

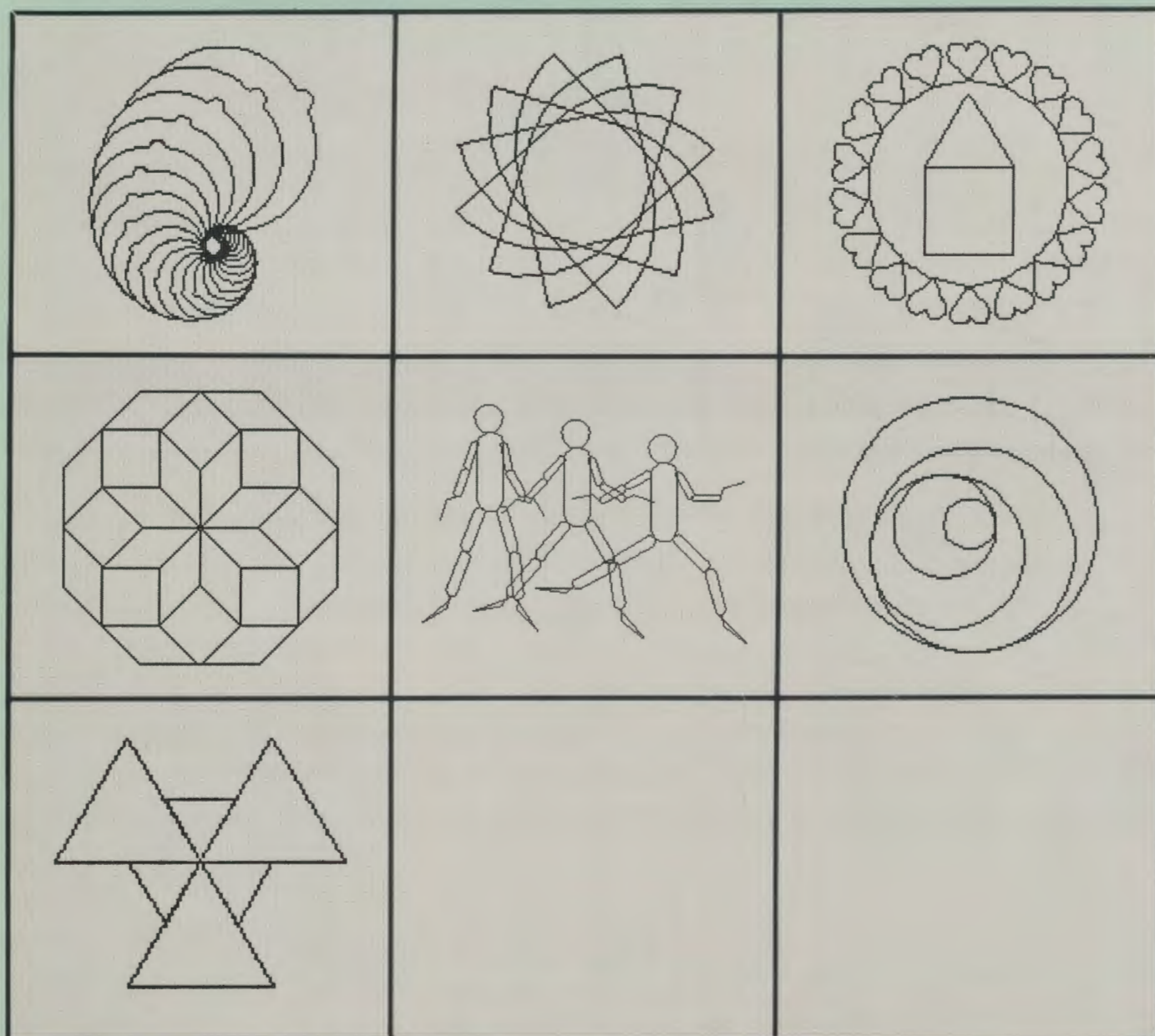
# LOGO EXCHANGE

*The Magazine for LOGO Activities Worldwide*

ISSN 0888-6970

VOLUME 5 NUMBER 7

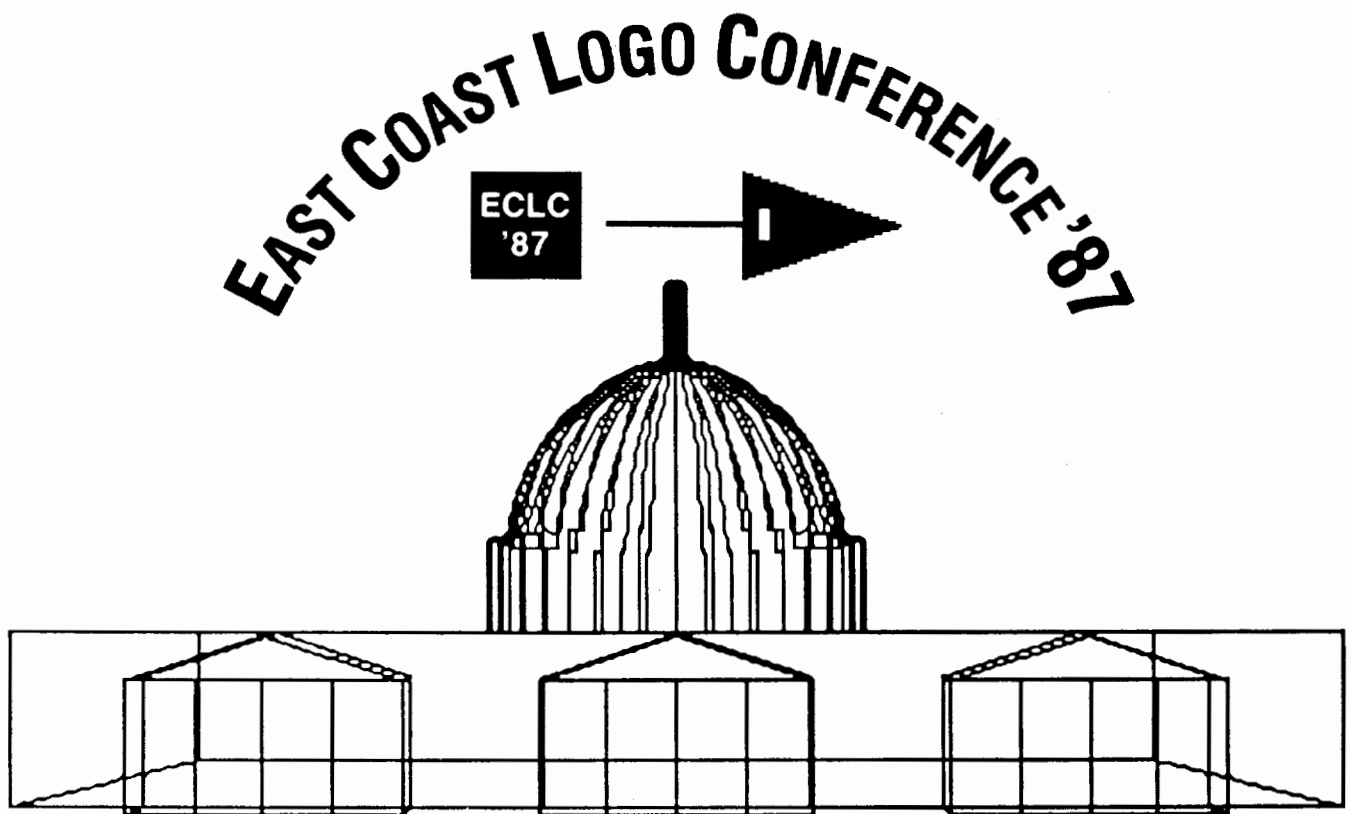
MARCH 1987



Second Class  
Pending  
Westport, CT and  
Add'l Mailing Offices

Meckler Publishing Corporation

The University of Virginia, in association with  
Meckler Publishing and *Logo Exchange* magazine, presents ...



*A Conference for Educators Using Logo*

April 2-4, 1987  
Stouffer Concourse Hotel,  
Arlington, Virginia

- 22 Sessions
- 6 Warmup Activities
- 65 Speakers
- Exhibits
- Special General Sessions
- Receptions and Socials

Featuring  
Seymour Papert  
Dan and Molly Watt

*For Program and Registration  
Information, write:*

**ECLC '87**  
11 Ferry Lane West  
Westport CT 06880  
or Call: 203/226-6967



# LOGO EXCHANGE

*The Magazine for LOGO Activities Worldwide*

ISSN 0888-6970

VOLUME 5 NUMBER 7

MARCH 1987

## Editor-in-Chief

Tom Lough

## International Editor

Dennis Harper

## Senior Editors

Glen Bull

Steve Tipps

## Field Editors

Eduardo Cavallo

Patricia Dowling

Michael Friendly

Anne McDougall

Richard Noss

Fatimata Seye Sylla

Hillel Weintraub

## Senior Contributing Editor

Robb Muir

## Contributing Editors

Elaine Blitman

Regina Bowden

Doug Clements

Paula Cochran

Bill Craig

A. J. (Sandy) Dawson

Barbara Elias

Anne Cairnes Federlein

James Fry

Judi Harris

Barbara Jamile

Jim McCauley

Rebecca Poplin

Barbara Randolph

Linda Sherman

## Managing Editor

Anthony Abbott

## Advertising Director

Jean-Paul Emard

## Publisher

Alan M. Meckler

*Logo Exchange* (ISSN 0888-6970) combines *National Logo Exchange* and *International Logo Exchange* and is published monthly September through May by Meckler Publishing Corporation, 11 Ferry Lane West, Westport, CT 06880, (203) 226-6967.

Copyright © 1987 by Meckler Publishing Corporation. All rights reserved. No part of this publication may be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage retrieval system, without permission in writing from the publisher.

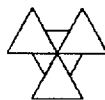
Application to mail at Second Class postage rates is pending in Westport, CT and additional mailing offices. POSTMASTER: Send all address changes to *Logo Exchange*, Dept. VV, P.O. Box 3000, Denville, NJ 07834

## CONTENTS

|                                     |    |
|-------------------------------------|----|
| From the Editor                     | 2  |
| Logo: Syntax and Sculpture          | 3  |
| Tipps for Teachers                  | 4  |
| Teaching Tools                      | 7  |
| TO BEGIN :COORDINATES               | 11 |
| Logo LinX                           | 13 |
| MathWorlds                          | 14 |
| SpecialTalk                         | 16 |
| Testudinal Testimony                | 18 |
| Logo Disserts                       | 20 |
| LXionary                            | 21 |
| Teacher to Teacher                  | 22 |
| Welcome to LogoWriter Again         | 23 |
| The Adventures of Jacques and Elsie | 26 |
| Q and A                             | 27 |
| IntLXual Challenges                 | 28 |

## International Logo News

|                 |    |
|-----------------|----|
| Global Comments | 29 |
| Africa          | 30 |
| Australia       | 31 |
| LogoPals        | 32 |



**Cover:** The triangle design on this month's cover was contributed by Anna P. Olson, of Neenah, Wisconsin, a former elementary school teacher. She developed the design as part of a Logo spelling program for her son, Paul.

Subscription: \$29.95 per year. Subscribers outside the USA please add \$15.00 for surface mail or \$50.00 for airmail. Orders from North and South America should be sent to *Logo Exchange*, Dept. VV, P.O. Box 3000, Denville, NJ 07834; elsewhere to Meckler Publishing, 3 Henrietta Street, London WC2E 8LU, UK, (tel.) 01-240-0856.

## From the Editor

by Tom Lough

While listening to a presentation recently, I heard a speaker say, "If you are driving on a trip from Charlottesville to Washington at night, the headlights of your car don't have to reach all the way to your destination. They just have to light the path immediately ahead."

Many of you know me well enough by now to guess correctly that I have no idea what else the speaker said from that point. As soon as I heard the above sentence, I was off on a thinking trip of my own, oblivious to everything else around me.

What a fascinating metaphor of many challenges we face in our Logo work, particularly when we are doing something for the first time! But the idea is not worthwhile without a goal of some sort.

One of my favorite stories about the importance of having a goal is about the person who rushed up to the airline counter, began placing hundred dollar bills on the counter, and feverishly asked the clerk, "Quick!! Give me a ticket!"

"But where do you want to go?"

"I don't know! But that doesn't matter. Just give me a ticket!"

Even though Logo allows us to work without a completely specified goal in mind, it is usually helpful if we have a general idea of what we are trying to accomplish.

For example, several months ago, I was asked by a colleague to write a Logo procedure which would accept any word and return all possible combinations of the letters of the word. Since there was no rush on getting the procedure written, I steadfastly refused to look up a solution, and decided to develop my own "from scratch." (There are a number of solutions available, however. See the April 1985 issue of *Microquests*, for example.)

At first, I tried to write the Logo procedure without much forethought. Just shift the letters somehow, or something like that, I guessed. That didn't work, of course.

Then, I decided to play with the letters of short words using pencil and paper. (These are ancient writing tools used for various purposes before the microcomputer was invented.) By writing the letters of the word CAT in all possible combinations, I was able to get a feel for some possible patterns which would give me the results I wanted.

Next, I played around with a four letter word, and then a five letter one. Soon, I could draw the diagram of a system of shifting and mixing of the letters which could be done in an organized way to produce all possible combinations.

At this point, I began to understand how it might be possible to translate this diagram into something which Logo could understand. However, try as I might, I could not find the proper sequence of recursion calls and letter shifts.

Finally, I got the idea of writing a procedure to produce the letter combinations just for three letter words. Since I had played with CAT a lot, I found this procedure easy to write.

Next, I wrote a procedure which accepted only four letter words and produced all combinations. Then came a procedure for five letter words. But there were problems. The sequences still were not coming out right.

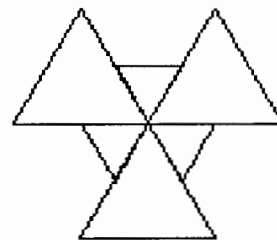
More work, more thought, more playing around with the letters and combinations. Finally, eight months after I accepted the challenge, the idea I needed came to me, and I was able to write the procedure to do the job!!

Although this project took me a long time, I was always able to make a little progress whenever I took time to work on it. The reason was that I could see a little bit of the job which had to be done next, the part which had to be fixed, the idea which had to be understood.

And, since I knew what I wanted to accomplish (but couldn't see how to get the job done at first), each "little piece of work" brought me closer and closer to my goal.

No, we don't have to be able to see how to get to where we want to go. We must decide what we want to do, and then pay attention to the road just ahead. It may be slow, it may be difficult, but it usually gets us there.

FD 100!

Permission to photocopy for internal or personal or the internal or personal use of specific clients is granted by Meckler Publishing Corporation for libraries and other users registered with the Copyright Clearance Center (CCC), provided that the stated fee is paid per copy directly to the CCC, 21 Congress Street, Salem, MA 01970. Special requests should be addressed to the publisher. The article fee code for this journal is 0888-6970/87 \$5.00 + 0.

# Logo: Syntax and Sculpture

by Louise Robinson

One predictable task of the graduate students enrolled in my "Logo in the Curriculum" course at Indiana University of Pennsylvania is to fill out the ubiquitous 3" by 5" index card. Along with the usual demographic information, the students are asked to finish the phrase, "Logo is . . .," and to identify what they hope to accomplish as a result of the course. This information is helpful to me as I prepare to dispel myths about Logo and help all students achieve their individual goals. But I was not prepared for the response of one particular student.

John wrote, "Logo is ... a computer language for children," and indicated that he wanted to incorporate Logo into the elementary art education curriculum. But then he said that he was a marble sculptor, and wanted to find out if Logo could be used as a sculpture design tool.

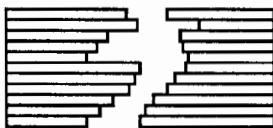
You can imagine my reaction when I read his card! "Aha! Another person who thinks Logo is just for kids! Oh, no! What do I know about sculpting? How will I answer his philosophical questions about turtle graphics and art? Would I be able to help John attain his goals?"

## Plunging In

The semester proceeded as usual, starting at the concrete level, playing turtle, and ending at an abstract level, with the Logo shape editor utility. Along the way, some very interesting things happened.

John was excited about Logo from the start. The language challenged his personal teaching philosophy and affected his perceptions of sculpting. His contributions to class discussions centered around how he saw Logo helping children with two-dimensional thinking and enhancing art.

At the same time he was taking my course, he was nearing the end of his work toward a Master of Arts degree. His sculpting was done with marble slabs of varying lengths, stacking them one upon the other, creating fascinating arrangements. Since each slab was several inches thick, his pieces sometimes reached a height of 6 feet!



His thesis was to be an exhibit of a series of his pieces at the University museum. John named his series "Syntax," partly because he always had difficulty with English grammar, and partly because he saw a conceptual similarity between written composition and marble composition. He describes his process of sculpting with marble slabs as the same idea behind building a sentence or writing a Logo program. The nouns, verbs, and adjectives combine to make an English statement. The Logo commands combine to form a procedure; any procedure can be combined into any other procedure. His work with marble slabs is also the combining of such parts. Therefore, he found his work with Logo to be synonymous with his sculpting.

John's latest sculpting work began to explore new directions. He began to mix different colors of marble, negative space, and differing degrees of roughness in a single piece's texture. He said his early two-piece sculptures showed a push / pull between emotional questions he had at the time. The classic tension was between being an artist and teaching art, being a student and being a teacher.

