Andrea Anfossi. The BID project has come to an end and its director, Claudio Gutierrez, has returned to his position as Professor of Computer Science at the University of Delaware.

After I left Costa Rica in July there was yet another round of workshops for more teachers and for ten new Tutors who had just been hired to provide the additional support needed for the growing project. This story is far from over. To be continued . . . ▲

### For More Information

The following publications are available from the Logo Foundation. Use the response form on page 15 to order them.

Fonseca, Clotilde, *Computadoras en la Escuela Pública Costarricense*, Fundación Omar Dengo, San José, Costa Rica, 1993. A comprehensive study, in Spanish, of the Programa Informática Educativa written by the Executive Director of the Omar Dengo Foundation.

Fonseca, Clotilde, "A National Transformation" in *Logo Update*, Vol. 1 No. 1, Spring 1993. A brief description of the project.

Papert, Seymour "Advancing Logo" in *Logo Update*, Vol. 2 No. 3, Spring 1994. Inspiration for the workshops described in this article.

*A Revolution of Opportunities*, Fundación Omar Dengo, San José, Costa Rica, 1993. A two-part video that provides a brief overview of the Programa Informática Educativa and of the Children's Logo Conferences.

Tempel, Michael, *Event Programming*, Logo Foundation, 1994, A collection of Logo tools and programs that were developed in conjunction with workshops offered at the Omar Dengo Foundation in July 1994. The programs are in both Spanish and English. The commentary is in English. An accompanying disk contains the Logo programs and tools for use with either the English or Spanish versions of LogoWriter for MSDOS computers.

# Logo Tool Box

by Michael Tempel

Here are a few examples of the kinds of game programs and tools we used in the Event Programming workshop described on the front page of this issue of Logo Update. These procedures were originally written in the Spanish version of MSDOS LogoWriter. I've translated them into English for this column.

In the first game the goal is to get the turtle to reach a square blue target by pressing the keys **f**, **b**, **r**, or **l**. The **setup** procedure creates a target and then places the turtle in the center of the screen. **Play** causes **go** to run repeatedly. In **go**, **getkey** remembers the last key that has been pressed. **Ifkey** causes something to happen if a certain key is pressed. For example, if the last key **getkey** got was an "f", the turtle moves **forward 10**. **Ifcolor** takes action if the turtle is on the specified color, in this case, color number 2. If the turtle is on the target "I win" and the program stops.

```

to game
  setup
  play
end

to setup
  rg ct
  target
  place.turtle
end

to target
  setsh 11
  setc 2
  pu setpos [120 80]
  pd stamp pu
end

to place.turtle
  setsh 0
  setc 1
  pu home
end

to play
  forever [go]
end

to go
  getkey
  ifkey "f [fd 10]
  ifkey "b [bk 10]
  ifkey "r [rt 90]
  ifkey "l [lt 90]
  ifcolor 2 [Print [I win!!]
              stopall]
end

```

The procedures **forever**, **getkey**, **ifkey**, and **ifcolor** are not primitives even though we are using them as if they were. They are tools written in Logo.

```

to forever :action
  run :action
  forever :action
end

to getkey
  ifelse key?
    [name readchar "current.key]
    [make "current.key "nothing]
end

to ifkey :key :action
  if :key = :current.key
    [run :action
     make "current.key "nothing]
end

to ifcolor :color :action
  if colorunder = :color
    [run :action]
end

```

## Logo Foundation

250 West 57th Street  
New York, NY 10107-2228  
Telephone: 212 765-4918  
FAX: 212 765-4789  
email: michaelt@media.mit.edu

*Board of Directors*  
Seymour Papert, Chair  
Clotilde Fonseca  
Tessa R. Harvey  
Geraldine Kozberg  
Michael Tempel  
Takayuki Tsuru

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.

© 1994 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.

(continues on the next page)



(continued from page 3)

The program could have been written without the tools. Instead of using **ifcolor** we could have written

```
if colorunder = 2 [Print [I win!!]
                  stopall]
```

Without **getkey** and **ifkey** our program might look something like this:

```
to go
if key? [name readchar "current.key]
if :current.key = "f [fd 10]
if :current.key = "b [bk 10]
etc.
```

Our tools reduce the use (and misuse) of Logo punctuation. They generally follow forms that are familiar to beginning Logo users. For example, **ifcolor** takes two inputs, a number and a list of instructions, just as **repeat** does.

An element of chance may be introduced by having the target placed at a random spot on the the screen. In target we can use **setrandompos** instead of **setpos [120 80]**. The target will appear at a different place on the screen each time the game is played. **Setrandompos** looks like this:

```
to setrandompos
seth 90
fd 10 * random 100
seth 0
fd 10 * random 100
end
```

What if we want the turtle to start with a certain amount of energy? Her energy runs down as she makes her way to the target, so she must get there before it runs out. We change **setup** to be

```
to setup
rg ct
target
place.turtle
setenergy 10
end
```

and **go** to be

```
to go
getkey
ifkey "f [fd 10 setenergy energy - 1]
ifkey "b [bk 10 setenergy energy - 1]
ifkey "r [rt 90]
ifkey "l [lt 90]
if energy = 0 [Print [I lose!]
              stopall]
ifcolor 2 [Print [I win!!]
          stopall]
end
```

The additional tools we need are

```
to setenergy :howmuch
make "energy :howmuch
end

to energy
output :energy
end
```