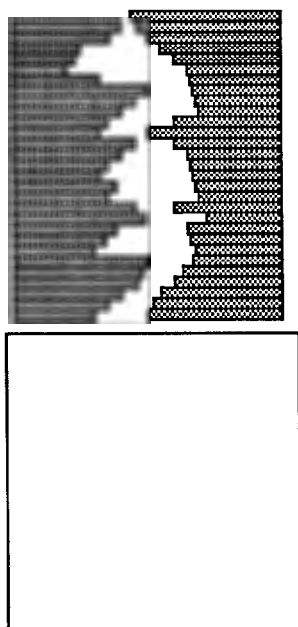
## Building Block Project

The major course requirement was to develop a Logo project that demonstrated the use of superprocedures and subprocedures, was streamlined, and could be used as a teaching aid. John chose to have his project draw the design of one of his beautiful marble creations.

When John presented his project to our class during the semester, he enthusiastically reported, "Logo has opened a whole new world for both my teaching and my sculpting." He was very eager to introduce his art students to Logo, because it had enabled him as an artist to experiment with different colors and sizes of marble, not in the art studio, but in the computer lab. Also, he could experiment easily with many different slab arrangements without having to move them into position physically.

Best of all, John invited the whole class to the showing of "Syntax." As we entered his exhibit, we saw his Logo project along with his breathtaking sculpture. He had hidden the computer beneath a table, so that only the monitor and its display was visible. A series of text screens introduced John, his work, and his exhibit. Then the design for one of his pieces was created on the screen, slab by slab.

It was an exhilarating moment, watching the turtle draw each slab, one upon the other, in the shadow of the actual sculpture. The class members watched in awe and astonishment. I could not help but imagine each slab representing an additional piece of information or knowledge falling into place as the fascinating design developed on the screen.



The Logo sculpture experience was a fitting end to the course. I'm sure the occasion will remain in my students' memory long after their ability to write Logo programs has faded. And I'm certain that it will have a profound impact on John's sculpture for the rest of his life.

### Questions, Questions

I see my teaching task as creating a Logo environment that allows students to reach their individual goals through guided discovery. As the Logo "coach," I'm sometimes baffled by my students' questions, however. I'm not always certain that I will have the answers. By having John as a student, I have been able to reinforce my notion that I need not always have the answers. In John's case, I didn't answer his questions; he answered them himself.

But John helped me learn new things about Logo and about marble sculpting as well. He also gave me more confidence in being able to meet the needs of my students, regardless of their personal goals with Logo. Now, who was it that said Logo was a computer language for children?

---

*Louise Robinson is currently an associate professor of educational computing in the Department of Professional Studies in Education at Indiana University of Pennsylvania.*

## Tipps for Teachers

by Steve Tipps

### The Logo Bookshelf

Writing is a difficult in the best of times. Writing about Logo is a more than difficult task. The spirit to convey about Logo as a thinking tool and as a medium for exploration is as important as the content. The Logo bibliography now contains hundreds of entries. Out of the hundreds, a few have a special place on my Logo bookshelf because they hold to the spirit of Logo.

#### In the Beginning

The importance of *Mindstorms* as the philosophical treatise for Logo does not need repeating. Each time I pick it up to hunt for a favorite phrase or quote, I get caught up again in the excitement of its first reading. It contains a spontaneous zest that sometimes hides the deeper meanings until the third, fourth, or fifth reading. I wonder if Seymour Papert created it in a month-long writing frenzy or whether it was carefully constructed over a long time by piecing together speeches, papers, and fragmentary ideas. My imagination holds the former is true.

Hal Abelson's books, *Logo for the Apple II* and *Apple Logo*, were also there in the beginning when we had little else.

I read and typed programs from each chapter. Sometimes I was bemused, occasionally confused, but always entertained by the presentations. The covers of my copies are now destroyed, and the spiral binders are bent and unsprung as testament to frequent use.

Five years ago, practically nothing else was available about Logo. The first activity book was Dan Watt's *Learning with Logo*. We thought we might never get it because the book was so long coming off the press. I remember the night in May, 1983, at the National Educational Computing Conference (NECC 83) in Baltimore when Dan got an enthusiastic round of applause when he held up the first copy. Finally, a resource designed for student and teacher activities was available.

Soon afterwards came the *Turtle's Sourcebook* by Jim Muller, Donna Bearden, and Kathleen Martin, also full of practical ideas for children using Logo. It was so popular with one book borrower that mine never returned...

Another book I peruse is *Thinking about [TLC] Logo* by John Allen, Ruth Davis, and John Johnson. Except in talking to John Allen on the phone once or twice, I don't know the authors personally. However, I don't they would be insulted by my saying that this little book is unique, wild, hilarious, and bizarre. It contains serious computing projects, but the text is Logo as might be done by the editorial staff of *Mad* magazine or *National Lampoon*, with cartoons in the style of Gary Larsen's "Far Side." You may need to be a little twisted yourself to enjoy all the puns and jokes, but I can't read



## The best way to introduce students to Logo is to

# == SEE LOGO ==

One reviewer said: As I proceeded, I kept thinking "What a great idea!" or "Why hasn't someone put this in before?"

Well now someone has. And the result is SeeLogo — a friendlier yet more powerful Logo graphics language that lets kids do more sooner. SeeLogo is not a full-featured Logo, but what it does it does well. For many children, the enjoyment of building Logo graphics is all too often impeded by the tedious typing and editing of commands and procedures. SeeLogo minimizes this, making it especially appropriate for beginning students (and teachers!) who can more quickly become involved in intriguing problem-solving & graphics challenges after relatively little instruction. But even if your students already know Logo, you'll still want to benefit from SeeLogo's many new features. And, because SeeLogo's other commands are 100% Apple and Terrapin Logo compatible, almost no relearning is required.

- FRIENDLIER ERROR MESSAGES PROVIDE MORE CONSTRUCTIVE HELP.
- EDITING IN THE DIRECT MODE IS FASTER, EASIER.
- GRID SYSTEM COMMAND ALLOWS THE USER TO SEE THE ON-GRAPH LOCATION AND X, Y COORDINATES OF THE TURTLE AT ANY TIME.
- PLACING TEXT ANYWHERE ON-SCREEN MAKES SEELOGO BETTER FOR STORY WRITING.
- HOW MANY TIMES HAVE YOU WANTED TO UNDO YOUR LAST STEP? 'UNDO' DOES IT!
- GRAPHICS CAN BE DEFINED AS PROCEDURES BY SIMPLY NAMING THEM.
- UNLIMITED BACKUPS AT \$8.00 EACH MAKE SEELOGO EXTREMELY AFFORDABLE TO ADOPT.

Best of all, all this "more" is available for less. SeeLogo costs just \$59.95 and is available for free, no-risk 30-day examination. We'd like you to see for yourself just how good SeeLogo can be in your classroom. Call 800-922-0401 to order today!

**K-12 MicroMedia Publishing**  
6 Arrow Rd., Ramsey, NJ 07446



"Hi, I'm Cecil, the new turtle in town. I'm 100% Apple and Terrapin Logo compatible and a whole lot more to boot!"

### NEW PRIMITIVES

- CREATE/SET Changes the shape of the turtle.
- GROWBY Increases the size of the graphic by any percent.
- HALF/QUARTER/THIRD Reduces the size of the graphic.
- TIME Times the intervals of action procedures.
- SPIN Allows for rotation.
- STAMP Replicates the turtle anywhere on the screen.
- MIRROR Creates the mirror image of the graphic.

*Logo graphics for the rest of us.*

a page without laughing or groaning. Unfortunately, the book disappeared when the computer and Logo version (TLC Logo) it was written for never made it out of the warehouse. If you can find a copy, grab it quick.

### Then Came More

After two years in which the Logo bookshelf contained only a few books and the *National Logo Exchange* newsletters, there came an explosion of Logo writing. Some of the writing was spirited; however, much of it was vulgar, fill-in-the-blank exercises and recipes. Textbooks and activity books were particularly guilty of this. Lists of commands for each grade level and projects which control the children lacked the essential Logo spirit.

The difficulty of maintaining this spirit with student activities hit home when Glen Bull, Tim Riordon, and I were working on *Nudges*. We three turtle-teers set about saving the world for Logo, or vice versa. A prime argument was about how much information to provide and how much to withhold. How much of a hint was a *nudge*? Should we provide entire procedures as tools? Should students develop everything from scratch? What should be in the teacher's manual? The final *Nudges* was not really what we had in our hearts in the beginning. However, I refer to it frequently and repeatedly. Many students, both adult and children, have used it.

The *Nudges* authors were part of a seven-person team organized by Tom Lough originally to write a series of books about IBM Logo for Holt, Rinehart and Winston. Another book produced by the team was *Beyond Mindstorms*. Despite its title, the book has several very good chapters which describe practical aspects of Logo in a classroom. Joyce Tobias, Sharon Burrowes, Jerry Short, and Tom developed an outline for a Logo based curriculum without falling into the traps of prescriptive activities or recipes. Unfortunately, the book all but disappeared when the IBM PCjr production was discontinued. Holt, Rinehart and Winston should bring 1000 of these books to the East Coast and West Coast Logo Conferences and sell them for \$5.00 each. The real shame is that IBM Logo is an excellent implementation which I still use regularly. We could never convince anyone at IBM that \$175 was too much to pay for a Logo in the school market.

### And More

Two books by David Thornburg, *Beyond Turtle Graphics* and *Exploring Logo without a Computer*, are superior examples of the spirit of Logo. Alison Birch wrote *The Project Book* for Terrapin Logo, which contains open ended projects. *Teaching with Logo* by Molly and Dan Watt contains vignettes of classroom use. Teachers who want models of sensitive Logo collaboration will enjoy the inter-

actions. Logo is clearly a vehicle for thinking in the stories.

*Beginning with Logo* has been published after a gestation period slightly longer than that of an elephant. I know what Glen Bull and I wanted to happen in the book, but I can't judge if the spirit we intended is there. We have tried to provide an exploration of Logo with emphasis on powerful concepts. Many parts of the book originated from the first two years of our columns in the *National Logo Exchange*.

Anyone interested in mathematics must procure issues of *Microquests*. The variety and depth provided by Kathleen Martin and Donna Bearden are astounding. Unfortunately, the periodical is no longer being published. Some back issues are available.

Another valuable resource is *Logo in the Schools*, a special issue of *Computers in the Schools*. It contains a range of Logo articles from the Logo language itself to classroom applications. A special issue such as this needs to be developed every year to track current thinking about Logo from a variety of perspectives. The wide range of contributors makes this compilation especially refreshing.

### Why I Love Brian Harvey

Not being able to attend Logo '86 meant that I missed Brian Harvey '86. I wanted to hear firsthand how he felt about the Logo revolution going awry and away. Brian, the H. L. Mencken of Logodom, always has a barbed observation. His barbs are dipped with equal parts of chocolate and arsenic. I do not always agree with him, but I always think about what he says. I like him because he challenges me to think.

But I love Brian Harvey because he has written two volumes of *Computer Science Logo Style*. *Volume 1 Intermediate Programming* takes up where Abelson's language books leave off. The exercises are challenging examples of programming with Logo. Hand this book to your computer science friends and get some respect!

*Volume 2 Projects, Styles, and Techniques* came for Christmas. Not since *Mindstorms* has a book been such a pure embodiment of Logo. This is the book Glen Bull and I dreamed about five years ago: interesting extended projects presented with the thinking behind a solution. The fun and adventure of Logo are intact, but Logo is not trivialized. The book does more than use Logo as a language; the spirit of Logo permeates the book. Brian may not know where the Logo revolution went; I find the revolution doing well in his books. I can hardly wait for Volume 3.

### A Wish List

I have some spots left on my Logo bookshelf. For instance, Paul Goldenberg suggested a Logo language book and a Logo algebra book several years ago. I have a place for both. I understand I won't have to wait much longer for the language book! Tom Lough should write a laboratory book of physics microworlds in his spare time. A yearly collection of Logo articles is needed. Dennis Harper is working on a Logo

anthology or reader which may serve this need. I don't have Sylvia Weir's book, but I am holding a place for it. Richard Noss also has a new book which I want. And I am always ready to be delighted by something new by David Thornburg.

The books on the wish list and those I use and have used regularly are special. They work toward the Logo spirit as well as presenting Logo information. They are written in Logo.

### The Bookshelf

Abelson, Harold. *Apple Logo, Logo for the Apple II*. New York: BYTE Books/McGraw-Hill, 1982.

Allen, John, Ruth Davis, and Jerry Johnson. *Thinking about [TLC] Logo*. New York: Holt, Rinehart and Winston, 1984.

Bearden, Donna, Kathleen Martin, and Jim Muller. *The Turtle's Sourcebook*. Reston, VA: Reston Publishing Company (Prentice-Hall), 1983.

Bull, Glen, Steve Tipps, and Tim Riordon. *Nudges: Apple Logo Projects*. New York: Holt, Rinehart and Winston, 1985.

Birch, Alison. *The Project Book*. Cambridge, MA: Terrapin, 1986.

Harvey, Brian. *Computer Science Logo Style. Vol 1: Intermediate Programming*. Cambridge, MA: MIT Press, 1985.

Harvey, Brian. *Computer Science Logo Style. Vol 2: Projects, Styles, and Techniques*. Cambridge, MA: MIT Press, 1986.

Maddux, Cleborne (Ed.) *Logo in the Schools*, Haworth Press, 1985.

*Microquests* back issues. For information, write to Kathleen Martin, 1015 Cimarron Circle, Roanoke, TX 76262.

Papert, Seymour. *Mindstorms*. New York: Basic Books, 1980.

Thornburg, David. *Beyond Turtle Graphics*. Menlo Park, CA: Addison-Wesley, 1986.

Tipps, Steve and Glen Bull. *Beginning with Logo*. Prentice-Hall, 1987.

Tobias, Joyce, Sharon Burrowes, Jerry Short, and Tom Lough. *Beyond Mindstorms: Teaching with IBM Logo*. New York: Holt, Rinehart and Winston, 1985.

Watt, Daniel. *Learning with Logo* (for four Logo versions). New York: BYTE/McGraw-Hill, 1983.

Watt, Molly and Dan Watt. *Teaching with Logo*. Menlo Park, CA: Addison-Wesley, 1986.

---

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX, and has been involved with Logo since 1982. He conducts Logo workshops for school systems throughout the United States, and is a popular conference speaker. His CompuServe number is 76606,1623.



## Teaching Tools

by Glen Bull and Paula Cochran

### Plotting Tools: Part I

Has the average amount of rainfall in the Amazon rain forest changed over the last 20 years? What is the rate of increase in the gross national product over the last decade? How does the temperature of a cup of coffee change over a 15 minute period?

These are questions which might be asked in a class in science, geography, or social studies. Frequently a graph is used to display the answer. A graph can present a large amount of information in a small space, but frequently students only glance at the picture and concentrate on the text.

Projects in which students create their own graphs can help make graphs in books easier to understand. The computer can provide tools to produce graphs. There are a number of excellent software packages, such as PFS Graph, Chart Master, and Cricket Graph, which can be used to create graphs of various kinds.

Charts and graphs can also be produced with Logo. This is particularly advantageous when the data plotted has been produced by another Logo procedure. Tools for producing bar graphs were discussed in "Chart Tools" (*Logo Exchange*,

September 1986). The graphing tools discussed this month can be used to plot a series of points connected by lines. Several kinds of data can be graphed with plotting tools: data typed in manually, data generated by the computer, or data acquired with external sensors.

In this month's column, four variations of a single PLOT procedure will illustrate how the first two types of data can be graphed with plotting tools. Next month's column will illustrate how data acquired with external sensors and probes can be plotted. With minor variations and modifications, the same tool can be used for a variety of different purposes.

#### The Basic PLOT

For example, let's examine the cumulative number of books read by a class over a six-week period.

| Week | Total Books Read |
|------|------------------|
| 1    | 8                |
| 2    | 17               |
| 3    | 26               |
| 4    | 43               |
| 5    | 60               |
| 6    | 69               |

## Key Reference Sources

### LOGO EXCHANGE RESEARCH DIRECTORY

Edited by Regina H. Sapona

Price: \$9.95 (paper)

ISBN: 0-88736-085-8 (1986)

### NATIONAL LOGO EXCHANGE INDEX (volumes 1-4 (1982-1986))

Compiled by Regina H. Sapona

Price: \$9.95 (paper)

ISBN: 0-88736-0904

### Order Form

Please enter orders for the following:

- ☐ LOGO EXCHANGE RESEARCH DIRECTORY @ \$9.95  
☐ NATIONAL LOGO EXCHANGE INDEX (1982-1986) @ \$9.95

☐ Payment enclosed (required for all personal orders, which must be prepaid by personal check or credit card).

Name \_\_\_\_\_  
 Organization \_\_\_\_\_  
 Address \_\_\_\_\_  
 City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

☐ Please bill my organization Purchase Order No \_\_\_\_\_  
 Authorized Signature \_\_\_\_\_



Meckler Publishing  
 Corporation  
 11 Ferry Lane West  
 Westport, CT 06880  
 203-226-6967

☐ Credit card payment (circle one):

AMEX MasterCard

VISA Diners Club

Card No. \_\_\_\_\_

Exp. Date \_\_\_\_\_

Cardholder's Signature \_\_\_\_\_

Here is a basic plotting tool which can be used to plot the cumulative total of books read. A setup procedure will also be needed, to position the turtle at the left-hand side of the screen before plotting begins.

```
TO SETUP
CS PU LT 90
FD 140 RT 90
END
```

```
TO PLOT1 :NUMBERS
IF EMPTY? :NUMBERS [ STOP ]
SETPOS LIST (XCOR + 40) (FIRST :NUMBERS)
PD
PLOT1 BUTFIRST :NUMBERS
END
```

In these and subsequent procedures, the following modifications are required for Terrapin Logo:

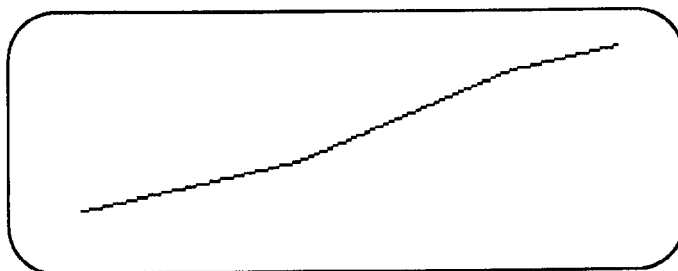
1. In the SETUP procedure, change CS to DRAW
2. In the PLOT1 procedure,
  - a. substitute EMPTY? for EMPTY?
  - b. omit brackets around STOP
  - c. substitute SETXY for SETPOS LIST

```
TO PLOT.1A :NUMBERS
IF EMPTY? :NUMBERS STOP
SETXY (XCOR + 40) (FIRST :NUMBERS) PD
PLOT.1A BUTFIRST :NUMBERS
END
```

The tool can be used to plot any list of numbers:

```
?SETUP
?PLOT1 [8 17 26 43 60 69]
```

The plot produced looks like this:



Creating a Chart Symbol

The graph would be easier to read if there were a symbol such as a block or circle at each point plotted. Here is a procedure to make a small solid block. It uses two subprocedures, STROKE and MOVE.

```
TO DRAW.BLOCK
HT MOVE -2.5
REPEAT 5 [STROKE 3 MOVE 1]
ST MOVE -2.5
END
```

```
TO STROKE :HEIGHT
FD :HEIGHT BK :HEIGHT * 2
FD :HEIGHT
END
```

```
TO MOVE :DISTANCE
PU RT 90 FD :DISTANCE
LT 90 PD
END
```

The DRAW.BLOCK procedure draws a solid block when it is run. Because we are using the block in plotting, it is important that the DRAW.BLOCK procedure leaves the turtle in the exact center of the block. This makes the procedure a little more complicated than it would be otherwise.

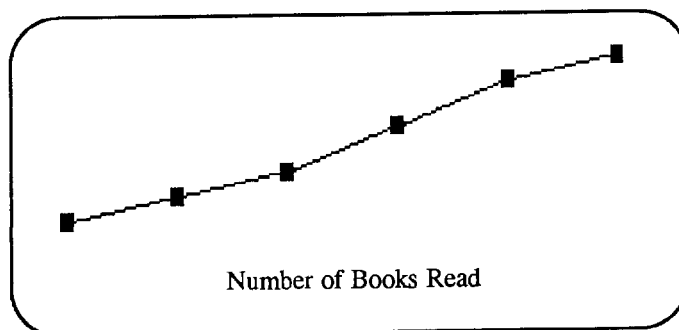
#### ?DRAW.BLOCK ■

The DRAW.BLOCK procedure is inserted into the PLOT tool.

```
TO PLOT2 :NUMBERS
IF EMPTY? :NUMBERS [ STOP ]
SETPOS LIST (XCOR + 40) (FIRST :NUMBERS)
PD
DRAW.BLOCK
PLOT2 BUTFIRST :NUMBERS
END
```

Here is what the cumulative total of books read looks like when the symbols are inserted at each point plotted.

```
?SETUP
?PLOT2 [8 17 26 43 60 69]
```



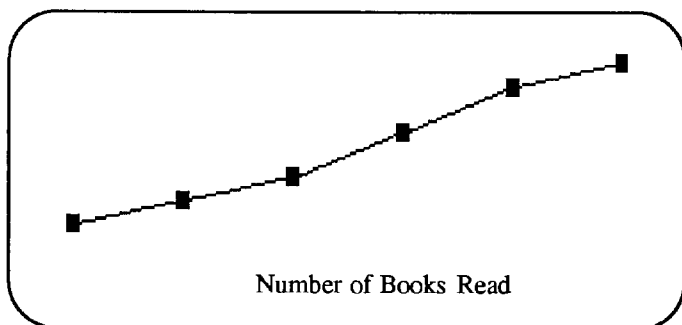
Storing Data in a Procedure

If you want, you can create a procedure which outputs the number of books read. That makes it possible to just add a new total to the end of the list each week.

```
TO BOOKS.READ
OUTPUT [8 17 26 43 60 69]
END
```

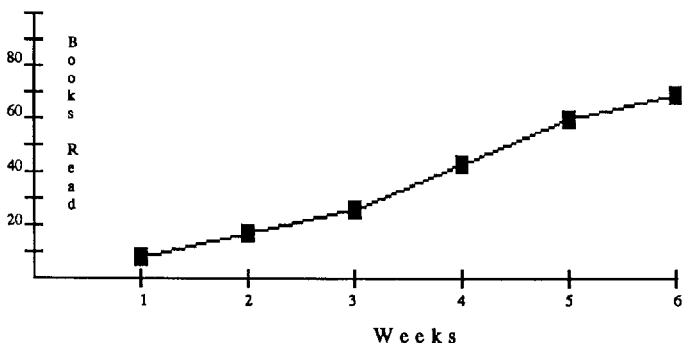
Here is how PLOT would be used with the BOOKS.READ procedure. The graph produced is identical to the one created when the data is typed in directly.

```
?SETUP
?PLOT2 BOOKS.READ
```



### LOGO CHALLENGE

An x-axis and a y-axis will make the graph easier to read. Can you (or your class) create procedures to draw axes such as the ones shown in the next figure?



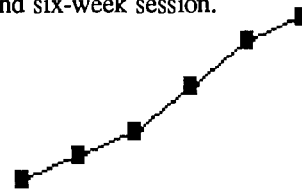
### Changing the Horizontal Scale

The number added to the x-coordinate in between each point determines the horizontal scale of the graph. For example, to shrink the width of the graph, the number added to the x-coordinate could be decreased from 40 to 20.

```
TO PLOT3 :NUMBERS
IF EMPTY? :NUMBERS [ STOP ]
SETPOS LIST (XCOR + 20) (FIRST :NUMBERS)
PD
DRAW.BLOCK
PLOT3 BUTFIRST :NUMBERS
END
```

Shrinking the width of the graph would allow text to be pasted on the other half of the page, or would allow more data points to be added in the second six-week session.

```
?SETUP
?PLOT3 BOOKS.READ
```



Suddenly the steeper slope of the graph makes it appear that the reading rate of the class has increased dramatically. In a social studies class, plotting tools provide an opportunity to discuss the effects of changing the way data is presented.

### Plotting a Computer-Generated Function

In the examples listed thus far, the numbers plotted have been typed in. Sometimes it is possible to save some work by allowing the computer to type the numbers for you. For example, the mathematical function  $y = x^2$  generates a series of numbers that look like this:

| x    | y    |
|------|------|
| -10  | 100  |
| -9   | 81   |
| -8   | 64   |
| etc. | etc. |

We could plot the series of values generated by this function by calculating them and typing them in manually:

```
?PLOT [100 81 64 etc.]
```

However, it is more efficient to allow the computer to generate the numbers for us.

```
TO GENERATE.FUNCTION
MAKE "DATA [ ]
MAKE "RANGE 10
FUNCTION.LOOP -1 * :RANGE
END
```

```
TO FUNCTION.LOOP :NUMBER
IF :NUMBER > :RANGE [ STOP ]
MAKE "DATA LPUT PARABOLA :NUMBER
:DATA
PRINT LAST :DATA
FUNCTION.LOOP :NUMBER + 1
END
```

```
TO PARABOLA :NUMBER
OUTPUT :NUMBER * :NUMBER
END
```

When GENERATE.FUNCTION is typed, a series of numbers is produced:

```
?GENERATE.FUNCTION
```

```
100
```

```
81
```

```
64
```

```
etc.
```

The GENERATE.FUNCTION procedure produces a variable containing a list of all the numbers which we might otherwise have typed in by hand:

```
?PRINT :DATA
```

```
100 81 64 49 36 25 16 9 4 1 0 1 4 9 16 25 36 49 64
```

```
81 100
```

This function generates a parabola. However, it is necessary to plot the numbers in order to see their relationship to a parabola. In order to fit all the data points of the parabola on the screen, the PLOT procedure must be modified so that the distance between each x-coordinate is only 10 (instead of 20 or 40). In addition, PLOT will run faster if the DRAW.BLOCK subprocedure is removed from this variation of PLOT.

```
TO PLOT4 :NUMBERS
```

```
IF EMPTY? :NUMBERS [ STOP ]
```

```
SETPOS LIST (XCOR + 10) (FIRST :NUMBERS)
```

```
PD
```

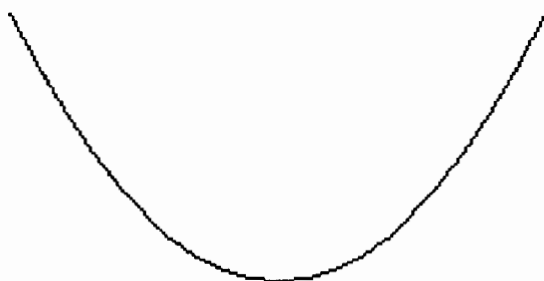
```
PLOT4 BUTFIRST :NUMBERS
```

```
END
```

After these two minor modifications to PLOT, the parabola function can be plotted:

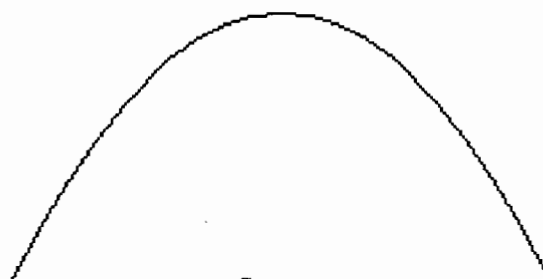
```
?SETUP
```

```
?PLOT4 :DATA
```



### LOGO CHALLENGE

Can you modify the function so that it generates a parabola which is standing up, like the figure at the top of the next column? Hint: The change required to generate the right series of numbers must be made in the procedure PARABOLA.



### Summary

Logo plotting tools can enhance other Logo procedures and applications. This is particularly the case when large quantities of data have been generated by the computer. In the illustration provided above, the function for a parabola is used, but any mathematical function could be substituted in the GENERATE.FUNCTION procedure. To generate another type of curve, simply substitute a procedure containing a different mathematical function for the PARABOLA procedure.

---

*Glen Bull is a professor in the University of Virginia's Curry School of Education, and teaches Logo courses at both the graduate and undergraduate level. His CompuServe number is 72477,1637. Paula Cochran is an assistant professor in the Communication Disorders Program of the University of Virginia's Curry School of Education. She is interested in Logo applications for language arts and special populations.*

---

## Coral Software Introduces Object Logo.

Coral Software Corporation has released Object Logo for the Macintosh, an advanced version of the Logo programming language which supports object-oriented programming. The language's advanced features have been designed as simple extensions of familiar Logo forms, so that experienced Logo users need learn only a few additional commands.

Its syntax and structure is compatible with most popular Logo versions, making it possible for programs written in other versions to run in the Macintosh environment.

Object Logo enhances traditional Logo graphics capabilities with multiple graphics windows, multiple turtles, and access to Quickdraw commands. Object Logo also incorporates advanced mathematics, such as unbounded integers, complex and rational numbers, and a full complement of transcendental functions.

For advanced users, a number of programming tools, such as an incremental compiler that automatically generates native 68000 code, an interactive editor and debugger, and full access to the Macintosh Toolbox, make Object Logo an unprecedented prototyping tool for Macintosh software development.

Object Logo retails for \$79.95 and is available directly from Coral Software. For more information, contact: Coral Software, PO Box 307, Cambridge, MA 02142, or call (617) 547-2662.

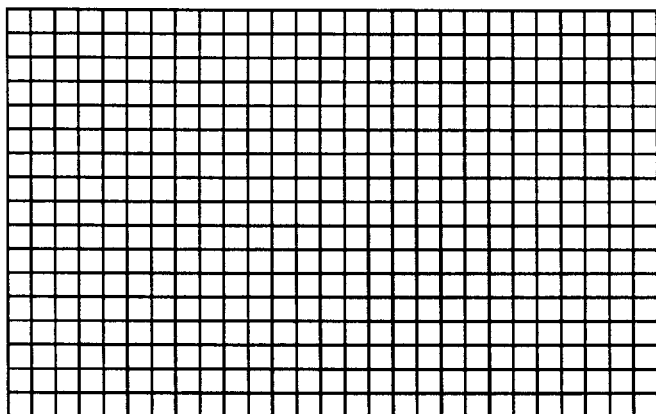
## TO BEGIN :COORDINATES

by Elaine Blitman  
and Barbara Jamile

Lush jungles, vast forests, and sparkling seas have long lured adventurers to explore the depths for unknown secrets. Such can be the attraction of Logo's coordinate grid which lies hidden in the graphics screen. Just what is it? What is its purpose? This month's column logs some of our adventures with the coordinate grid. We hope it will attract you to explore the territory for yourself and your students.

### Revealing the Grid

The grid is invisible and will remain a mystery until ways are devised to reveal it. Teacher help is usually needed to make the grid visible to students, and we've used several ways to do this. The following program draws a grid on the computer screen so that the children know what it looks like.



```
to x.lines
  pu setpos [130 90]
  repeat 9 [seth 270 pd fd 267 pu seth 180 fd 10 seth 90
pd fd 267 pu seth 180 fd 10]
end
```

```
to y.lines
  pu setpos [130 90]
  repeat 14 [seth 180 pd fd 170 pu seth 270 fd 10 seth 0
pd fd 170 pu seth 270 fd 10]
end
```

```
to grid
  x.lines
  y.lines
end
```

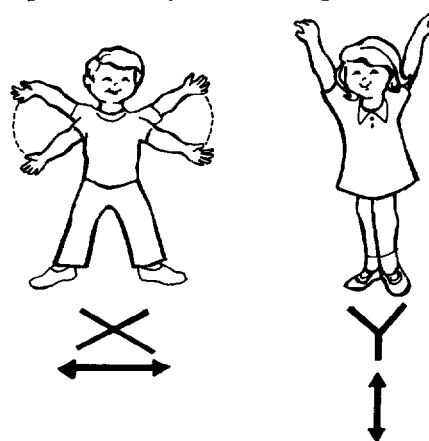
y

a  
x  
i  
s

x axis

A transparent grid taped to the monitor screen can make the grid visible and allow children to move the turtle to specific points as they experiment with the SETPOS (LCSI and IBM) or SETXY (Terrapin / Commodore) command. The same transparency can be used with an overhead projector to project a very large grid onto the chalk board or onto a piece of chart paper for class use. It's also useful for each student to have notebook-size paper grids for experiments and reference. A poster with information about the sign (+,-) of numbers in the various quadrants is helpful.

An "exercise" that we sometimes do while learning about x and y-coordinates is to remind the class, (with vigorous arm movements) that the x-coordinates determine the horizontal position and y-coordinates plot the vertical position.



After various experiences with grids, a less complex reminder can be used. The x-axis is printed on an adhesive strip placed across the top and/or bottom of the monitor frame, and the y-axis placed on either or both sides. Eventually students may not need this type of aid, but it certainly is helpful at first, and allows children to solve problems with less teacher assistance.

A large floor grid has many uses for learning about coordinates. Physical movement on a large grid of this type is fun, and reinforces those children who learn best kinesthetically. The chalk board grid or floor grid can be used for these group activities. A few children pick a pair of numbers and locate or stand on the point, while others locate the corresponding point on their individual grids or on the computer. Floor grids work well with hopscotch type games or with scooter boards that children ride to simulate turtle moves. If you have a turtle robot, this is also a fine opportunity to put it to use!

If you want to begin with one axis at a time to avoid confusion, use SETX and SETY before combining the commands with SETXY or SETPOS. PRINT POS, SHOW POS, and TOWARDS are other commands you will want to know about when working with locations on a grid.

### Projects and Activities

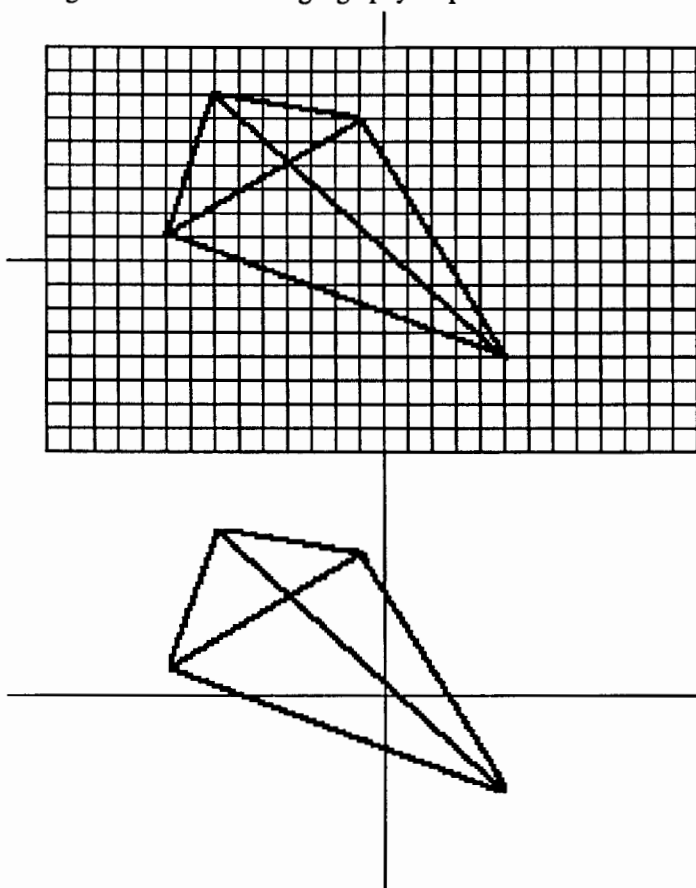
Once children are familiar with the appearance of the grid, the vocabulary, and its special features, it is time to develop an understanding of how and why grids are used.