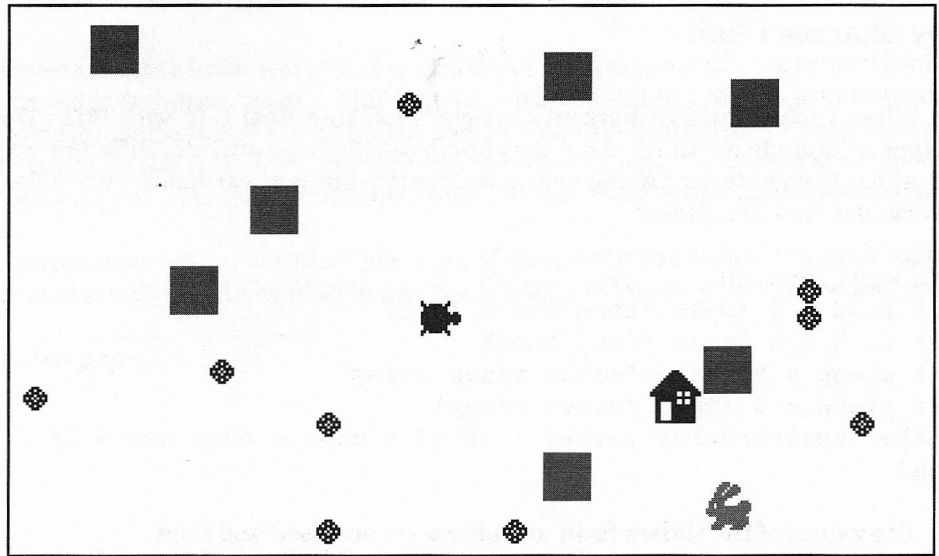


**Energy** and **setenergy** are like the primitives **heading** and **setheading**, **color** and **setcolor**.

I prepared ten games for the workshop, each one with a new twist. I added flowers, randomly planted around the screen. The turtle's energy was replenished when she ate one. I added obstacles. In the final version the turtle had to avoid a dangerous rabbit which unpredictably changed position on the screen and caused a major loss of energy if touched.

During a week of heavy workshop use these tools went through major modifications. I was reminded once again that one can never fully debug one's own program. With 25 expert Logo programmers using the tools for their own purposes, problems and bugs emerged which I never anticipated. By the second week the tool collection had been stabilized.

There isn't enough room in this column to describe the debugging process we went through, or to list all the tools and samples we used. See "For More Information," on page 2 to find out how to obtain the full story of Event Programming.▲



Having gained energy from eating flowers the Turtle heads home. She has had to navigate around square obstacles and stay away from the rabbit.

## Open the Power of Logo with WinLogo™

A New Logo Language for Advanced Learning

*WinLogo for Windows*



### Features:

- 12 turtles, color graphics, import of PCX, BMP, EPS, PS, TIFF, TGA files, etc.
- 300 built-in primitives, math capability, list processing, debugging tool, extension using C or assembly language
- font and printer support, icons, on-line help

	<u>DOS</u>	<u>Windows</u>
Student Edition*	\$ 49	\$ 59
Single user	\$ 99	\$149
5 User Pak	\$249	\$349
10 User Pak	\$299	\$449
20 User Pak	\$499	\$649

(\* no documentation)

Also available: WinLogo/LASY Robotics System

**SoftEast Corp.**

PO Box 764  
Concord, MA 01742  
Call: (617) 893-4858 or Fax: (617) 893-4948

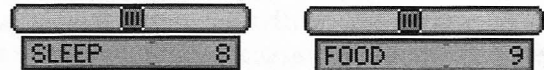
# New Ways of Creating Variables With MicroWorlds

by Sharnee Chait

When I read Kathryn Verzoni's article "Modeling Real Life with MicroWorlds Project Builder" in the last *Logo Update*, it made me think about how MicroWorlds Logo provides different ways for creating variables. The program that Kathryn's student wrote demonstrates two kinds of variables – variables created with the **make** command and variables that are sliders.

```
to calc.health
if food > 5 [make "food minus food]
if food < 6 [make "food food]
if sleep > 8 [make "sleep minus sleep]
if sleep < 9 [make "sleep sleep]
make "healthrating :sleep + :food + mood - nuisance + 38
end
```

The values of the sliders **food** and **sleep** are accessed and then stored or changed with the **make** command. The values of the sliders **mood** and **nuisance** are used to calculate **healthrating**.



## Dynamically Changing Values of Sliders

A little known feature in MicroWorlds is how to change a slider's value under program control. For example, **sleep** is the name of a slider. Say you wanted to change the value of **sleep** if it's less than 9 to one more than it already is. This instruction would look like:

```
if sleep < 9 [setsleep sleep + 1]
```

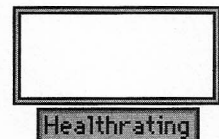
The command **setsleep** is automatically created when you create the slider **sleep**. **Setsleep** talks directly to the **sleep** slider. **Sleep** reports its current value. If **sleep** is 5 then **setsleep sleep + 1** sets the slider to 6.

## Text Boxes Can Be Variables Too

This concept of using the value of a slider as a variable can be extended to text boxes. For example, Kathryn's project calculated a health rating. Here is an excerpt of the instructions:

```
talkto "text1 ct
print [under these conditions the cat's health rating is] insert
:healthrating
```

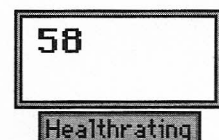
You could rename the text box called **Text1** to call it **Healthrating**. A text box has its own dialog box in which you can change its name and make the name show as a label. The text box would then look like this:



Instead of writing another procedure to show the value of **Healthrating**, you can directly set it when you're doing the calculation.

```
sethealthrating :sleep + :food + mood - nuisance + 38
```

This number would be printed in the **Healthrating** box. When you want to set a new value, just run the instruction again.



## Page Names

There is one minor aspect of Kathryn's code that is a hold-over from LogoWriter. Changing pages in the project could be simplified. In MicroWorlds, you don't need to **getpage** "page1. This instruction is redundant. Any page name such as **page1** is a command. So instead of typing **getpage** "page1, simply type **page1**. You would still need to use **getpage** if the page name were a variable. This might be the case if you were getting a series of pages to make a slide-show.

This idea of a name working like a command is also true of melodies and sounds. Melody and sound names are commands that cause them to play.

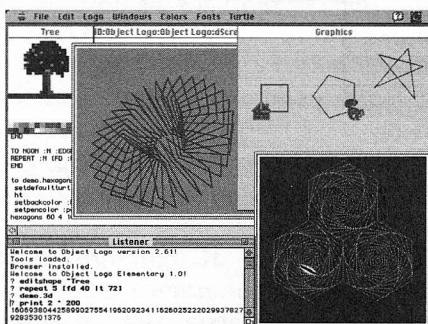
In the MicroWorlds programming environment, there are multiple ways of doing any one thing. It's exciting to discover new ways of doing things, but also comforting to be able to use the familiar old ways when you want to.▲

*Sharnee Chait is Director of Product Development at LCS*

**Object Logo™ version 2.7** which is compatible with 32 bit addressing and virtual memory, is available as a free upgrade to users of Object Logo 2.6 or later, and for \$35 for users of earlier versions. Contact Paradigm Software with your license number to request upgrades.

See the advertisement below for telephone and fax numbers.

## ObjectLogo™ for the Macintosh® - The Logo Language for All Ages!



Regarded by educators as the most powerful Logo on the market, *Object Logo* is now also one of the easiest Logo languages to learn and use. Thanks to the 180-page highly acclaimed tutorial, *Logo for the Macintosh*, by Harold & Amanda Abelson. Whether your interest is for home or school, give *Object Logo* a try. The Student Edition (includes the tutorial) is well-suited for the beginner. The Full version (includes the

tutorial and the Reference Manual) is for a more serious exploration of programming on the Macintosh. Lab Packs include both the tutorial and Reference Manual.

### Features Include:

- Support for 32-bit Addressing
- Extensive Math Features
- Advanced List Processing
- Full Macintosh color
- Complete Object System
- Fast Incremental Compiler
- Supports Multiple Windows
- LEGO® Logo Support
- Comprehensive Tutorial

**Student Edition**

**\$75.00**

**Full Version**

**\$195.00**

**Five Lab Pack**

**\$341.00**

**Ten Lab Pack**

**\$487.00**

**Twenty Lab Pack**

**\$585.00**



**Paradigm  
SOFTWARE**

P.O. Box 2995 • Cambridge, MA 02238 • tel: (617)576 7675 • fax: (617) 576 7680



# Turtle Math in the Classroom

by Douglas H. Clements and Julie Sarama Meredith

In our previous article, we described principles on which we based the design of *Turtle Math*. Here we describe what happened when we took *Turtle Math* into classrooms. *Turtle Math* is more than a revised Logo environment. It is also a set of activities in seven topical areas of mathematics: Paths and Measurement, Shapes, Scaling, Coordinates, Motions and Symmetry, Number and Computation, and Counting and Chance. There are several activities in each area. We can only describe a few in this article.

Twenty-four teachers helped us by testing early versions of *Turtle Math* in their classrooms. They taught grades 3 to 6 in public schools in New York, Massachusetts, and Florida. About half used computer laboratories; the rest had from 1 to 6 computers in their classrooms. Some used *Turtle Math* almost every day, cycling their kids through about 1 activity every week. Others did *Turtle Math* for 1-2 days intensely several times a month. We closely observed 6 classrooms. The teachers did all the teaching, we observed students as they worked on the activities and conferred with the teachers at the end of the day. We will organize our classroom findings by the design principles we used to create *Turtle Math*.

## 1. Encourage construction of the abstract from the visual and intuitive

Children have difficulty with measurement of distances and turns, even sometimes confusing side length with turns. *Turtle Math* provides two commands, **rtf** and **lrf**, to help beginning students build on previous experience ("**rtf**" stands for "right face") and appreciate the idea of turns before meeting measurement in degrees. It also provides measurement tools. Finally, the activities ease students into measurement. For example, in the first activity, "Scavenger Hunt" (Fig. 1), students have a list of objects to "find" with the turtle. The grid in units of 10 helps them get a feel for the size of turtle steps. It also re-

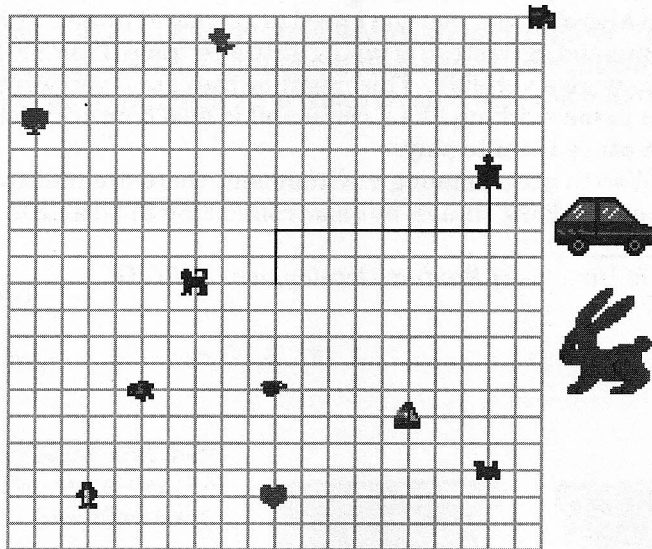


Fig. 1. Scavenger Hunt

quires only **rtf** ("right face") and **lrf** commands. For all students we observed, turning "right face" was meaningful and easy. There was no confusion between these turn commands and the movement commands (**fd** and **bk**), as research frequently reports for beginners using **right** and **left**.