Many games help develop the concept of location on a grid, some old favorites can be adapted to Logo, and others are created by teachers with this purpose in mind. "Battleship" is a game familiar to most students, and can be played on paper grids with a partner. Instead of locating and torpedoing ships, the children might hide a treasure or try to find a lost turtle.

Another variation on this theme is the placement of a transparent grid over maps of the classroom, the city, the world, etc. Children can give the coordinates for certain locations: the coordinates for Rome, Italy, or the name of the place closest to coordinate pairs drawn from a stack of cards. Another possibility is to give a designated route to follow, recording all the cities, countries, rivers, etc. which are traversed. The Logo grid identifies specific points on the screen with great precision.

After some experience with locating points, it's fun to play a hidden dot-to-dot picture-finding game on a large grid. One child plans a picture ahead of time, limited to five or six points. He or she calls out the points to plot. Others plot the points and draw the connecting lines. As the picture emerges, other members of the class may guess what it represents. Simultaneously, other teams of children enter the same information on the computer. Commercially produced coordinate grid worksheets, on which children make dot-to-dot drawings from a list of coordinate pairs, allows them to do the same type of activity. We've found that Logo experiences in the early elementary grades provide good foundations for learning longitude and latitude in geography map studies.



Adhesive-backed shelf paper comes in a grid pattern and can be used to cover tag board for a variety of board games. A second grade teacher at our school places stickers with specific themes (for example space, seasons, or animals) at various coordinates, creating games using coordinate points. Children also love to make their own board games and often learn a good deal more when they're teaching the game to other students. Graph-making is a natural extension from Logo coordinates to social studies, science, or other areas of the curriculum. LogoWriter has two activity cards and a project book devoted to graphs and grid projects.

### Thinking Strategies

You may want younger children to begin working only with the upper right quadrant (positive numbers) and gradually expand to other quadrants. One technique for familiarizing children with negative numbers is to let them play grid games that include negative numbers on the axis. Children will begin to see the relationship between positive and negative numbers through active play.

Some of the thinking strategies being discussed in recent books, articles, and conferences are made-to-order for working with Logo. One example given by Ron Brandt, editor of *Educational Leadership*, is an adaptation of Arthur Whimbey and Jack Lochhead's "Pair Problem Solving." The following might take place while a pair of children are working together at the computer with Logo:

The problem solver tells what he or she is thinking at every step, while attempting to solve a problem. The partner checks continually for accuracy, points out errors (but doesn't correct them), reminds the other person to speak aloud about his/her thinking, may ask the partner to wait a moment while he/she "catches on," encourages the problem solver to persist, and seeks clarification.

The strategy of peer tutoring also works especially well with Logo. Because Logo experiences take children in many different directions, it is helpful for them to share their experiences and special techniques with one another. Older children paired with younger children, and children of the same age, all seem to benefit from this reciprocal teaching. We see improved self-esteem as another advantage of this model as children share their expertise.

### Grid Invitation

We hope these ideas about the "secrets" of the hidden grid will encourage many explorations into unknown territory. Grid adventures can be both fun and educational for everyone.

We'd be interested in hearing about your grid activities. Please write to us in care of Punahou School, 1601 Punahou Street, Honolulu, HI 96822.

---

*Elaine Blitman and Barbara Jamile are the K-2 and 3-4 supervisors at the Punahou School in Honolulu, HI. They have been using Logo with young children since 1982. Their CompuServe number is 76067,211.*

# Logo LinX

by Judi Harris

## Time for a Challenge

*Rich problems* make the best Logo projects. Think, for example, of "Logo Poetry," tessellations, "Pig Latin," and fractals. All are, at first glance, good Logo challenges. But as your students (and you!) write Logo procedures that translate their patterns into computer activities, ideas for related projects spring to mind. And as you work on those new challenges, "idea reproduction" continues. Investigation into these broad areas of interdisciplinary, intrinsically fascinating inquiry reveal what Papert called *powerful ideas*. The subject's inherent patterns provide the pliable mess of material; Logo's structure encourages "messaging around" in search of patterns and structure.

### The Catalyst

Patricia Davidson first introduced me to the notion of "rich problems" at a National Council of Teachers of Mathematics conference. Ms. Davidson is best known for her resource books of mathematical learning activities that incorporate the use of manipulative materials. She also has done extensive research work on computational learning styles. At the conference, she spoke of *mathematically* rich learning situations with no intentional reference to Logo applications. I would hasten to add that there are rich problems that can be investigated in all subject areas, and that those with definable patterns make for the most engrossing inquiry when using Logo as an investigatory tool.

Davidson suggested one particularly rich store of problems that dealt with digital time patterns. Few of us will forget the pride and excitement we felt when we received our first wristwatch. (I'd even wager that yours is still in a box somewhere, preserved for posterity.) Now that digital watches are often less expensive to buy than analog timepieces (watches with hands), many children use them daily.

Some of the children that I taught had digital watches that could display so-called "military time," or time based upon a 24-hour clock. Alternate methods of telling time fascinated them, and the educational implications of the options intrigued me. When Ms. Davidson suggested digital time as a good source of rich problems, I "oooo-ed" along with the rest of the audience. The idea had all of the ingredients necessary for a good project: interest, innovation, relevance, multiple levels, powerful ideas, unanswered questions, and many "right answers."

### The "Mess"

Intuitively, we recognize when we stumble upon a rich problem. Groups of related sub-challenges quickly suggest themselves, crossing disciplinary and methodological boundaries, and making apparent new connections. I wondered, for example:

1. What are the frequencies of digit occurrence in each hour of digital time (how many 0's, 1's, 2's, etc. appear)?
2. How many times in a day are palindromic (able to be read forwards or backwards, such as 12:21)?
3. If we assume that the colon (:) in-between the hours and minutes is an equals sign (=), and that there are mathematical operators (+, -, X, /) in-between the hours and minutes digits, how many times could be computationally correct in a day (i.e., 7:43 could mean  $7=4+3$ , 8:24 could mean  $8=2*4$ )?
4. Does the choice of a digital or analog watch change the wearer's conception of time? If so, how?
5. How did the advent of standardized time change history?
6. What might happen if we asked someone to live for a period of time without a clock for reference?
7. What common expressions and famous sayings mention time and time-keeping? Are there general themes that run through these quotations?
8. How do quartz crystal, liquid crystal, and spring mechanisms work?

Perhaps these ideas have inspired some additional thoughts to add to your list. If so, or if you "ooo-ed" earlier in the article, then this is a rich problem for you, too.

### Messaging Around

I find it quite interesting that I originally began by suggesting investigation of some of these sub-problems by more traditional means: paper, pencil, and calculator. The students pointed out that, once they recognized the patterns of a particular piece of "the mess," there really was no point in completing the computations repeatedly for all examples of the pattern. I agreed, but added that the computations must all be executed, in any case, to be able to fully generalize the pattern.

Has your "Logo light bulb" brightened up? "Why not direct the computer to do the tedious computations, then let us find the patterns?" as my students impatiently suggested?

Sub-problems, of course, turned out to be conglomerations of challenges. It seemed logical first to write a set of procedures that simulate the functioning of a digital clock. Here is one solution:

```
TO DIGITAL.CLOCK :HOURS :MINUTES
IF :HOURS > 12 [ MAKE "HOURS 1 ]
IF :MINUTES > 59 [ STOP ]
PRINT.TIME
DIGITAL.CLOCK :HOURS :MINUTES + 1
MAKE "MINUTES 0
DIGITAL.CLOCK :HOURS + 1 :MINUTES
END
```

```
TO PRINT.TIME
CLEARTEXT
SETCURSOR [12 19]
TYPE (WORD :HOURS ": :MINUTES )
END
```

(Note: To use these procedures with Terrapin, Krell, or Commodore Logo, eliminate all brackets, change SET-CURSOR to CURSOR, and TYPE to PRINT1.)

Notice that there are two recursive calls in the DIGITAL.CLOCK procedure. This is logical, since for every one time that the value of :HOUR is increased by one (in the second recursive command,) the value of :MINUTES has to have been increased by one (in the first recursive call) 60 times.

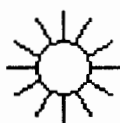
### Messy Temptations

Now it's your turn. How might you amend and append this fundamental idea to get the computer to:

- count digit frequencies as it displays the times?
- graph these frequencies? (See: Glen Bull and Paula Cochran's "Teaching Tools," Sept. '86 LX, pages 10-13.)
- display, count and graph the frequencies of all times displayed between two particular starting and ending times?

If you'd like to see a listing of procedures I've developed to solve this particular set of sub-problems, please send a long, self-addressed, stamped envelope to:

Judi Harris  
292 Ruffner Hall  
Department of Educational Studies  
University of Virginia  
405 Emmet Street  
Charlottesville, VA 22903



But, please! First give *yourself* time for the challenge.

*Judi Harris was an elementary school computer use facilitator, graduate education instructor, and computer consultant for a number of public and private schools in Pennsylvania. She is now a doctoral student in education at the University of Virginia. Her CompuServe electronic mail address is 75116,1207.*

### Logo Conference Information

March 14, 1987. *Beyond Computer Literacy - Preparing for the Next Generation*, sponsored by the Network for Action in Micro Computer Education (NAME), Computer Learners Users Educators, the NJ State Department of Education, New Jersey School Boards Association, and the Ridgewood NJ Public schools. The keynote speakers for the conference are Dan and Molly Watt. The conference is being held at Ridgewood High School, Ridgewood, NJ. For more information, contact: Gary S. Stager, NAME, Fallon Education Center, 51 Clifford Drive, Wayne, NJ 07470, or call (201) 694-7800.

## MathWorlds

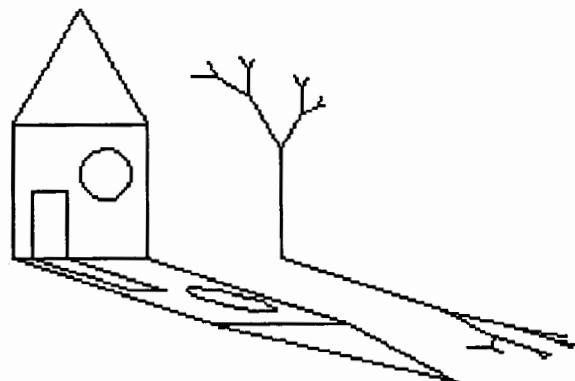
edited by  
A. J. (Sandy) Dawson

This month I have the pleasure of welcoming Jim King as a guest columnist. Jim is a professor of mathematics at Washington University in Seattle. He teaches, among other things, a senior level geometry course required by all students studying for a mathematics degree in secondary education. In the past few years, Jim has incorporated Logo into his course. One of the sections of the course deals with three-dimensional Turtle geometry. The column which follows is a condensed version of a much longer piece Jim wrote about this unique Turtle which can draw shadows. Now, for an introduction to a new, shadowy microworld, read on and explore the realm of the Affine Turtle.

### The Affine Turtle: A Turtle That Can Draw Shadows

by James King

When the sun shines on squares and circles standing upright on a flat surface, they cast shadows which are parallelograms and ellipses. Other 2-dimensional figures, such as trees, cast shadows with lengths and angles distorted. This same sort of distortion occurs when we represent a figure drawn on a sheet of paper as it is seen from an oblique angle (from a distance). Both are examples of parallel projection.



One way to draw such shadow figures in Logo is to use a 3-dimensional turtle to mimic the 3-dimensional sunlit scene. However, it is easier to draw them using the two-dimensional turtle described below. It should not be surprising that this is possible; after all, architects create such renderings on two-dimensional drafting tables. The mathematics of such figures is in fact a part of plane geometry, a part called Affine Geometry.

#### Using the Affine Turtle Commands

First we will describe how to create figures with the Affine Turtle. The basic commands for the Affine Turtle correspond to the usual turtle commands but begin with an A; namely, AFD, ABK, ART, and ALT are used in place of FD,

BK, RT, and LT respectively. There is also an ASETPOS which moves the Affine Turtle to a fixed (x,y) position on the screen and ACS, which clears the screen and moves the Affine Turtle to the center of the screen.

A new command needed by the Affine Turtle is SETHL :H :L; it is used a lot like the usual SETHEADING command. What SETHL does from the point of view of the observer is to "slant" the turtle's plane; when the turtle carries out a procedure the resulting figure appears slanted just as shadows are.

More precisely, the Affine Turtle carries around two vectors, HEAD and LAT, indicating the forward direction of length one and the left direction of length one from the turtle's point of view. The reason LAT points left is because that is the traditional convention in mathematics and physics for the standard orientation of the plane. With a minor adjustment in the ALT procedure, LAT could be pointed to the right. [For full details on these vectors and the mathematics behind the Affine Turtle, readers are directed to Abelson and DiSessa's *Turtle Geometry* (pp. 136-159), from which many of the ideas for this article were derived. Alternately, readers could write to Jim King for the complete manuscript from which this encapsulated column is derived.]

The conventional Logo turtle would have a HEAD vector pointing forward one screen unit and a LAT vector obtained by rotating HEAD left by 90 degrees. Thus

```
?SETHL [0 1] [-1 0]
```

has the same effect as the usual SETHEADING 0. After the above command has been run, a procedure such as

```
TO ASQ
REPEAT 4 [AFD 50 ART 90]
END
```



draws a square in the normal manner (except slower, alas).

On the other hand, if we choose the HEAD and LAT vectors to be vectors which are not perpendicular or are not both of the same length, ASQ draws a parallelogram. What we have done is redefine the rules for how angles and lengths are measured in the turtle plane. For example, if an Affine Turtle is positioned at [0 0], then

```
?SETHL [1 0] [2 1]
?ASQ
```

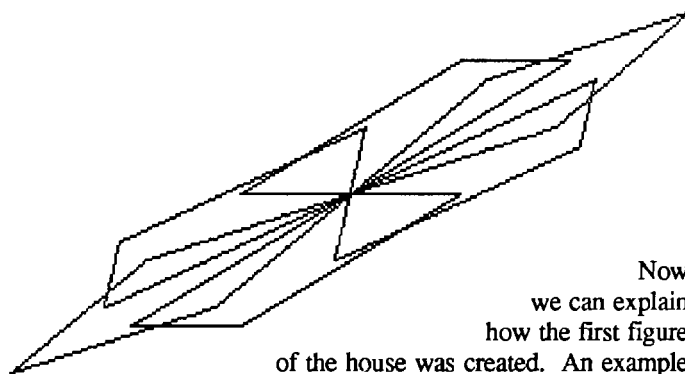


draws a parallelogram. Such a turtle given the command

```
?REPEAT 6 [ASQ ALT 60]
```

draws a standard "spin-square" figure which appears slanted, or viewed from the side.

An important point to note is that the SETHL command must be given each time you wish to slant the plane. The HEAD and LAT vectors together form what is called the moving frame (or just frame) of the Affine Turtle.

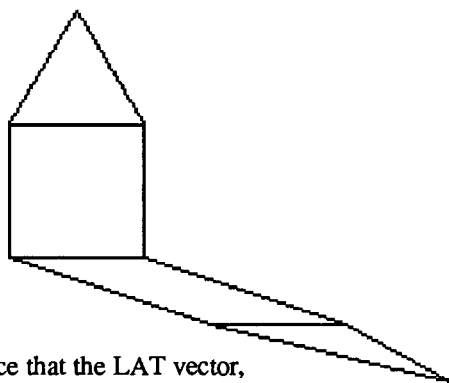


Now we can explain how the first figure of the house was created. An example of a house-drawing procedure (with the door and window omitted for brevity) is this one:

```
TO HOUSE
ASQ
AFD 50 ART 30
REPEAT 3 [AFD 50 ART 120]
ALT 30 ABK 50
END
```

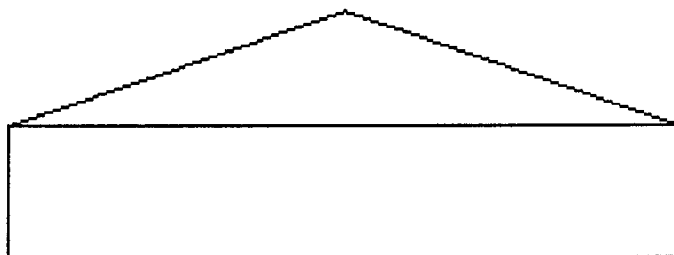
It is important that at the end the turtle returns to its original position and heading, at the lower-left corner pointing upward. To draw a house with a shadow, type

```
?SETHL [0 1] [-1 0] HOUSE
?SETHL [1.5 -0.5] [-1 0] HOUSE
```



Notice that the LAT vector, which is lined up with the base of the house, is left unchanged, but the HEAD vector is lined up first with the vertical side of the house and then with the slanted side of the shadow.

New Affine Turtle commands can be defined using SETHL. For example, SCALEL :SCALE multiplies the LAT vector by the real number :SCALE. Thus SETHL [0 1] [-1 0] SCALEL 5 HOUSE is equivalent to SETHL [0 1] [-5 0] HOUSE and gives this result:



One notable aspect of the shadow drawing is that the ART command actually turns the turtle to the left because the LAT vector is on the right side. This suggests defining a command FLIP which reverses the LAT vector (define it as SCALEL -1); this can be used for exploring bilateral symmetry and reflection in a line.