Research also indicates that children typically use idiosyncratic number inputs in their initial exploration of Logo (Weir, 1987), and use successive approximation in getting the turtle to a certain goal (Noss & Hoyles, 1992). Children using *Turtle Math* did not use arbitrary inputs. They also combined commands rather than use successive approximation. For example, one girl said to her partner, "We already typed **fd 60**. And we're 30 short. So let's change the **fd 60** to **fd 90**."

Students also benefited from the measurement tools. Some time after the teacher introduced the turtle turner to the class, one boy and his partner met a difficult turn in a missing-lengths problem (Fig. 2). Neither had developed a strong measurement sense for turns and stopped to consider the problem. They looked at each other and then back at the screen. They used the turtle turner to ascertain the correct command and continued on to finish the problem successfully (Fig. 3).

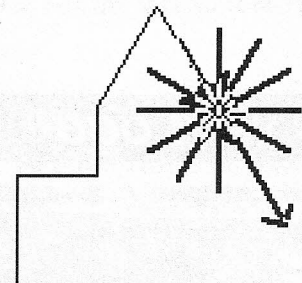


Fig. 2. A missing-measures problem. Students write a procedure to draw the house

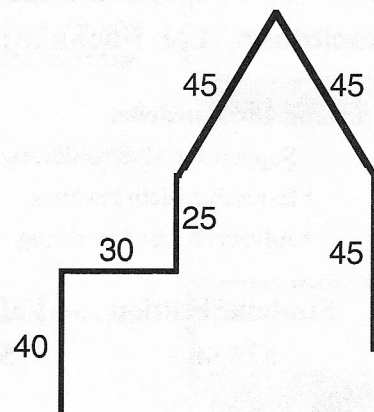


Fig. 3. Using the turtle turner tool to help figure out a turn on the house.

Additional tools allowed students to label the measures of turns and line segment lengths. One pair of students was trying to complete a rectangle. Although the rectangle appeared to be closed, the commands indicated to the teacher that the figure was not a rectangle. Rather than announce this to the pair, the teacher asked them to use the Label Lengths tool to ascertain whether the figure was indeed a rectangle (Fig. 4).

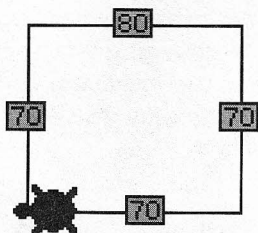


Fig. 4. The label lines tool.

The students accepted the suggestion and realized that the figure was not a rectangle, as the last side drawn was too short. They repaired the situation, using the second side's length as shown by the Label Lengths tool for the input to the final `fd` command.

## 2. Maintain close ties between representations

The structure of the command window appeared to significantly enhance children's use of Logo as a tool for geometric thinking: It is tall and displays more commands than is usual in Logo. Two girls constantly discussed the code and the turtle's path while solving the problems. In an activity called "Coming Home" (Fig. 5), they said, "Try `bk 50`. Because we put `fd 40` in (pointing at the Logo code), but then we had to put in a `fd 10` (pointing first to the code, but immediately then to that section of the path), so it's got to be `bk 50` to get back (gesturing back along the section of length 50)."

The command window also maintains a dynamic link between Logo code and the turtle's drawing. For example, if the commands `fd 100 rt 90 fd 50` are in the command window, and the students change the 100 to 200, the figure immediately is re-

drawn twice as tall. This feature increased students' exploratory behavior. It also encouraged them to connect knowledge of geometric figures and Logo commands. In "Coming Home" they often combined two commands, such as `fd 50 fd 20`, into `fd 70`. Such combination was easy to do, and the turtle's rerunning of the commands showed them their logic was correct.

Students using typical versions of Logo usually "add on" `fd` and `bk` commands, rather than revising commands (Clements & Battista, 1991; Noss, 1984; Weaver, 1991). Such adding on does little to promote the comparison of one quantity to another or to reflect on one's overall activity. *Turtle Math's* structure encourages the latter, which then supports children in comparing one representation of a number ( $50 + 20$ ) to another (70).

## 3. Facilitate examination and modification of code

So, *Turtle Math* helps students think about and change their Logo code in an exploratory fashion, our third principle. As another example, a student was trying to draw his first

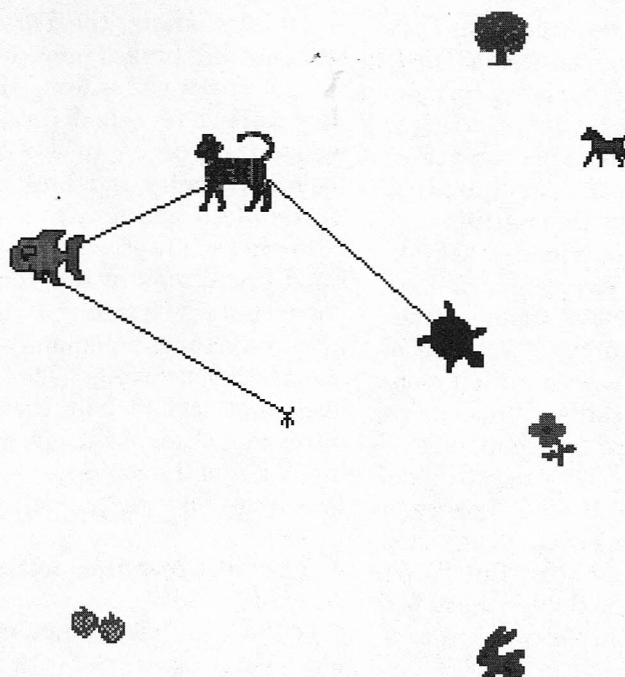


Fig. 5. The "Coming Home" activity. Students teach the turtle to visit each object, then return along the same path. When they reach each object, it grows larger.

equilateral triangle. He typed in `fd 80` for his first side. For the first turn, he said, "`lt`," but he changed his mind and typed `rt 60`. He was distressed with the effect and said, "No, you have to go over," as he pointed to the screen in the area that he wanted the turtle to go. Again, he changed the command, to `rt 120`. So, the *Turtle Math* environment allowed him to easily try different turns to show the effects of the change.

*Turtle Math's* features are synergistic. For example, in "Coming Home," a girl guesses a turn: `rt 45`. She realizes it doesn't line up, and is unsure where to go. She clicks on the turtle turner tool. This shows her that she needs an additional  $15^\circ$  turn to the right. Instead of entering `rt 15`, she deletes the 45 and replaces it with 60. Another girl, in a similar situation, had entered `rt 100` and then used the turtle turner to discover that she had gone  $40^\circ$  too far. She changed the `rt 100` to a `rt 60`.

(continues on the next page)



(continued from page 9)

Such episodes were common. They illustrate several advantages of *Turtle Math*. First, students used the measurement tools instead of resorting to mere trial and error. This helps them keep their attention on quantity. Moreover, it helps them achieve independence in mathematical thinking. Second, it is natural for them to change the previous command, rather than just accumulating a string of commands. This exercises their mental computation skills. It also helps them think of the final command as a single quantity. That is, students who enter `rt 100 lt 40` do not learn about turn measure even though they have achieved their goal. But *Turtle Math* students who change the `rt 100` into `rt 60` do learn about turn measure.

Examination of code in the Teach window also benefited students. One girl created several rectangles of perimeter 200, always by trial-and-error. Having these procedures effortlessly organized and displayed in the Teach window led her to recognize that she could view adjacent sides as a pair whose lengths summed to 100. This resulted in her using, for the first time, a systematic approach in generating other such rectangles.

#### 4. Encourage procedural thinking

Thinking with procedures is one of Logo's big ideas. In most versions of Logo, however, there are two limitations to using procedures. First, students do not learn to define procedures early in their Logo experience, and even when they do, they tend not to do it often. Second, students often see defining procedures merely as a way to "save work." That is, writing a procedure is just a way to "store away" their commands. When they use procedures this way, they often forget the code in the procedures. For example, they recall that their procedure `eqtri` draws an equilateral triangle but they forget *how* it does so.

We found that students used *Turtle Math's* Teach tool to define procedures right from their first day. Developing a working sequence of commands in the Command Center and defining that as a procedure became

their regular way of working with Logo. Like others, the *Turtle Math* students did think of procedures as a way to "save work" at first. However, they were so used to defining procedures, they began to use them as tools for planning and thinking. Also, *Turtle Math* maintains connections between the graphic figures and defined procedures in the Teach window. That is, if a change is made to a procedure, that immediately changes the turtle's drawing. This encouraged students to edit their procedures and observe the effects. They didn't forget the command sequences that procedures were made of.

#### 5. Provide freedom within constraints

These examples show that *Turtle Math* encourages and facilitates both structured activities and open-ended explorations. There were, in addition, several instances of students extending particular projects well past what the teacher had originally proposed. In several of these projects, the provided tools (e.g., rotation and length measurement tools) were critical in scaffolding the students' problem-solving processes and thus facilitating the completion of the project. The activities were critical in allowing children freedom within constraints, a final principle we followed. For example, in one activity students make any face they wish, given the constraint of a fixed perimeter for each face part. This enhanced moti-

vation by encouraging ownership of the problem and allowed children to choose a shape that, to them, "fit" the given perimeter, yet forced the students to think of the shapes they chose simultaneously as parts of a picture and as geometric figures with particular length characteristics.

#### Final Words

We are pleased with the work teachers and students have done with *Turtle Math*. The five principles which we described above have been valuable in organizing this work. We would like to communicate with others about how they use and apply *Turtle Math*. ▲

*Douglas H. Clements is Professor of Education and Julie Sarama Meredith is a Ph.D. candidate at State University of New York at Buffalo.*

**Turtle Math™** is a product of LCSi. It is available for Macintosh and MSDOS Computers.

Call 800-321-5646  
for more information.

#### References

- Clements, D. H., & Battista, M. T. (1991). *The development of a Logo-based elementary school geometry curriculum (Final Report: NSF Grant No.: MDR-8651668)*. Buffalo, NY/Kent, OH: State University of New York at Buffalo/Kent State University.
- Noss, R. (1984). *Children learning Logo programming. Interim report No. 2 of the Chiltern Logo Project*. Hatfield, England: Advisory Unit for Computer Based Education.
- Noss, R., & Hoyles, C. (1992). Afterword: Looking back and looking forward. In C. Hoyles & R. Noss (Ed.), *Learning Mathematics and Logo* (pp. 427-468). Cambridge, MA: MIT Press.
- Weaver, C. L. (1991). *Young children learn geometric and spatial concepts using Logo with a screen turtle and a floor turtle*. Unpublished Doctoral Dissertation, State University of New York at Buffalo.
- Weir, S. (1987). *Cultivating minds: A Logo casebook*. New York: Harper & Row.





# Introducing the next best thing to two months off in the summer.



## Turtle Math and MicroWorlds Math Links from LCSi

After years of development and consultation with teachers like you, LCSi introduces **Turtle Math** and **MicroWorlds Math Links**: two math tools for teachers who want to make math exciting.