A procedure like SCALEL has the advantage of being an intrinsic turtle geometry command, unlike the global command SETHL, because SCALEL stretches in a direction that depends on the state of the turtle. For example,

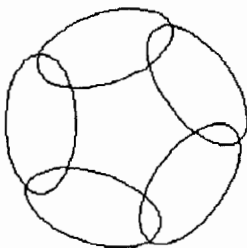
### ?SCALEL 3 ASQ

produces a rectangle because the stretch is aligned with the side of the figure; then SCALEL 1/3 will return the turtle to its original state. Thus

?SETHL [0 1] [-1 0]

?REPEAT 6 [SCALEL 3 ASQ SCALEL 1/3 ALT 60]

produces 6 congruent rectangles spun around the center of the screen while the previous spin-square produced parallelograms. This idea is the basis of the ELLIPSEGON procedure which was used in this figure:



Though brief, I hope this introduction to the Affine Turtle has whetted your appetite sufficiently so that you will explore the fascinating world of parallel projection on your own. Indeed, if you present this Affine Turtle microworld to your students, new vistas will open to them for the expression of their imaginations in the realm of the third dimension. Perspective drawing now becomes possible as the Affine Turtle glides across the screen. For a listing of the procedures needed to create this microworld, send a stamped, self-addressed envelope to:

Affine Turtle Procedures  
Logo Exchange  
PO Box 5341  
Charlottesville, VA 22905

For more information on the Affine Turtle microworld, write to:

Jim King  
Department of Mathematics  
University of Washington  
Seattle, Washington 98195

A. J. (Sandy) Dawson is a member of the Faculty of Education at Simon Fraser University in Vancouver, British Columbia. His CompuServe number is 76475,1315.

## SpecialTalk

by Paula Cochran and Glen Bull

### Logo: The Binding for a Book with 50,000 Pictures

The classroom and the therapy room are the scene of many simulations. It is usually not possible to take a class to an actual site, so the teacher attempts to simulate reality within the classroom. A field trip to the local Coca-Cola bottling plant can be arranged, but a picture of the Andes has to suffice in place of the real thing.

Simulations are also used in language testing. A picture of a puppy running serves as the stimulus item for:

"The puppy is running"

since it is usually not possible to arrange for a real puppy in the therapy room.

Simulations vary in their degree of reality. The most concrete representation is an actual object, such as a toy truck representing a real truck. A picture of a truck would be less concrete, and a sketch or line drawing even less so.

Where do computer simulations fall on this continuum? Unfortunately, computers do not provide all the power that we might like for simulations with special populations. In comparison with a photograph, computer graphics have limited resolution. Computer graphics also require a lot of room to store. It is only possible to save about a dozen Logo pictures on a disk using the SAVEPIC command.

Pictures are widely used by clinicians and special educators. It is not uncommon to hear a teacher ask, "Where can I find a picture of \_\_\_\_\_ for Johnny?" Or, "I need ten pictures of things which begin with 'r' for my fourth period class tomorrow." The limited resolution of computer graphics and limited storage capacity of computer disks make it unlikely that they will replace pictures in books for lessons.

However, computer graphics can do something that no picture in a book can do. They can move. Consider the difficulty of trying to represent these two concepts with a picture:

"The boy is about to walk."

"The boy is running."

Recent research in language intervention has compared the use of pictures with use of animated graphics to teach action verbs such as "run," "jump," and "dance." These are verbs which are difficult to capture in a still picture, and awkward to demonstrate repeatedly. Preliminary research indicates that computer graphics can be used effectively to teach these concepts.

Useful as computer animation can be, animation on a microcomputer is relatively crude in comparison with filmed animation. Compare the animation in a Walt Disney film such as "Snow White" with the animation in a computer game, for example.



In addition, animation is difficult to achieve with Logo. A few versions of Logo have made animation possible through Logo sprites. Unfortunately, these versions of Logo often required special hardware, and never became extremely popular. LogoWriter may be an exception.

### The Videodisk

A new storage medium has made it possible to access real pictures -- lots of them -- with Logo. The new technology is the optical disk. A videodisk is about the same size as a phonograph record, but is read with a laser. The videodisk player is attached to a television or video monitor to display the picture.

One videodisk can hold about 50,000 pictures on each side. This is a large number. Try imagining it this way:

- \* Imagine one teacher holding 50 photographs in her hand.
- \* Now imagine her holding a 500-page book with a photograph on each page.
- \* Now imagine 100 teachers, each with a 500-page photograph book, a total of 50,000 pictures.
- \* NOW -- imagine yourself holding the equivalent of that many pictures on a single disk about the size of a long playing phonograph record. And that's just on one side!

This is a far greater number of pictures than could be held on any floppy diskette -- and these are real photographs, rather than relatively crude (by comparison) computer graphics.

The implications for education are great. If you only looked at each picture for 10 seconds, it would take a week to look at 50,000 pictures. In articulation therapy, it would be possible to have as many pictures of "r" words as anyone could want.

With a videodisk, it is possible to have segments of videotape as well as still pictures. The way language testing is conducted could change. No longer is it necessary to have only still pictures in language testing. A videotaped sequence could be used instead of a still photograph to illustrate the difference between the "The boy is running" and "The boy is skipping," or between "running" and "ran."

### Logo Enters the Picture

Some videodisk players are easily controlled by any computer with a serial port. We have used a Pioneer LDV-6000 player and a Philips VP 935 videodisk player thus far. These players can be controlled by any computer language, but we have used Logo because that makes it possible to extend other things which we are doing with Logo.

Of course, the videodisk players can be operated with hand controls in the same way that a videocassette recorder (VCR) is operated. There are PLAY and PAUSE buttons on the remote control which can be used to advance or stop the picture. However, with 50,000 pictures it may be difficult to

remember that the picture of a robin is number 37,342. Logo procedures make it possible to type commands such as:

**?FIND ROBIN**

(a picture of a robin appears)

**?FIND RABBIT**

(a picture of a rabbit appears)

It takes about as long for the next picture to appear as it would for a slide in a regular 35-millimeter slide projector to change. The main difference is that, with a videodisk, more "slides" are available -- thousands more. And you can ask for each slide by name, without having to go through them in any particular order.

We have created other Logo procedures which access the power of the videodisk player. For example, PLAY and HALT will stop and start video sequences. Imagine a lesson in which the teacher or clinician shows the child a drawing of a fish. The teacher says, "The fish is swimming." Now imagine how much more concrete the same lesson could become. The teacher and child look at the computer screen. The teacher types SWIM, a Logo procedure which FINDs the beginning of the fish sequence and starts it PLAYing. A fish appears and swims around near the floor of the ocean. The teacher says, "The fish is swimming."

Synthesized speech can be included as well (see January's SpecialTalk column about making Logo talk). We have created a Logo program we call "Alpha-Birds." Pressing any letter on the keyboard causes a picture of a bird whose name begins with that letter to appear. Then the speech synthesizer says the name of the bird. For example, when the letter "P" is pressed, a picture of a parrot appears, and the speech synthesizer says, "Parrot."

In another program, Logo poetry is illustrated by pictures of the objects which are mentioned in the poem spoken by the synthesizer. These applications only hint at the possibilities.

### A Demonstration

The printed word does not always do an adequate job of conveying the power of Logo. We would like to include a videodisk with each copy of the *Logo Exchange*, but unfortunately that is not possible. However, if you live the east coast of the USA, and are planning to attend the East Coast Logo Conference, April 2 - 4, 1987, we will be demonstrating the combination of Logo with a videodisk player at a presentation there.

---

*Paula Cochran is an assistant professor in the Communication Disorders Program of the University of Virginia's Curry School of Education. She is interested in Logo applications for language arts and special populations. Glen Bull is a professor in the University of Virginia's Curry School of Education, and teaches Logo courses at both the graduate and undergraduate level. His CompuServe number is 72477,1637.*

# Testudinal Testimony

by Douglas H. Clements  
Mathematics and Logo

## Does Logo Affect General Mathematics Achievement?

We saw in the first column that early studies reported mixed results, especially on mathematics achievement tests. Therefore, the following will come as no surprise! Forewarned, let's take a look at several "snapshots" of recent studies on the effect of Logo on mathematics achievement.

Blumenthal (1986) worked with 11-year-old children identified as having low mathematics achievement and low academic self-concepts. Her goal was to develop their persistence and desire to increase their effort in the face of failure. To accomplish this, she guided them through 12 days of Logo, 30 minutes each day. Although there were gains in the emotional domain (see the December 1986 column), there was no significant difference between the Logo and control group on mathematics achievement. The researcher questioned the validity of the test results, however, because her observations suggested that many of the children did not attempt to work to their ability level, skipping pages and neglecting to read test questions (the children knew the testing was not part of their regular school curricula).

Another group of fifth graders in a school district attempting to build a Logo program was randomly assigned to Logo or control groups (Studyvin and Moninger, 1986). This produced a close match in terms of IQ scores (109.88 for the control group and 109.33 for the Logo group). The Logo group was then provided 12 weeks (1 hour per week) of structured Logo instruction (using *Logo Discoveries* from Creative Publications). This instruction blended discovery and structure, with students working in groups of two or three. The control group worked individually with mathematics drill and practice, keyboarding, and word processing programs.

On the mathematics test from the Iowa Test of Basic Skills, the Logo group maintained their scores. The control group evidenced a decline slightly less than that representing the familiar pattern for this age in the district. Differences, however, were not significant.

After four weeks of working with Logo, a group of kindergartners made a greater gain than a control group on all the number subtests of the Brigance Readiness test (Reimer, 1985). However, for only one, "number recognition 10-100," was this gain significant (only individual subtests were subjected to tests of statistical significance).

Yet one more group of fourth and fifth graders experienced 9 weeks of Logo programming (Mohamed, 1985). The content was fast-paced and structured. They performed significantly better than a control group on the total Developing Cognitive Abilities Test. They also performed better on each of the subtests. However, the difference on the quantitative subtest did not reach significance, whereas the

difference on the spatial subtest did. This was supported by observational data indicating that the majority of students improved in their ability to estimate the sizes of line segments and angles and to recognize symmetry.

Considering these, along with previous studies on achievement, one might say, "Some promise, possibly, but not extremely impressive." What's happening? Many teachers claim that they can see students doing mathematics in Logo. Are they just fooling themselves? Possibly not. The research picture is a little brighter when focused on students working within the Logo environment.

## What Mathematics is Occurring Within Logo Environments?

Classroom observations have demonstrated that students do utilize certain mathematical notions in Logo programming. For example, the following represent some of Kull's (1985, 1986) observations of primary grade children.

- Many students used self-chosen "standards of measure"; for example, first-grader Ryan consistently constructed drawings with moves of 33. FD 33 for the first side of a rectangle, FD 33 FD 33 for the long side, and so on.

- They also use notions of inverse operation. Ryan wanted to turn the turtle to point into his rectangle. He asked the teacher, "What's half of 90?" After she responded, he typed RT 45. "Oh, I went the wrong way." He said nothing, eyes on the screen. "Try LEFT 90," he said at last. This inverse operation produced exactly the desired effect.

- Arithmetical operations were not always precise, but the use of logical-mathematical reasoning—for a purpose—was apparent. Lynn was making a square for a "frame." She had completed two sides and two turns. She typed FD 50. Seeing that the line segment was too long, she said, "I forgot it was 40." She changed the pen color to the background color and typed BK 9. The teacher asked why she chose that instruction. She replied, "Because I pushed 50 and I knew it should be 40, so it needed 9 less".

- Amy had used a series of "FD 9's" to reach a certain point and needed just a bit to finally close the gap. Checking the size of the gap with her finger to determine "how much of 9" was required, she typed FD 7. Amy was using estimation skills and was gaining a sense of proportion.

Thus, Kull observed these first graders utilizing such mathematical notions as number, place value, division, estimation (of arithmetic operations, distances, and angles), standard units of measure, accumulation of linear quantity, patterning, ordering, proportion, symmetry, inversion, compensation, and part-whole relationships. She maintains that this is a reflection of what Piaget called the "spontaneous mathematical intelligence of the young child."

Hillel (1984) wanted to see precisely what students might learn about mathematics from Logo programming on a session-by-session basis. Working with third graders for 12 weeks (1 hour per session per week), he reported the emergence of certain mathematical ideas.

By session 3, there was clear-cut evidence of use of only two ideas. The students consistently used the inverse concept of both rotation (i.e., like Ryan, they saw that LT 60 could be "undone" with LT 60) and displacement (FD 30 "undoes" BK 30). They also consistently used 90 degrees as inputs for rotation commands. Four out of five of the pairs showed consistent use of relative length by session 4, an indication of the acquisition of good estimation skill for length. The definition of a CIRCLE procedure was not well understood by most of the groups, although they did understand that larger inputs produce larger circles. Such ideas as the relationship among radius, diameter, and circumference were not mastered.

Additivity of length was not used consistently, but Hillel noted that while some contexts call for the application of this construct in a direct and obvious manner, others do not. After the children understood the role of 90 degrees, most students showed consistent differentiation between acute and obtuse angles. Hillel concluded that through the use of Logo, traditional obstacles to the acquisition of mathematical concepts do not disappear. However, he stated that he did not underestimate the achievement of the 8-9 year olds.

For many teachers involved in the Carmichael et al. (1985) study, turtle geometry dispelled the misconception that all mathematical activity must occur in a highly sequenced, tightly controlled order. Certain Logo explorations allow students to explore a concept at an initial, intuitive level that forms a foundation for later learning at a formal, deductive level. A junior-level student who understood something of this notion said that "with Logo I am doing things in math before we ever learn it" (p. 288).

This is Papert's notion of "concretizing" abstract mathematics. Logo's ability to make mathematics concrete and real to students was especially important for one of the learning disabled students. His teacher prescribed at least two addition and multiplication worksheets per day because he hadn't mastered the arithmetic operations. He did not like the exercises and still had to use "crib sheets" to finish them.

Immediately thereafter, he sat down and drew a circle with Logo. He set out to inscribe a "diamond" within it. According to a researcher observing him, he used trial and error, estimating how far the turtle would need to move, while simultaneously adding and subtracting two-digit numbers mentally "with great ease. I could not believe it. Here was a boy, who only ten minutes earlier needed a 'crib sheet' and here he was engaged in very quick and sophisticated arithmetic work" (p. 288). He finished the design, a process that "would have taxed the most competent adult" (p. 289).

Those familiar with John Holt's work might be reminded of his story of the boy who "could not do" school mathematics. Holt was shocked to find him in a bowling alley keeping score, doing the necessarily complex arithmetic flawlessly. The next time, he came to school, Holt gave him arithmetic in the form of bowling scores. This seemingly wise move yielded naught; the boy could not repeat his Saturday performance. Holt hypothesized that he was sure that he would fail at all school tasks, not matter what the form.

Another interpretation is that the boy had one set of strategies—schema—for school tasks and another for out-of-school tasks, and the two did not connect.

Logo may occur within the school context, but the warning is still sound: Teachers must try to build connections for children such as the learning disabled child, or the mathematics they learn and use in programming situations may never significantly affect their overall mathematical development.

While the Carmichael report borders on the anecdotal, it does vividly demonstrate that Logo has the potential to illuminate important mathematical concepts and engender a mathematical way of thinking, even, surprisingly, in classrooms in which the teacher has had little or no training.

Not surprising, however, is evidence in the report that the greatest effects were due to dedicated work by teachers who struggled to build a close correspondence between their students' computer and classroom experiences.

These teachers were committed to developing their students' understanding of Logo not just as a computer language but as a means of exploring specific mathematical ideas.

The authors conclude that Logo made it possible to explore certain mathematical concepts in a different sequence and much earlier than is currently believed. They state that their findings "confirm that Logo has indeed given many children a powerful medium in which to explore, to experiment, and to reflect upon their own thinking."

### Where Does All This Leave Us?

Let us reflect on the first group of studies. That effects are postulated in some studies raises the question: Should such effects have been expected? Should measurable and significant effects emerge after exposures which ranges from 12 weeks to 12 hours to 12 days (with a total of only 6 hours)? If Logo is seen as a "conceptual framework" for learning mathematics, such expectations may be unreasonable. Even if they were reasonable, there are additional design problems that may obscure any possible effects. For example, some studies utilize 10 or fewer students per group.

What does this mean? It means that even if the ideal "true difference" between the groups was a full standard deviation, one would have less than a 50% chance of "reaching" statistical significance. That is, even with a "strong effect" (Cohen, 1977) (of a magnitude that even direct teaching of rote knowledge often fails to achieve), you have less than a 50-50 chance of "turning it up." With a more moderate—but still practically significant—effect, your chances are even worse.

Then, to add insult to injury, too many researchers go on to make a common, but regrettable, error. They conclude that not finding evidence for treatment effects is tantamount to finding evidence that there is no such effect. Especially with a low number of subjects, they have no justification for that conclusion, which may be more likely wrong than right.

Nevertheless, this body of research makes it clear that significant, automatic increases in arithmetic achievement are

not to be expected. (Note that this applies to those using similar Logo environments and achievement tests.) As teachers, we face several difficult tasks. We must develop Logo tasks that illuminate specific mathematical ideas. We must learn better ways of helping students reflect on the mathematics in their Logo programming. Interaction emerges as a key factor from these studies. Interaction with adults must emphasize of mathematical thinking in all contexts, and the similarity of the mathematical structures of those contexts.

Although the observations of students doing mathematics while programming are promising, we cannot stop there. A critic might well argue:

"These things might be observed. But if they have no measurable effects on students mathematics ability in other situations, they're of little use and probably represent an inefficient use of time."

If we can't demonstrate their worth, we are on ground no firmer than those that proclaim that doing distasteful mathematics is good for children's character.

It is possible that some researchers are "looking in the wrong place." A common observation of the reports was that standardized tests probably did not tap what might be learned in Logo.

Will more suitable, more specific instruments yield more consistent results? That question will be addressed in forthcoming columns.