**Turtle Math** and **MicroWorlds Math Links** provide a true advantage over any other math software. Each easy-to-use package provides students with an invaluable exploratory environment plus dozens of activities that help them think mathematically. So they learn more about math. And that means increased satisfaction for you.

**Turtle Math**, designed for students in grades 3 – 6, lets students use a collection of activities and challenges in which measurement and geometry is the context for exploring various math concepts.

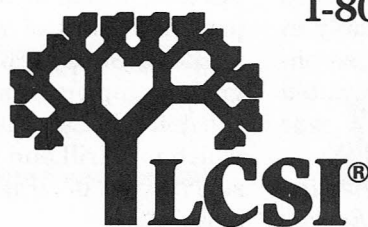
Aimed at students in grades 4 – 8, **MicroWorlds Math Links** is an interactive learning environment that

gives students concrete ways to explore abstract ideas and visualize answers to mathematical questions.

Both packages support the NCTM Standards. **Turtle Math** is available for Macintosh computers; **MicroWorlds Math Links** is available for Macintosh and IBM computers.

If you're interested in exploring a new standard in math teaching tools, why not call us today for a free demo disk. Ask for Helen at:

**1-800-321-5646.**



*And bring a little more sunshine  
into your classroom*



# Book Review



by Carol Sperry

## ***Computer Environments for Children***

by Cynthia Soloman, MIT Press, Cambridge, MA, 1986

I've just returned to my dorm from giving a LogoWriter workshop to teachers from the Vilnius area of Lithuania. Twenty teachers gathered, sharing computers and working in English and Russian LogoWriter. (The Lithuanian version is still in the works and five chapters of *Mindstorms*, so far, have been translated into Lithuanian.) The fascination and enthusiasm were contagious as usual, and the noise level rose in an ordinarily quiet room as teachers helped each other and discovered new techniques and creations.

During the discussion the debate about CAI and Logo came up. The topic is especially provocative in a country that is so energetically pursuing educational reform. The old Soviet system was strictly a matter of transmission of information (often misinformation) in a manner spelled out precisely by the Kremlin. Fifty years of this has taken its toll, but I have seen in the faces of many teachers here during the four summers I've been coming to Lithuania, the desire to move toward a more dynamic teaching and learning method. Most promising is the active Logo community here that was started and supported by Ricardas Alisauskas, Valentina Dagliene, and others.

The debate mentioned above reminded me of so many we've all heard and participated in, and not only in the past. It seemed, therefore, serendipitous to me that the book I wanted to talk about in this review was Cynthia Soloman's *Computer Environments for Children: A Reflection on Theories of Learning and Education*. Though published in 1986, it is viable today in the ongoing struggle to use computers as dynamic learning tools. Her descriptions and critiques of four early research projects lay out clearly the fact that your ap-

proach to the question of how people learn will influence your use of the computer as either a textbook, information-transmitting device, or an expressive medium. The questions Soloman asks are as important and relevant today as they were eight years ago (sorry to say for the learning community).

Soloman tells us of the mathematically based projects of four researchers: Patrick Suppes of Stanford University, who developed the CCC system to provide students with sequenced drill and practice exercises; Robert Davis of the University of Illinois, whose Plato system embodied a Socratic style of intervention; Tom Dwyer and his colleagues at the University of Pittsburgh's Soloworks Project who developed eclectic programming activities around BASIC; and Seymour Papert and the MIT Logo Group with their early Logo work. Soloman succinctly describes each project, but also gives a sharp and balanced critique of each approach to learning theory.

Suppes' drill and practice approach is still popular today in many schools, despite the fact that educators, at least theoretically, acknowledge the existence of different learning styles. It appears to be an "easy way out" for overworked teachers and overcrowded classrooms, and seems to satisfy what the system calls "mastery of basic skills." To me (Soloman is appropriately objective), it reduces mathematics to a computational Skinner Box, providing no entree into the powerful ideas of mathematics. In fairness, Suppes considers his programs supplemental and I suppose the real criticism might go to schools which use drill and practice as their sole method of "transmitting" mathematics.

Davis, as Solomon points out, "...set out to embed the learning of arithmetic in a richer context," and sought to help children realize the connections of arithmetic to algebra, geometry, science, etc. More in sync with

progressive theories of learning, and resonating with what readers of *Logo Update* might call Logo-type philosophy, Davis believes that children learn best "informally and through discovering for themselves." He developed manipulative materials and used everyday experiences to evoke children's curiosity and draw them into the beauty and intricacies of mathematical thinking. Davis was responsible for the Madison Math Project, a large-scale teacher education effort in the 1970's and, now at Rutgers University, continues to work with teachers and children in developing rich approaches to mathematical learning.

Dwyer, a former high school science and math teacher, seems allied with Davis in believing that children will learn if teachers create "conditions favorable to exploration [that] will lead to discovery and effective learning." Dwyer views the computer as an expressive medium and he developed activities in which the child could use the computer as a personal tool. He is refreshing in that he understands that teachers are an essential part of the interaction of school and could enhance their practice with the stimulation and inspiration a computer can provide. An interesting aspect of Dwyer's philosophy is his belief in apprenticeship, based on his own experiences while learning how to fly. He feels that a successful teacher brings the child to a point where she can "solo" and that the computer plays a powerful part in the child's ability to "solo" in her own unique way. This work was done when BASIC was most readily accessible and so Dwyer used it. I don't know what he's up to these days, but I'd love to see him play with Logo.

Soloman worked with Papert in the early seventies, most notably in the Brookline Project. She gives an extensive explanation of Papert's constructivist ideas, his deep involvement with creating a "mathland" that engages children in a profound explo-



ration of mathematics, and how Logo is a most evocative tool to accomplish this. Solomon debugs some of the myths that grew out of superficial readings of Papert's work, including *Mindstorms*, and points out some of the difficulties with implementing such an innovative program. The chapter makes for a good refresher course.

Solomon shares her own incisive and expansive image of the future of educational computing and gives excellent advice for the development of teachers. She understands the need in schools of education for "plans, visions, and competent computer educators" and how essential it is for teachers to grapple with their own understanding of what computers might mean to them and their practice. The time given to teachers to approach these issues and evolve their own understanding and knowledge was woefully lacking in 1986 and remains so today. Solomon describes several models in her chapter on

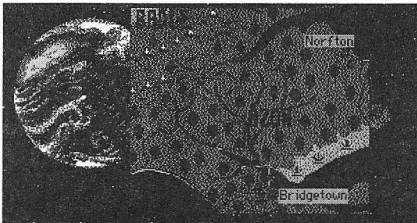
"Computer Educators" that clearly emerge from long experience and give, at the very least, many ideas to adapt or build upon. *Computer Environments for Children* is a small book — only 164 pages — but it is packed with history, theory, vision, and critical viewpoint. It could certainly provide a catalyst for many a long evening's discussion on this topic still so vital in today's learning and teaching community. ▲

*Computer Environments for Children*  
may be obtained from the Logo Foundation. Use the Logo Foundation Response Form on page 15 to place your order.



## THE CRYSTAL RAIN FOREST

The planet Oglo is in trouble. Its rain forests are being destroyed. The king has been poisoned. Only YOU can save them!

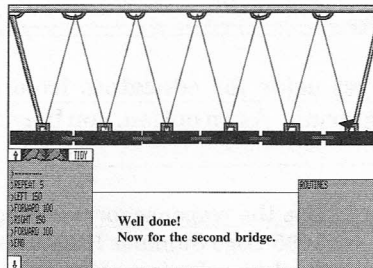


*The Crystal Rain Forest* helps students in grades 3-8 use math to learn about the environment. They hunt for clues in the town, then search for the lifesaving magical crystals deep in the rain forest.

On their quest, they face a series of mathematical puzzles and challenges to solve. They give instructions to robots, guide and rotate shapes to mend bridges, navigate a boat, estimate distances and angles to

connect wires, draw shapes to make nets, change box sizes using simple algebra, and so on.

As a result of these carefully sequenced activities, students learn to use the Logo language. *Crystal Logo*, an easy-to-use version, can be run separately from the adventure, and its command names can be modified.



*The Crystal Rain Forest*, award-winning software from England, is available here as a single user version (\$49.95), as a single version for school use with curriculum materials (\$59.95), and as a building site license (\$250.00).

*PC version requires a 286 or better with VGA and a mouse.  
Mac version requires System 7, and color monitor.*

### Terrapin™

TERRAPIN SOFTWARE, INC.  
400 RIVERSIDE ST. • PORTLAND, ME 04103  
207-878-8200

**1-800-972-8200**

LU 9/94



## What can I do with my old Apple II?

Use it for Logo, enhanced with a special purpose Toolkit!

These Toolkits work with your existing Logo to provide numerous activities and project ideas. Each package includes software and extensive written materials.

These Logo Toolkits are now available from the Logo Foundation at greatly reduced prices. Use the Response Form on page 15 to order.

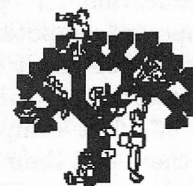
### For use with LogoWriter™

**LogoWriter Hypermedia Tools** originally \$39.95 **now only \$9.95**  
Create hypermedia documents which include graphics, text, and sound.  
Control a laserdisc player (optional).

**LogoWriter Language Arts Tools†** originally \$39.95 **now only \$9.95**  
Contains activities to develop writing skills and encourage self-expression.

**LogoWriter Graph Tools†** originally \$39.95 **now only \$9.95**  
Create bar graphs, pie charts, line graphs, and pictograms.

**LogoWriter for Special Needs** originally \$59.95 **now only \$9.95**  
Includes single key interface, large type word-processing, and other supports for students who are physically challenged or have learning disabilities.



†also available for use with Apple IIgs or MSDOS LogoWriter

### For use with Terrapin™ Logo or Logo Plus™

**Logo Data Toolkit** originally \$49.95 **now only \$9.95**  
Take surveys, create data bases, generate graphs and charts.

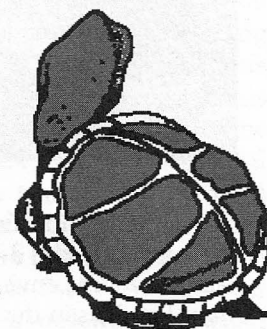


The Logo Foundation and the St. Paul Public Schools announce . . .

## Logo St. Paul

### The 1995 Logo Summer Institutes

Over the past fourteen years the St. Paul Logo Project has provided a comprehensive professional development program for hundreds of elementary and secondary school teachers. The cornerstone of this program has been the **Logo Summer Institute**, an intensive one-week workshop which provides for an immersion in Logo theory and practice. The individualized approach of the Logo Summer Institute accommodates experienced Logo users as well as novices.



A limited number of places are being set aside for educators from outside the St. Paul Public Schools. The registration fee includes all workshop materials. As an option, you may receive three graduate quarter credits from Hamline University.