### References

- Blumenthal, W. (1986). *The effects of computer instruction on low achieving children's academic self-beliefs and performance*. Unpublished doctoral dissertation, Nova University.
- Carmichael, H. W., Burnett, J. D., Higginson, W. C., Moore, B. G., & Pollard, P. J. (1985). *Computers, children and classrooms: A multisite evaluation of the creative use of microcomputers by elementary school children*. Toronto, Ontario, Canada: Ministry of Education.
- Cohen, J. (1977). *Statistical power analysis for the behavioral sciences* (rev. ed.). New York: Academic Press.
- Hillel, J. (1984). *Mathematical and programming concepts acquired by children, aged 8-9, in a restricted Logo environment*. Unpublished manuscript, Concordia University, Montreal, Quebec.
- Kull, J. A. (1985, April). *Programming, problem-solving, and mathematical learnings in young children learning Logo: A collaborative, qualitative study in the first grade*. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.
- Kull, J. A. (1986). Learning and Logo. In P. F. Campbell & G. G. Fein (Eds.), *Young children and microcomputers* (pp. 103-128). Englewood Cliffs, NJ: Prentice-Hall.
- Mohamed, M. A. (1985). *The effects of learning Logo computer language upon the higher cognitive processes and the analytic / global cognitive styles of elementary school students*. Unpublished doctoral dissertation, University of Pittsburgh, Pittsburgh, PA.
- Reimer, G. (1985). The effects of a Logo computer programming experience on readiness for first grade, creativity, and self concept. "A pilot study in kindergarten." *AEDS Monitor*, 23, 8-12.
- Studyvin, D., & Moninger, M. (1986). *Logo as an enhancement to critical thinking*. Paper presented at the Logo 86 Conference, Cambridge, MA.

---

Douglas H. Clements is an associate professor in the College of Education at Kent State University in Kent, OH. He teaches Logo courses and conducts research on the use of Logo to develop cognitive, mathematical, and social-emotional competencies. His CompuServe number is 76136,2027.

---

## Logo Disserts: Dissertations Dealing with Logo by Barbara Elias

Among the studies which examine the relationship between learning to program in Logo and problem solving ability is one conducted by M. G. Fickel at the University of Nebraska. Fickel investigated the effects of learning to program in Logo on the development of cognitive ability, problem solving ability, and locus of control. Two additional variables, gender and computational ability, were also included for study.

Utilizing a representative design (no artificial conditions were created for the research), this study was conducted under normal school conditions, for one academic year, with 96 sixth grade students. These students, the entire sixth grade class, attended a departmentalized public middle school in a western Nebraska town. They had virtually no prior computer experience. Scores from a computational skills test, administered by the school, were used to assign each student to a mathematics class at one of three ability levels. Students in the single highest level class were assigned randomly to treatment or control individually. Since there were two classes each at the other two levels, these students were randomly assigned to treatment or control as a class - one each at each level.

Students studied Logo or CAI text-related software in the regular mathematics classroom in 20 minute sessions. Logo instruction was provided by a set of Logo cards from the Nebraska Department of Education. Students used the cards individually, with minimal teacher assistance, approximating a guided discovery approach. They saved work on a disk which was examined periodically for the number and complexity of procedures saved. CAI software was assigned to students in the control group on the basis of need. Over the school year, students in each group spent about 16 hours on the computer.

Instruments which provided data for this study included: The *Developing Cognitive Abilities Test (DCAT)*, published by Scott Foresman, the *Iowa Problem Solving Program*

(IPSP) test, the Nowicki - Strickland Locus of Control Scale for Children (NSLCC), and a Logo test developed from the Logo study cards. Pretest and posttest scores were analyzed and anecdotal data, a collection of typical classroom events recorded over the year, were listed.

The results of this study have implications for curriculum planners and those concerned with student achievement. There were no differences in the development of cognitive ability or locus of control resulting from the study of Logo. It is interesting to note that no gender differences were observed in problem solving, cognitive ability, or locus of control. Differences were observed, however, with respect to ability in problem solving and in cognitive development and Logo learning. Students in the highest group gained more than those in the other two groups.

The researcher suggests further study of ways computers are used in classrooms if computer use is expected to aid in developing the higher order thinking skills and if the computer is expected to assist in erasing the disparity between high and low achievers.

Are there benefits to be gained from computer throughout the elementary school years? Can mastery learning be applied to the use of the computer? Anecdotal records from this study indicate that Logo learning can be exciting and motivating for all students, both high achievers and low achievers.

>>>Fickel, Monty G. (1986). *The Effects of Logo Study of Problem Solving Cognitive Abilities of Sixth Grade Students* (Doctoral Dissertation), University of Nebraska, 1986.

Interested readers may write to:

Monty G. Fickel  
Department of Mathematics  
Chadron State College  
Chadron, NE 69337

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the LX Dissertation and Thesis Repository. Reports may be mailed to Barbara Elias, 4223 Hickory Road, Ettrick, VA 23803.)

---

*Barbara Elias is an assistant professor in the School of Education of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.*

---

## LXionary

### A Lectionary of Selected Logo Readings with Commentary and Opinion by Bill Craig

>>>"Huge Learning Jumps Show Potency of Brain Based Instruction" by Charmaine Della Neve, Elsie A. Hart, and Edgar C. Thomas. *Phi Delta Kappan*, October, 1986.

This article describes a pilot program of instruction at a New Jersey elementary school. The program piloted what the authors call brain based or brain compatible instruction. While the article details the design and impressive results of the pilot, what I found most interesting was the description of Proster Theory, the learning theory on which the pilot was based. Proster Theory defines learning as the "acquisition of useful programs," with programs being defined as a "chain of steps needed to accomplish some foreseen goal." Because the brain stores thousands of programs, knowing which program to use requires the recognition of patterns. So it is the role of the teacher to help students build programs and recognize patterns. The authors outline seven principles on which brain compatible lessons are based:

1. Create a non-threatening climate. "Anything which students might interpret as punitive ...will have an adverse effect on learning."
2. Provide huge amounts of input. Input can come from many sources: materials, equipment, audiovisuals, library research, presentations from staff, visitors, students.
3. Emphasize genuine communication. This may take the form of talking, listening, writing, or reading. Communication is always done for a purpose, not as an exercise.
4. Provide for much manipulation. This is particularly important in mathematics, science, and writing.
5. Emphasize reality. Problems and examples should come from the real world rather than from textbooks and worksheets.
6. Address learning activities to actual, productive uses. Students could prepare signs for an event as part of a language activity, for example.
7. Respect natural learning. "Everyone accepts that children's brains work in highly individual ways."

The authors cite two major differences in brain compatible instruction and traditional instruction, which they call brain antagonistic instruction. First, brain compatible instruction recognizes that learning is a slow, gradual, and individual process. Second, the brain does not go down one path. It potentially goes down a hundred paths simultaneously. Random learning is aided by the provision of many different experiences. Primary grade students in the pilot were involved in running businesses, holding elections, and publishing a newspaper.



Is instruction with Logo compatible with brain compatible instruction? Obviously, a Logo environment shares the climate of the brain compatible classroom. But what about some of the other principles? Do our uses of Logo emphasize genuine communication? In what ways does Logo provide our students with different kinds of input? Are students manipulating anything when they work with Logo? Do our Logo activities have productive, real life goals?

Your answers to those questions will depend on your ways of using Logo. The value of considering these questions is the realization that sound learning theories, Proster or other, have much to offer teachers of any discipline. Familiarity with these theories and the thinking on which they are based will only help us become better teachers. To quote a quote, "There is nothing more practical than a good theory."

I welcome your opinion about the articles featured in this column. Also, I would like to know of your recommendations for other articles to include. Write to:

Bill Craig  
Chesterfield Schools  
Professional Development Center  
2318 McRae Road  
Bon Air, VA 23235

---

*Bill Craig is the Computer Education Program Specialist for the Chesterfield County Schools, Chesterfield, VA.*

---

## LIFT's

# Teacher to Teacher

by Anne Cairns Federlein

*Logo: A Learner's Guide*, by George Burns and Michael Cook, Prentice-Hall Publishers, Englewood Cliffs, NJ 07632, 290 pages, \$30.75.

**Target Audience.** This book is for teachers who are beginning the quest of understanding the "game" of Logo.

**Content.** There are seventeen easy to understand chapters in the book. They include Drawing with Turtle Graphics, Making a Program, Changing a Program, Programs, Procedures and Subprocedures, Using Variables, Another Way to Move the Turtle, Recursion, Changing Variables and Conditionals, Words as Variables, Procedures That Ask for Information: Using Request with Words, and Using Request and Make with Numbers and Conditionals.

Others are Recursion in an Interactive Procedure with Request, Controlling Procedures with a Single Key, Randomness in Procedures, Planning a Program: Guessing Games, Using Variables with Make and Request to Gather Many Pieces of Information, More in Lists: A Survey and a Drawing Program, Circles, Arcs and Spirals, Making Drawings that Move, and Putting It All Together.

The appendices include Workspace and File Management, Differences between Terrapin and Apple Logo, Glossary of Terrapin Logo Terms, and Solutions.

At the end of each chapter, there are exercises for teachers to try specific programming skills. And, thank goodness, the solutions are at the end of the book to check your procedures.

**Strengths.** The preface gives an excellent explanation of Logo as a programming language and the pros and cons of teaching it to children. For example, Burns and Cook criticize the designers of Logo for not having made an effort to teach teachers how Logo works or ideas on the best way to teach the language to children. In fact, the authors of Logo assume children and teachers will just "discover" and "implement" how to use the language effectively when the teacher understands Logo.

"While Logo does allow a lot of play and experimentation with turtle graphics, to really program with Logo you must learn its arbitrary vocabulary, syntax and logic. A method for making a square by using FORWARD and RIGHT may be 'discovered,' but no one will discover how to make a procedure for that square unless ... taught it. The further you go into the language, the more this is true."

Burns and Cook are teachers at Bank Street School for Children in New York City. They have based the book on years of teaching Logo to children and adults. Terrapin Logo is used in the book, but explanations for adaptations of the procedures to Apple Logo are included.

The authors give the best advice possible on how to get the most out of the book.

"It is important to relax and be patient with yourself, especially if you do not have any computer or programming experience. The more you relax, the easier it will be for you to play and experiment with the language. Patience is necessary because programming is a new skill to most people, and, like any new skill, some things take a while to sink in. You will probably make lots of 'mistakes'; everybody does and should. It's by making mistakes and figuring out how to correct them that most of us learn anything. What's nice about Logo is that mistakes can be corrected easily and without embarrassment; it is not like watching your soufflé fall as the guests are waiting, or striking out as your team watches, or putting up shelves that refuse to stay on the wall. In fact, with Logo you can throw whatever ingredients you want together, you can swing as wildly as you wish, and you can put things together in whatever way strikes your fancy. Some of these efforts will work, some of them won't, but the main point is to relax and let your mind imagine as it works."

Enough said! Buy the book if you are new at the game of Logo. It's well worth the investment.

---

*Anne Cairns Federlein is an associate professor of child development at Oakland University, Rochester, MI, and the editor of The Journal of the Michigan Association for Education of Young Children and the INDEX: CHILD CARE newsletter.*

# Welcome to LogoWriter Again

by Sharon Burrowes

If you have been conscientiously reading your issues of LX this year, you probably know that this is another in a series of articles on LogoWriter. However, if this series is new to you, then you should know that this article is divided into three sections: one for those training others, one for those making the switch from "traditional" Logos, and one for classroom teachers. The section for classroom teachers will focus on primary and secondary students since the LogoWriter materials are not specifically aimed at these grade levels.

## Teacher Trainers

One of the joys of using LogoWriter is knowing that it is an evolving product which grows to meet the needs of students. However, that ongoing growth keeps those of us training teachers "on our toes". Just about the time we think we have those introductory LogoWriter sessions "down pat" along comes LogoWriter Version 1.1. (If you haven't yet received your Version 1.1, be sure to write to LCSi. Give them your license number and the kinds of machines for which you want upgrades.)

Version 1.1 includes two very helpful "safety nets": LAST PAGE and UNDO. If you accidentally press *Esc* and save over an old version of a page that you want to keep, you can get the page LAST PAGE from the Contents page. This contains the previous version of the page that you just saved.

Have you ever accidentally pressed CT or CP? Rescue is now available in the form of UNDO. UNDO restores text which you just erased. Sorry, UNDO won't rescue your graphics...so you still have to be careful about CP.

Printing is perhaps the most noticeable difference with Version 1.1. Suddenly everything prints double spaced and with margins. That's fine for a paragraph written for a language arts class, but not so great for procedures. What to do? Use SSPACE and DSPACE. As you have no doubt guessed, SSPACE stands for single space and DSPACE stands for double space. If you would rather have your LogoWriter disks automatically single space, simply follow these steps:

- (1) Start with a new page and enter the following procedure on the flip side:

```
TO STARTUP
SSPACE
CONTENTS
END
```

- (2) Name this page STARTUP.
- (3) Press *Esc* to save this page on a LogoWriter Master disk.

Now when you start your computer with this disk, the page STARTUP will automatically load and the procedure STARTUP will automatically run, causing LogoWriter to be set to single spacing. Then the command CONTENTS will return to the Contents page as usual.

You should also be aware that FORMTOOL is no longer among the tools on the LogoWriter disks you make with Version 1.1. It is not needed now that margins are automatic. Also, you need to mark disks carefully because you can only copy LogoWriter Master disks with the version of the Disk Manager that was used to make them.

Now that wasn't so bad...was it? The new features are well worth the time it will take to make the switch!

If you have been training groups of teachers on an ongoing basis, perhaps you have noticed some repeated problems just as I have. If not, maybe you can inject special notes into your training to help teachers (and students) avoid these pitfalls. First, NAMEPAGE does *not* save a page. More experienced users assume that because the disk is accessed, the page is saved. Caution users to press *Esc* before they leave the computer. In our school system, we tell students to be sure that they see the Contents page on the screen before they leave their computer station.

If you have taught "traditional" versions of Logo to others, you know that a common problem is lost files. There seem to be many fewer problems with lost pages with LogoWriter. However, there is an occasional user who is forever losing pages. Often this can be traced to CT. "Neatness freaks" always clear their screens before saving... oops... no text the next day!

Another source of lost work is found among those who make use of the ability to move up and down in the Command Center. I have noticed some people using the *Return* key instead of the arrow keys to move down in the Command Center. If there is a CT (or CG or CP) in the command center and *Return* is pressed when the cursor is on the line containing that command, LogoWriter erases the text, the graphics, or the procedures. This is a particular disaster for those writing procedures! Tell explicitly those you are training to use the *Return* key *only* when they are sure they want to run the command under the cursor.

Finally, there are two other issues worth mentioning. First, a serious management problem can arise when students learn to define Shapes if there are several students using a disk. Some suggestions are given in the last section of this article. Second, some teachers are sticking so closely to the booklets that they forget what a wonderful tool LogoWriter is. When you are working with teachers, try to plant some seeds that are not in the booklets that relate to the specific material they are teaching in the classroom.

## Making the Switch

Have you yet discovered that procedures written on the flip side of the page are unchanged by LogoWriter? In most versions of Logo, once you leave the editor, Logo "does it's

thing" with your procedure format and there is nothing you can do about it. Not so with LogoWriter. This has the marvelous advantage of being able to add comments and formatting to your procedures. Text that is not between the word TO and the word END is simply ignored when you flip the page. Further you can be free of those unreadable long lines that can easily follow a REPEAT or IF statement.

In fact, this ability to format procedures may dramatically affect the way you program in Logo. I, personally, prefer readable code, so I often write a separate procedure to avoid long lines. For example, suppose you want a classic "spin-square". You can write

```
TO SPIN.SQUARE
  REPEAT 10 [REPEAT 4 [FORWARD 50 RIGHT 90]
    RIGHT 36]]
END
```

On a forty-column screen this will wrap, as it did above. Before LogoWriter, I wrote this program as follows:

```
TO SPIN.SQUARE
  REPEAT 10 [SQUARE RIGHT 36]
END
```

```
TO SQUARE
  REPEAT 4 [FORWARD 50 RIGHT 90]
END
```

With LogoWriter, you might prefer

```
TO SPIN.SQUARE
  REPEAT 10
    [REPEAT 4[FORWARD 50 RIGHT 90]
      RIGHT 36]]
END
```

While this is clearly a matter of personal style, you should make use of the capabilities that LogoWriter offers.

A complaint about LogoWriter that is beginning to surface here and there is "missing" primitives. In order to fit all of LogoWriter's capabilities into the available memory, some of the richness of versions like Apple Logo II and IBM Logo were lost. However, if you have been following the articles in LX this year and last, you are probably aware of the possibility of writing missing primitives using what you have available. Some of the missing primitives are easy to write; others are not. Once created, these procedures can be placed on a page (or pages) and then loaded as tools when needed.

Do you miss XCOR, YCOR, and DOT? These are really not very hard to create.

```
TO XCOR          TO YCOR
  OUTPUT FIRST POS OUTPUT LAST POS
END              END
```

```
TO DOT :WHERE
  MAKE.DOT POS :WHERE
END
```

```
TO MAKE.DOT :OLD.SPOT :WHERE
  PU SETPOS :WHERE
  PD FD 1 BK 1 PU
  SETPOS :OLD.SPOT
END
```

Another bothersome omission is READWORD. You can create a passable READWORD as follows:

```
TO READWORD
  OUTPUT FIRST READLIST
END
```

The READWORDCC procedure for the command center would be

```
TO READWORDCC
  OUTPUT FIRST READLISTCC
END
```

A command that is quickly missed, but not easily reproduced is SETCURSOR (or CURSOR if you are a Terrapin user.) Because of the layout of the LogoWriter screen, you must first fill the screen with blanks and then move the cursor. This, of course, could cause side effects that you may not want. However, here is a rough equivalent of SETCURSOR.