**When:** June 26 - 30 or  
August 21 - 25  
**Where:** St. Paul, Minnesota  
**Cost:** \$490 per person  
\$111 for graduate credit

Use the response form on page 15 to request registration materials for the 1995 Logo Summer Institutes and to obtain more information about the St. Paul Logo Project. If you are a teacher in the St. Paul Public Schools these registration procedures and fees do not apply to you. Instead, contact Ms. Geraldine Kozberg at 360 Colborne Street, 228-3631.

# Logo Foundation Response Form

- Enter my free subscription to *Logo Update*.
- Send me the complete list of Logo Foundation Publications.
- Send me more information and registration materials for the 1995 St. Paul Logo Summer Institutes described on page 14.

Enter my order for:

	<u>quantity</u>	<u>amount</u>
Items pertaining to "A New Approach" on page 1 and "Logo Tool Box" on page 3:		
<input type="checkbox"/> <i>Computadoras en la Escuela Pública Costarricense</i> by Clotilde Fonseca \$12.00.....	_____	\$ _____
<input type="checkbox"/> <i>A Revolution of Opportunities</i> \$15.00.....	_____	\$ _____
<input type="checkbox"/> "Advancing Logo" and "A National Transformation", reprinted from <i>Logo Update</i> \$1.25...	_____	\$ _____
<input type="checkbox"/> <i>Event Programming</i> by Michael Tempel \$9.50.....	_____	\$ _____
<input type="checkbox"/> <i>Event Programming</i> Samples and Tools Disk \$5.00.....	_____	\$ _____

The book reviewed by Carol Sperry on page 12:

<input type="checkbox"/> <i>Computer Environments for Children</i> by Cynthia Soloman \$13.95 .....	_____	\$ _____
---	-------	----------

Logo Toolkits described on page 14:

<input type="checkbox"/> <i>LogoWriter Hypermedia Tools</i> \$9.95.....	_____	\$ _____
<input type="checkbox"/> <i>LogoWriter Language Arts Tools</i> \$9.95 <input type="checkbox"/> IIe <input type="checkbox"/> IIgs <input type="checkbox"/> MSDOS.....	_____	\$ _____
<input type="checkbox"/> <i>LogoWriter Graph Tools</i> \$9.95 <input type="checkbox"/> IIe <input type="checkbox"/> IIgs <input type="checkbox"/> MSDOS.....	_____	\$ _____
<input type="checkbox"/> <i>LogoWriter for Special Needs</i> \$9.95.....	_____	\$ _____
<input type="checkbox"/> <i>Logo Data Toolkit</i> \$9.95.....	_____	\$ _____

And don't forget:

<input type="checkbox"/> <i>The Children's Machine</i> by Seymour Papert \$22.50.....	_____	\$ _____
<input type="checkbox"/> <i>Mindstorms</i> (2nd Edition) by Seymour Papert \$13.00 .....	_____	\$ _____

Tax deductible contribution to the Logo Foundation \$ \_\_\_\_\_

**Total** \$ \_\_\_\_\_

**Name** \_\_\_\_\_

**Organization** \_\_\_\_\_

**Address** \_\_\_\_\_

**City** \_\_\_\_\_ **State** \_\_\_\_\_ **Zip** \_\_\_\_\_ - \_\_\_\_\_

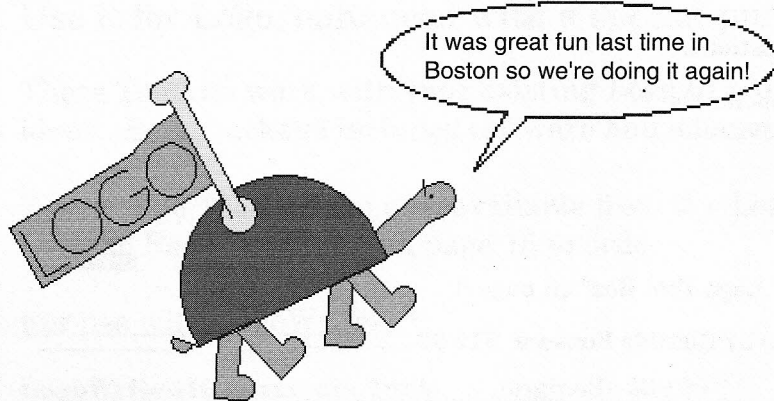
**Day Phone** (    ) \_\_\_\_\_ **Evening Phone** (    ) \_\_\_\_\_

Please enclose payment or a school purchase order.

Overseas orders require additional shipping charges. Please inquire before ordering as the amount depends upon destination and carrier.

## Logosium '95

Friday, June 16, 1995



- A full day of Logo discussions, sharing sessions, and presentations
- Sponsored by ISTE's SIG-Logo and the Logo Foundation
- A pre-conference activity at NECC '95, Baltimore

Mark the date on your calendar now and stay tuned for further details!

### LOGO USERS GROUPS

Long Island Logo Users Group

Contact: Marilyn Tahl  
516 333-4018 (evenings)  
516 627-8110 (days)

Los Angeles Logo Users Group

Contact: Carolina Goodman  
Campbell Hall  
4533 Laurel Canyon Blvd.  
North Hollywood, CA 91607  
818 980-7280 ext.234

Logo Anonymous

Contact: Marian Rosen  
Conway School  
9900 Conway Road  
St. Louis MO 63124  
314 993-2878

New York Logo Users Group

Contact: The Logo Foundation  
212 765-4918

Philadelphia Logo Users Group

Contact: Mel Levin  
Prince Hall School  
Godfrey and Gratz Avenues  
Philadelphia PA 19141  
215 276-5369

# Logo Foundation

250 West 57th Street • New York, NY 10107-2228

Nonprofit Org.  
U.S. Postage  
**PAID**  
New York, NY  
Permit #6378