First, run these procedures:

```
TO FILL.PAGE
  LINE TWO.LINES
  REPEAT 4 [TOP CD CD MAKE.COPY]
  TOP
END
```

```
TO LINE
  REPEAT 40 [INSERT CHAR 32]
END
```

```
TO TWO.LINES
  TOP MAKE.COPY
  CU MAKE.COPY
END
```

```
TO MAKE.COPY
  SELECT BOTTOM COPY
  UNSELECT PASTE
END
```

This fills the page with blanks. Now you can use the following SETCURSOR.

```

TO SETCURSOR :WHERE
TOP
REPEAT FIRST :WHERE - 1 [CF]
REPEAT LAST :WHERE - 1 [CD]
END

```

The side effect occurs when you PRINT or INSERT text. The blanks that you have put on the page "move out of the way". You can unexpectedly run out of space on the page this way.

Another omission that may have disturbed you is the lack of TOWARDS. That too is possible...but not easily. You must first have an ARCTAN, which is also not implemented. Don't panic, just type the procedure given at the end of this article and don't worry about what it does...unless you are a mathematician!!!

### In the Classroom

If you have been using LogoWriter since the beginning of the school year, then by now your students have acquired enough skills that they should be able to use LogoWriter in ways that go beyond the booklets. LogoWriter is a delightful tool to use in other subject areas. For example, have students use the word processor in language arts to write (and illustrate) a paragraph that you have assigned. In science, they might produce a diagram for a report or project using a combination of defining new shapes and the word processor. In math they can use LogoWriter to check worksheets that they have done by hand. Art and music can certainly make use of the graphics and sound capabilities of LogoWriter to produce "computer art" or creative songs. In social studies, students might make simple graphs of data they have collected. Yet another neat use in social studies is to have students draw and label maps. You can make transparencies of maps, tape them to the monitor, and then use a simple program like the one given below to trace the map.

```

TO DRAW
DRAW.IT READCHAR
END

```

```

TO DRAW.IT :LETTER
IF EQUAL? :LETTER "S [STOP]
IF EQUAL? :LETTER "F [FORWARD 5]
IF EQUAL? :LETTER "R [RIGHT 10]
IF EQUAL? :LETTER "B [BACK 5]
IF EQUAL? :LETTER "L [LEFT 10]
IF EQUAL? :LETTER "U [PU]
IF EQUAL? :LETTER "D [PD]
DRAW.IT READCHAR
END

```

Even though the LogoWriter materials are designed for intermediate aged children, they are proving to be quite successful with older students -- including high school

students! However, older students may move through the booklets at such a rapid pace that you can't keep up with checking their work.

Steve Ray, who teaches LogoWriter to upper middle school students in the Vandalia/Butler system in Ohio, has developed assignment sheets that are tied to the booklets. For example, Steve has a project called HEART which says: "Create a greeting card design on the screen using at least two stamped shapes and a label. Save it and print a copy." This project requires students to work through pages 11 - 15 in *On Your Mark*, but allows Steve to see if the student has mastered the important concepts. Perhaps such project sheets would be helpful for you!

If you are working with young children and want to use the large type available from PRINTSCREEN, but your text is longer than one page, simply follow these steps:

- (1) Get the first line of your text at the top of the screen.
- (2) Type PRINTSCREEN
- (3) Without moving the paper, move the cursor down until the text you have printed is off the screen and type PRINTSCREEN again.

You can repeat this for as many screens as you have. Be careful, though. If you have graphics on your first page, it will also be printed with your second page of text.

If you are teaching LogoWriter to older students, perhaps as part of a "computer literacy" course, I have found that it is important to add a new "spark" every now and again to pique their interest. Defining shapes keeps those early days of learning to manage the screen interesting. Animation is a natural way to introduce recursion and high school students love doing it. Music is a natural way to encourage modular programming. Once introduced, our lab is filled with songs for weeks to come!

Whatever your students' ages, perhaps you have encountered some management problems if students are sharing disks. Once they learn how to define their own shapes, shape numbers 1 through 10 quickly fill with student creations. At this point, it is best if each student has his or her own disk.

However, if that is not possible, you may need to establish guidelines for creating shapes. You might allow each student particular shape numbers that s/he can redefine. Thus Mary could use 1 - 5 and Sam could use 6 - 10. You could also restrict modification of existing shapes so that a particular student could modify only, say, 11 - 15. Sharing disks can create headaches for you, but it does help teach students to respect other people's work.

Another approach might be to have some class disks on which students are not allowed to modify any shapes, and others on which any or all of the shapes can be changed. You could establish a rule that once a shape is changed, only the "artist" is allowed to change it again.

Regardless of how you deal with this situation, it would be best if you have a plan before problems arise!

### Conclusion

Once again, I have gotten carried away and this article contains a lot of information. If you aren't ready to digest it all now, put it away for future reference. Some of the above ideas will certainly come in handy somewhere in the future!

And now, as promised, here are the procedures necessary for the TOWARDS and ARCTAN operations as implemented in LogoWriter.

```
TO TOWARDS :POINT
  OUTPUT TOWARDS.1 FIRST :POINT - FIRST
  POS LAST :POINT - LAST POS
END
```

```
TO TOWARDS.1 :DX :DY
  IF AND (:DY = 0) (:DX > 0) [OUTPUT 90]
  IF AND (:DY = 0) (:DX < 0) [OUTPUT 270]
  OUTPUT TOWARDS.2 :DY ARCTAN (:DX / :DY)
END
```

```
TO TOWARDS.2 :DY :AT
  IF :DY < 0 [OUTPUT 180 + :AT]
  IF :AT < 0 [MAKE "AT :AT + 360]
  OUTPUT :AT
END
```

```
TO ARCTAN :X
  OUTPUT IFELSE (ABS :X) < 1 [ATN :X][IFELSE
:X > 0 [90 - ATN 1 / :X][-90 - ATN 1 / :X]]
END
```

```
TO ATN :X
  OUTPUT ( ( ( (1.19376 * :X * :X - 4.87776) * :X *
:X + 10.32132) * :X * :X - 18.92477) * :X * :X + 57.2881)
* :X)
END
```

```
TO ABS :X
  IFELSE :X < 0 [OUTPUT - :X][OUTPUT :X]
END
```

---

*Sharon Burrowes is a computer coordinator for the Wooster (OH) Public Schools, and the author or co-author of numerous books and articles on Logo. Her address is 807 College Avenue, Wooster, OH 44691, and her CompuServe number is 73007,1645.*

---

## The Adventures of Jacques and Elsie

*drawings by Linda Sherman*

The post office crush meant most subscribers received their January issue too late to suggest captions for the cartoon. So we telephoned several local LX'ers and discussed with them some caption ideas. Here is the result.

The January cartoon showed the continuation of Elsie's dream. The computer was saying something to her, something so startling that Elsie pulled in her head momentarily!

**FIRST DOESN'T  
LIKE [ ] AS INPUT:  
JUST BEFORE  
LEAVING ADDLIST**



In the March cartoon, it looks as if the students have discovered the TONE or TOOT commands. This is causing Jacques to make a comment to Elsie. What is he saying?



Please send your caption suggestions no later than March 26, 1987, to: Jacques and Elsie, Logo Exchange, PO Box 5341, Charlottesville, VA 22905. Please include your name and address so that we can give proper credit should your caption be chosen.

All captions become the property of Meckler Publishing. None can be returned.

---

*Linda Sherman is a freelance author and artist living in Shipman, VA, with her husband and two-year-old son.*



Q

and

A

by

Jim

McCauley

Q. Why won't this procedure work properly?

```
TO SUMIT :LIST
  IF EMPTY? :LIST [ STOP ]
  OUTPUT FIRST :LIST +
SUMIT BUTFIRST :LIST
END
```

A. You have two problems here. The first is conceptual. You are trying to write a recursive procedure that will output the sum of a list of numbers. Procedures that output values are called *operations*. They are distinctly different from *commands*,

which are procedures that *do* something (move the turtle, for instance), but don't output a value.

The idea behind your procedure is to recursively decompose a list of numbers. Each call to SUMIT will "hold on" to the first number in the list it receives, waiting to add it onto the result returned by another copy of SUMIT.

You are quite correct to begin the last line of the procedure with OUTPUT. This will enable each copy of the recursive procedure to output the sum of the numbers to the copy that called it into existence. Where you have gone astray is in the second line. Your instructions here order the program simply to STOP when the :LIST is empty. Let's look at the simplest job your procedure could possibly have to do: add up an empty list.

```
?PRINT SUMIT [ ]
```

What happens? Your procedure will simply STOP without returning any value for PRINT to print, and you'll get an error message. What is the sum of a list which contains no numbers? Zero, of course. (Some formal-minded mathematicians may disagree, but let that pass.) If you rewrite the second line as:

```
IF EMPTY? :LIST [OUTPUT 0]
```

this problem is solved quite nicely. Actually, this solves the problem for *any* list of numbers. Suppose that this is the job:

```
?PRINT SUMIT [2 4 6]
```

The first copy of SUMIT "holds on" to 2. The second copy (created by the recursion) receives the list [4 6], and it holds on to 4. The third SUMIT gets the list [6], and it keeps the number and passes on the empty list. The last copy of SUMIT gets [ ], so it outputs 0 to the third copy, which can now carry out the addition:  $6 + 0 = 6$ . This sum is output to the second copy, which adds 4 to 6, passing the result of 10 to the first copy, which performs  $2 + 10$ . This result is passed to PRINT, which prints 12.

Or rather, it would, if the other problem were to be fixed.

In the last line, you use the arithmetic operator "+". This is a synonym for another Logo primitive called SUM. If you had written the last line with SUM, it would look like this:

```
OUTPUT SUM FIRST :LIST SUMIT BUTFIRST :LIST
```

Notice that when SUM is used, the inputs *follow* SUM. This is called *prefix* notation, because the operator is placed before its inputs. When the plus sign is used, the inputs occur on *either side of* "+". This is *infix* notation; the operator is placed between its inputs. Infix notation is familiar to us all because it is commonly used in ordinary arithmetic.

(Ed. note: In some versions of Logo, the "+" can function as either a prefix or an infix operation.)

Unfortunately, special rules sometimes have to be used in decoding the meaning of infix expressions. This is one of those times. In your procedure, the "+" is confused: it assumes that you want it to add :LIST to the output from the recursive call to SUMIT. Of course, what you actually want to do is to add FIRST :LIST to the result of the recursive call. You must make this clear through the use of parentheses:

```
OUTPUT (FIRST :LIST) + SUMIT BUTFIRST :LIST
```

With these changes, your procedure will work correctly.

Each month, I will answer a selected question related to Logo. If you have a question, send it along with a stamped self-addressed envelope to:

Jim McCauley  
Logo Exchange Q and A  
876 East 12th Avenue, #4  
Eugene, OR 97401

---

*Jim McCauley is a graduate student at the University of Oregon, studying with David Moursund, and has written Logo articles for many national publications. His CompuServe number is 70014,1136.*

---

## IntLXual Challenges

by Robs Muir

### Monday, Monday...

One of the most important aspects of a hobby or sport is the way that it can transform time. For the participants, deeply involved in their avocations, time seems to dissolve and the hours just "slip" away. Such a pleasurable experience is usually cherished and is often cultivated for its own sake.

Observe Webster's definition of avocation (emphasis mine):

AVOCATION n. 1. archaic: *diversion, distraction*, 2. a subordinate occupation pursued in addition to one's vocation experience for enjoyment.

I am attracted to the archaic definition, implying time and state of mind are directly involved with such activities.

As adults, we can probably all cite experiences that have stopped the clock while we were deeply involved. Archery, mountaineering, tennis, sewing, mathematics, computing(!), and even physics (for Tom Lough, anyhow), all can evoke such deep concentration. The key here is engagement; this engagement keeps us coming back for more. Engagement is a big part of the Logo movement. Engagement is at the heart of Papert's vision for Logo. Logo, like all computer languages, can evoke an intense engagement for its users. And like other activities or sports, the more you do it the easier the engagement becomes.

Sad to say, we don't see enough of this engagement in the classroom! Stopping the clock seems to be a phenomenon of the playground...and perhaps the computer. Let's stop--or at least gain control of--the clock.

Here's a challenge forwarded to me by an editor of a widely read Logo journal. Create a DAY.OF.WEEK procedure which requires three inputs, e.g.,

```
TO DAY.OF.WEEK :MONTH :DATE :YEAR.
```

This procedure would output the day of the week for any date entered. For example,

```
?PRINT DAY.OF.WEEK JULY 14 1942
TUESDAY
```

Challenge #1: Why didn't we quote the month in the following example? Shouldn't we have used PRINT DAY.OF.WEEK "JULY 14 1942? This notation flavors the technique we use to solve this problem. Could we have used dots instead? PRINT DAY.OF.WEEK :JULY 14 1942 would imply a global variable, right?

Solution #1: If we used a procedure to output the days in each respective month, we would not need the quotes. For example,

```
TO JULY
OP 31
END
```

This approach would require twelve procedures.

Challenge #2: Is there a simpler approach?

Tentative Solution #2: "Thirty days hath September, April, June, and November. All the rest have thirty-one save February, which has twenty-eight in fine, and leap year brings it twenty-nine." Here's the Logo equivalent of the children's counting rhyme:

```
TO DAYS.IN.MONTH :MONTH
IF MEMBERP :MONTH [SEPTEMBER APRIL
JUNE NOVEMBER] [OP 30]
IF NOT :MONTH = "FEBRUARY [OP 31] [OP 28]
END
```

On second thought, perhaps this will be the children's rhyme for the future. Gad, what an unpoetic thought!

You'll have to extend this to include leap years and make places for the date and year, but at least we've eliminated eleven procedures from cluttering our workspace.

### Tools for the Future!

While reading the new Reference Manual for *Object-Logo*, a highly recommended new Logo for the Macintosh from Coral Software (PO Box 307, Cambridge, MA 02142), I ran across an elegant solution for the leap year problem. If nothing else, it is an interesting demonstration of logical operations in Logo. It may give you what you need for a solution to this month's challenge.

```
TO leapyear :year
OP AND divisiblep :year 4 OR NOT divisiblep :year
100 divisiblep :year 400
END

TO divisiblep :n :factor
OP 0 = (REMAINDER :n :factor)
END
```

For the ambitious, you might examine the Julian Day calender for another related challenge. You may even find this to be a more interesting approach. March 15, 1987, for example, is the same as 2,446,870 using the Julian Day notation.

Send a listing of your DAY.OF.WEEK program to

IntLXual Challenges  
c/o Robs Muir  
1688 Denver Avenue  
Claremont, CA 91711

We'll send you a sampling of other readers' solutions. Remember to include a self-addressed, stamped envelope. I wonder what day it will get here...

*Robs Muir is a physics and computer science teacher in Claremont, CA, and an instructor at the Claremont Graduate School. His CompuServe number is 70357,3403, and his Blnet address is MUIR@CLARGRAD.*



# International Logo News



Dennis Harper  
Logo Exchange International Editor

**Global Comments**  
*by Dennis Harper*  
*Institute of Education*  
*469 Bukit Timah Road*  
*Singapore 1025, Republic of Singapore*

This month's international selections once again show the wide gap between the realities of implementing Logo into the schools. While Australians attend a conference specifically concerned with plans to design a Logo-based mathematics curriculum, African countries are struggling to answer the question of what area of computer education should be pursued with the very limited funds available. Our African editor puts her finger on a major problem of making use of computers in education -- trying to do too much with too little.

Everyone seems to know what is "best" for the students. Administrators want to use management software, some want word processing, some CAI, others BASIC and others Logo. Then there are some who want to do it all.

The last three days of 1986 found me attending the Comparative and International Education Society's Western Region Conference in Honolulu. The title of the conference was *Education in the Age of Information and Technology: International Perspectives*. In the confines of this column I could not begin to summarize all the papers, so I'll emphasize one of the major presentations.

This presentation was a report on the Stanford/UNESCO Symposium by Liza Loop of Stanford entitled, "Computers in Education: Which Role for International Research?" The symposium was organized by Stanford University under contract with UNESCO, to examine the role of international research on the use of computers in education. Thirty high level specialists from all over the world participated in the meeting. Among them, members of academies of science of socialist countries, leading researchers in western universities, heads of specialized journals on high technology for education and a representative of the World Bank.

In preparation for the symposium, the Secretariat organized specialized meetings in Tokyo (October, 1984), Abidjan (November, 1984), Damascus (March, 1985), Algiers (April, 1985) and Caracas (August, 1985).

Although there were many topics of interest in the report, I searched for references to Logo.

The symposium concluded that evaluations of Logo in the U.S. yielded mixed results. "Some studies showed significant gains in problem-solving skills, including gains in divergent and reflective thinking. But a major two-year study of Logo found no significant effect on cognitive skills." No mention was made as to what studies these were.

The second of the three Logo references stated: "There is a lack in all studies of an underlying theory of learning that can explain why or why not computers will enhance learning. Papert's seminal work is as yet unsupported by firm data. The claim that learning with computers will create new conceptual skills in children has to be supported by evidence. Because learning with computers is at its inception, its effects may take a generation or two to be felt."

Finally, in the recommendations section of the report, the following suggestion was made: "A cross-cultural, multi-year study of the effects of Logo programming and the learning environment at the elementary and secondary level was proposed. Information about the long-term consequences of computers on learning is inadequate. Logo is a case of a fully promoted application which has clearly specified and attractive theoretical objectives. Such a study could take into consideration many of the research questions raised in the symposium."

For more information on the symposium, write to:

Ms. Liza Loop  
School of Education  
Stanford University  
Stanford, CA 94305 USA

It is certainly gratifying to know that Logo is taken seriously by UNESCO. I hope this interest continues and that major studies such as those they recommend are undertaken.

LX readers everywhere are invited to send in Logo news for this section of the magazine. You may mail reports to the continental editors at the addresses given, or directly to me. Everyone benefits when everyone shares information. And our students benefit the most of all. Thank you.

FD 100!

*Dennis*

## Africa

*Fatimata Seye Sylla  
Laboratoire Informatique et Education  
Ecole Normale Supérieure  
BP 5036 Dakar, Senegal, West Africa*

Through this article, I would like to know what the International Logo community thinks about a problem which is starting to be crucial for African Logo users (especially the Senegalese): the conflict between Logo and computer aided instruction software. Maybe some of you have already faced the kind of problem I am going to describe and already found a convenient solution to share with us. Maybe you don't have a solution but have some ideas about this.

Here in West Africa, when people talk about computers in education, they really mean computer aided or assisted instruction (CAI). They are referring to specific computer software to teach or learn a specific school subject.

I cannot count the number of people coming into our Logo research laboratory to ask questions about what kind of software will be appropriate for them or their children to learn mathematics, physics, writing,... The Senegalese research team has collected children's work with Logo and written reports on users' (teachers and children) behavior towards Logo programming and learning other school subjects. The researchers use those results to show how Logo users can change the image they have of themselves and their relationship to learning and teaching.

Even though we have a good team in explaining how Logo can have positive effects on the learning process in general, some people do feel disappointed when we are not able to present them a ready-made software saying things like "understanding mathematics or physics in high school."

At a higher level, software companies are trying hard to convince our decision makers that CAI software will do the most good in solving the problems African schools are facing. These problems are known to be the insufficiency of schools and teachers.

- Even in the cities there are not enough public schools. Private schools are available for people who can afford to pay for their children's education.
- Because of economic problems, the governments cannot hire enough teachers, even though there are enough training schools to provide more teachers than needed.

This lack of public schools and teachers results in overcrowded classrooms. One can count up to one hundred pupils in a primary classroom, from first to seventh grades.

It is difficult for a teacher to teach one hundred children in one classroom and get good results. In fact, only few pupils really follow the lessons. The miracle is that one quarter to one third of the seventh grade children do pass the national exam to attend junior high school.

Of course, if technology can help these teachers carry out half of their job (drill & practice, for example), and get better results, the government should not hesitate to provide

computers and appropriate software for this purpose. Thus, it is very easy to convince people knowing these problems that CAI (and not Logo) is the most efficient use of the computer.

However, there are two crucial problems experts should raise when schools are planning in the use of computers to enhance their educational system.

(1) It is not proven that the use of CAI software would solve African educational problems. Experiments in the real environment should be carried out before making any decision. Obviously, this would take more time than these experts have.

And in the meantime, technology is growing fast, new software with higher performance is being created and spreading throughout the world.

(2) Financial problems -- How much would it cost to buy a computer and CAI software? How many computers do we need in a school? How much would maintenance of the computers cost? How much would teacher training cost? Would it be more or less expensive to build a new classroom and hire a new teacher than buying computers? These and other questions must be answered.

It is clear that teachers with higher levels of education would produce better results. Thus, they should be trained regardless of cost. They should know about computers and the technological tools that are entering the homes and the private schools, so that they can answer children's questions correctly.

Fortunately, in our governments, some people do care about these above-mentioned issues, and, as they do not know the answers, will consult experienced people in this domain before making any decisions.

In Senegal, people who are informed about the computer's power are eager to see this technique in use everywhere. They are not wrong. Some private schools already have their students learning how to use computers. Professional schools are introducing word processing and some other management software in their curriculum.

Concerning the public secondary schools, the government has to make a big decision to give guidelines on how computers should be introduced. Would the students learn computer science as a new subject in their curriculum? Would they use CAI software? Would they use Logo? Would they use both CAI software and Logo? All these questions necessitate deep studies to give some appropriate answers.

As our laboratory is the only one dealing with the use of computers in education in Senegal, we have been invited to make some recommendations. We mainly suggested not to be hasty. Computers should be in the schools now if our governments can afford them. But, besides some training lessons to acquaint people with the tool in the first place, it is not possible to decide quickly what the best use of computers will be.

For this reason, our Logo research laboratory is extending its activities. Beside our research activities mentioned in previous LX issues, the Senegalese Logo research team is writing some CAI programs in Logo to show to the parents, the teachers, and the decision makers that Logo can deliver -- it is a question of choice.

We also made plans to get some CAI programs to:

- demonstrate how to use computers,
- study the impact that CAI could have on African children and whether it would really enhance education, and
- study in what kinds of environments computers should be used and how.

Meanwhile, a Senegalese study group composed of Logo people, computer using educators, and computer education experts are working hard to complete a working document to present to the government.

### Australia

*by Anne McDougall  
Faculty of Education  
Monash University  
Clayton, Victoria 3168 Australia*

**Not Another Logo Conference...** When teachers talk about Logo and mathematics, you often hear questions like:

- "What mathematics do children learn with Logo?"
- "Do Logo skills transfer to school mathematics?"
- "Do real mathematicians use Logo?"

In December 1986, the Computer Education Group of Victoria held a four-day working conference, "Not Another Logo Conference....," to tackle issues arising when Logo is used for teaching mathematics.

The major guest speaker for the conference was Richard Noss, from the Logo and Mathematics Project at the Institute of Education at London University. The other guest presenter was Dr. John Pottage, from the History and Philosophy of Science Department at the University of Melbourne. The thirty or so registrants included participants from the Northern Territory and Queensland, and several from New South Wales as well as more local Victorians.

The "non-organizers" of the conference wanted to avoid "sessions at which we listen to what we could have read." Rather than timetabled regular paper presentations, a less structured format was followed in an effort to provide opportunities for discussion of issues and interests as they emerged, as well as following up questions and points raised in the submitted papers. As it turned out, this structure, or lack of it, proved incompatible with their aim of generating by the end of the conference a collection of materials for later use by teachers in Australian schools. However, the organizers felt that the conference overall was most valuable for the participants, with many hard questions being tackled and many friendships and plans to "keep in touch" being made.

Richard set the meeting off to a good start with a stimulating talk on the importance of reflection in mathematical thinking, and the role for computers in mathematical learning assisting the synthesis of symbolic activity with visual modes of thinking. He outlined some related research

which had been done at the University of London, mentioned some fundamental questions not yet addressed but needing investigation, described a new teacher inservice education project at the Institute, and touched on the controversial question of what is meant by microworlds. (His efforts in this regard deserve no small praise because he and his family had been greeted on arrival in this country by cold wet weather, despite rumors that it's summer here in December, and with it a ferocious influenza virus which he fought valiantly throughout the conference.)

In later sessions, he spoke on "the absolutely indispensable role of the teacher," and work by the London group in the area of teacher intervention, plans for design of a Logo-based mathematics curriculum, more about microworlds, and teacher professional development. An interesting point he made in this last context is the value of giving teachers good access to research results and materials. A final session on his work on variables left many of us wishing there had been more time for him to talk about other parts of his doctoral thesis as well.

As John spoke of mathematical problem solving by ancient artisans, architects, and observers responding appreciatively to their work, he showed a stream of magnificent slides of ancient geometric designs on various artifacts and structures. His criticisms of mathematics curricula which fail to engage the energies of students and his obvious enjoyment of the beautiful mathematics he was showing found an enthusiastic response from the participants.

Other issues addressed in the working group sessions of the conference included curriculum and timetabling constraints, legitimization of Logo activities in school mathematics, teacher roles, teacher professional development, understanding and teaching about recursion, practical matters of equipment and classroom organization, structuring of learning environments and many, many more.

Some copies of the papers prepared for the conference are available from the Computer Education Group of Victoria (CEGV), at a cost of AUS\$25 (including air mail postage). The address is: CEGV, PO Box 88, Balaclava, Victoria 3183, Australia.

**ACEC 87** - The fifth Australian Computers in Education Conference is being hosted by the Computers in Education Group of South Australia, and has the theme Tomorrow's Technology Today. The conference will be held from September 28-30 in Adelaide. Within the major theme, there will be strands addressing a wide variety of sub-themes, including Logo, of course.

The Programming Committee is now inviting expression of interest in the submission of formal papers or in poster sessions. For further information write to:

ACEC 87  
PO Box 931  
Norwood, South Australia 5067  
Australia



by Barbara Randolph

Less talk, and more LogoPals! The ads in this month's column are excerpts from some LogoPal letters received from the students of John Schell of Wisconsin. (To be fair, I closed my eyes and grabbed a bunch to print this time. More next month.) John's students are asking for LogoPals from all over the world. And are they enthusiastic!

**Mike Duffey (Wauwatosa, Wisconsin, USA):** I'm eight years old and in the third grade. My hobbies are playing soccer, football, basketball, golf and videogames. I like videogames because I like computers. I like computers because I like Logo. I want a Logo penpal from New York, New York.

**Vita Lusty (Milwaukee, Wisconsin, USA):** I like to play soccer and chess. I like reading and writing stories. I like making designs in Logo. I like the surprises Logo gives me. The most exciting thing I am looking forward to is a new penpal - from Australia. I'm an eight year old girl.

**Erin Hanson (Milwaukee, Wisconsin, USA):** I'm in the 4th grade and I am 10 years old. I like listening to rock and roll music. I take piano and ballet lessons. I can speak Italian. I would really really like a LogoPal from Italy (my mom was born in Sicily) or from Switzerland or Florida.

**Andy Stilp (Wauwatosa, Wisconsin, USA):** I like to play soccer, football and basketball. I've been studying Logo and my favorite patterns are shapes like snowflakes or a star. I would like a penpal from Las Vegas, Nevada. I'm eight years old and in the third grade.

**David Nagl (Wauwatosa, Wisconsin, USA):** I am nine years old. I have a large family and I am an uncle. I like Logo because I can make different designs. I like football, soccer and sledding in the winter. I would like my penpal to be from New York.

**Philip Vacca (Wauwatosa, Wisconsin, USA):** I am eight years old. I'm pretty good with Logo and like making up my own programs. I also like to swim and make puppet shows. I'd like my LogoPal to be from Wisconsin.

**Christine Phillips (Wauwatosa, Wisconsin, USA):** Hi! I am in third grade. My hobbies are knitting, swimming and cooking. My favorites in Logo are making designs and creating things. I would like my penpal to be from Canada.

**Kathy Erdman (Milwaukee, Wisconsin, USA):** I would like a LogoPal from one of these three places: Spain, Germany, or Italy. I'm nine years old and in 4th grade. I love math and art and reading. I enjoy going outside with my cat and dog.

**Vinnie Piquette (Milwaukee, Wisconsin, USA):** I like to do stunts on my bike and skateboard. I like to play golf, football and baseball. I would like a LogoPal from Paris, France, Japan, or one of the king's kids from Thailand.

**David Cieszki (Milwaukee, Wisconsin, USA):** I'm new at learning about Logo. I'm in the fourth grade. I would like to have a boy LogoPal from Australia or Africa.

Teachers, share these LogoPal ads with your students. Your students can request one of these penpals or one from another place if they choose. Remember, to get themselves matched with one of these LogoPals (or to have their own ads considered for publication), they must send me a brief ad describing themselves and include a self-addressed, stamped envelope. If they are writing from outside the USA, have them enclose international postal coupons for a 1-ounce or 28 gram reply. (To save postage, your class could send their ads in one large envelope.)

If you are attending Logo workshops or conferences in your country, please be a LogoPal "ambassador" and spread the news about the children's LogoPal network.

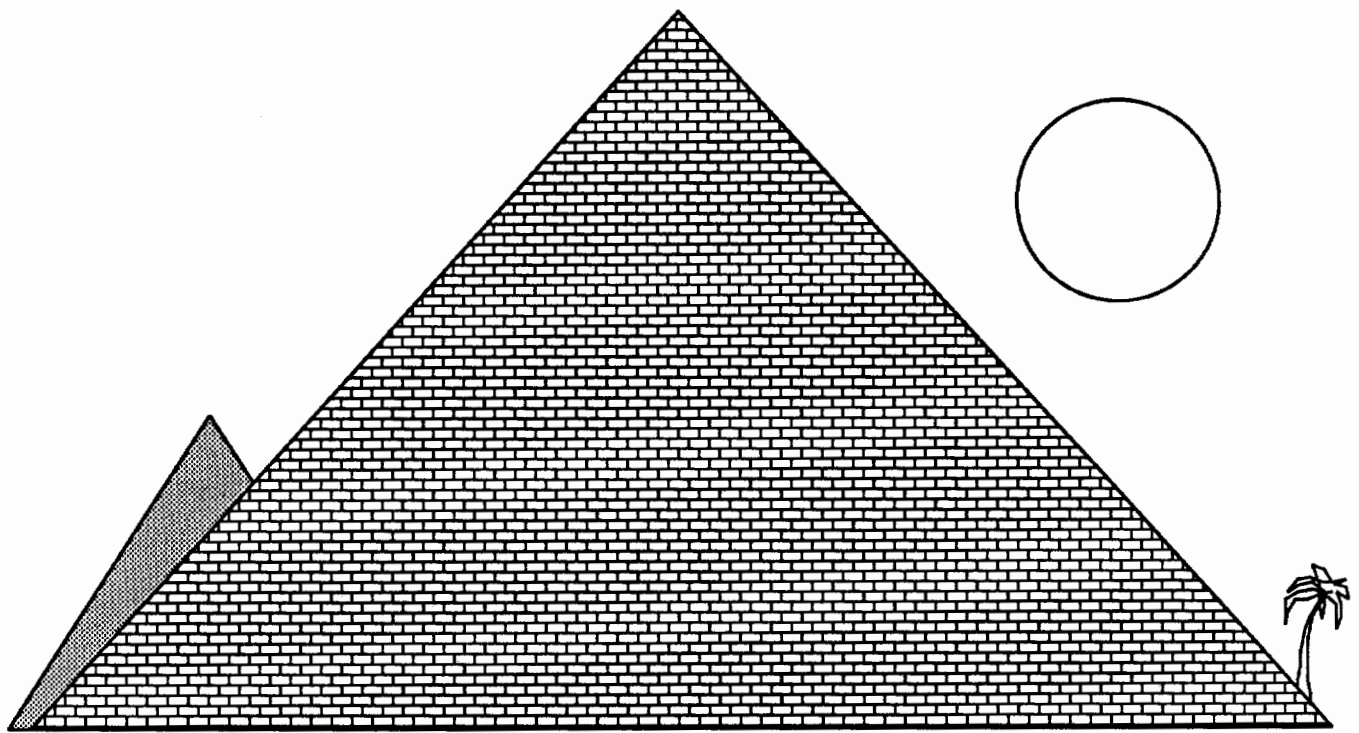
Write to:

LogoPals  
c/o Barbara Randolph  
1455 East 56th Street  
Chicago, Illinois 60637  
USA

P.S. LogoPals: Watch for something special coming in the mail just for you!

*Barbara Randolph is a librarian and instructional media center teacher in the Chicago Public Schools.*





“If something’s not broken, don’t try to fix it . . . support it!”

- Ancient Turtylonian

Terrapin™ Logo, from the original Logo people, is the Logo of choice in education. Terrapin has developed the **LogoWorks™** series of curriculum support materials including *Logo Works: Lessons in Logo* and *The Logo Project Book: Exploring Words and Lists*. Terrapin is also developing a host of Logo tools and applications for its new **Logoware™** product line including *The Logo Data Toolkit*—many more are on the way. So you can be sure that Terrapin Logo is the best possible Logo to use as a foundation for *your* choice of Logo activities—to suit *your* classroom needs. Plus, with Terrapin’s 10-Paks, 20-Paks, networked version and district pricing, Terrapin is the best value for your money. Terrapin. The support you’ve been looking for!

***Terrapin Logo***  
***The Original Logo People***

Terrapin, Inc., 222 Third Street, Cambridge, MA, 02142 (617) 492-8816



# **\*\* Teacher's Pet \*\***



*"...the best robot I've ever seen..."*

—Seymour Papert, M.I.T. LOGO '85 Conference, Cambridge, Massachusetts.

*"We have found the Valiant Turtle's presence has increased the interest level of students and teachers in Logo, facilitated the understanding of Logo commands, and changed personal interaction during Logo work, providing students more time to share and discuss their observations."*

—June 1986 review in *The Computing Teacher* by Jim Ellickson-Brown, Mountain View Elementary School and Doris Mimnaugh, 4th grade teacher, Wilson Elementary School, Corvallis, Oregon.

Meet Valiant Turtle, state of the art in Logo programming! Controlled from your Apple II+, IIe, IIc, IBM PC, PCjr or Commodore 64 computer by an infrared beam, the Valiant Turtle requires no cords or wires to move. Create a design on the screen and watch Valiant draw it on paper! Watch as Valiant executes your commands in three dimensions. Plan some fancy footwork and choreograph a dance in Logo!

Valiant Turtle is the ideal learning tool for students learning Logo as well as a great introduction to the world of robotics. Valiant is simple to use and easy for even the very youngest Logo learner to understand. Complicated and sophisticated ideas are presented in simple, graphic form. Students build artificial intelligence concepts and learn to think about space and spatial relationships.

Valiant Turtle operates with most popular versions of the Logo language. Watch through Valiant's transparent plexiglass dome as the Valiant Turtle draws with extreme accuracy with its built-in pen. Valiant comes completely assembled with easy-to-read instructions and control software. Because it's so easy to set up and use, the Valiant generates much enthusiasm in the classroom! Young imaginations fly whenever Valiant Turtle enters the room!

The Valiant Turtle is available from Harvard Associates, Inc., 260 Beacon Street, Somerville, Massachusetts 02143. Harvard Associates provides full technical support for the Valiant. For more information, or to order your Valiant Turtle, please call (617) 492-0660.