

Logo Exchange

Journal of the ISTE Special Interest Group for Logo-Using Educators

INSIDE

- David Thornburg on Logo
- MicroWorlds Multimedia Tips
- Getting Started with StarLogo
- Logo and the Writing Process
- Stop the Net, I Want to Get Off! By Brian Harvey
- Fractal Explorations

NEW COLUMNS

- Commentary by Dan Kinnaman
- Book Review
- Teacher Feature
- Logo Online
- Logo News



Peter Reynolds
97



Logo Exchange

Volume 16 / Number 1

Editorial Publisher

International Society for Technology in Education

Editor-in-Chief

Gary S. Stager, Pepperdine University
logoexchange@moon.pepperdine.edu

Copy Editing, Design, & Production

S.M. Summerlight

Founding Editor

Tom Lough, Murray State University

Design and Art Direction

Peter Reynolds, Fablevision Animation Studios

Additional Design Assistance

John Boucher, Fablevision Animation Studios

Contributing Editors

Dr. Doug Clements, SUNY Buffalo
Dr. Carolyn Dowling, Australian Catholic University
Alan Epstein, Metasoft
Dr. Brian Harvey, U.C. Berkeley
Daniel Kinnaman, University Associates
Dr. Julie Sarama, Wayne State University

International Editor

Jeff Richardson, Monash University, Australia

International Editor Emeritus

Dennis Harper, Olympia, Washington School District

SIGLogo Officers

Chuck Friesen, President
Steve Sesko, Vice-President
Hope Chafian, Secretary/Treasurer
Gary S. Stager, Editor

Director of Advertising Services

Judy Stickney

1997-1998

ISTE BOARD OF DIRECTORS

ISTE Executive Board Members

Lynne Schrum, President *University of Georgia-Athens (GA)*
Dave Brittain, Past President *MGT of America (FL)*
Cheryl Lemke, Secretary *Milken Family Foundation (CA)*
Michael Turzanski, Treasurer *Cisco Systems, Inc. (MA)*
Chip Kimball, At Large *Lake Washington School District (WA)*
Neal Strudler, At Large *University of Nevada-Las Vegas*

ISTE Board Members

Jose Calderoni *ILCE, Mexico*
Penny Ellsworth *Western Springs School District 101 (IL)*
Cameron Gonzales *New Mexico State University*
Cathy Gunn *Northern Arizona University*
Dennis Harper *Olympia School District (WA)*
Paul O'Driscoll *Salem-Keizer Public Schools (OR)*
Jorge Ortega *FACE/Leon County SD (FL)*
Heidi Rogers *University of Idaho*
Carla Schutte *Technology Specialists (FL)*
Peter Wholihan *Sts. Paul & Peter School, Virgin Islands*

ISTE Committees

LaJeane Thomas *Accreditation and Standards*
Dave Brittain *Awards*
John Ketelhut *Distance Learning*
Michael Turzanski *Finance*
Paul Resta and Gerald Knezek *International*
Jenelle Leonard *Minority Affairs*
Lary Smith *Policies and Procedures*
Gwen Solomon *Publications*

ISTE Executive Officer

David Moursund

Logo Exchange is published quarterly by the International Society for Technology in Education Special Interest Group for Logo-Using Educators. *Logo Exchange* solicits articles on all aspects of Logo use in education.

Submission of Manuscripts

Manuscripts should be sent by surface mail on a 3.5-inch disk (where possible). Preferred format is Microsoft Word for the Macintosh. ASCII files in either Macintosh or DOS format are also welcome. Submissions may also be made by electronic mail. Where possible, graphics should be submitted electronically. Please include electronic copy, either on disk (preferred) or by electronic mail, with paper submissions. Paper submissions may be submitted for review if electronic copies are supplied on acceptance.

Send surface mail to:

Gary S. Stager
21825 Barbara St.
Torrance, CA 90503 USA

Send electronic mail to:

logoexchange@moon.pepperdine.edu

Deadlines

To be considered for publication, manuscripts must be received by the dates indicated below.

Vol. 16, No. 2	June 1, 1997
Vol. 16, No. 3	Oct. 1, 1997
Vol. 16, No. 4	Jan. 1, 1998
Vol. 17, No. 1	Mar. 1, 1998

Logo Exchange is published quarterly by the International Society for Technology in Education (ISTE), 1787 Agate St., Eugene, OR 97403-1923, USA; 800/336-5191.

ISTE members may join SIG/Logo for \$24. Dues include a subscription to *Logo Exchange*. Non ISTE member subscription rate is \$34. Add \$10 for mailing outside the USA. Send membership dues to ISTE. Add \$4.00 for processing if payment does not accompany your dues. VISA, MasterCard, and Discover accepted.

Advertising space in *Logo Exchange* is limited. Please contact ISTE's director of advertising services for space availability and details.

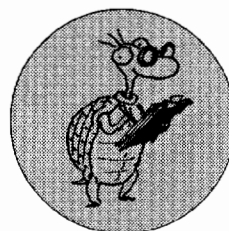
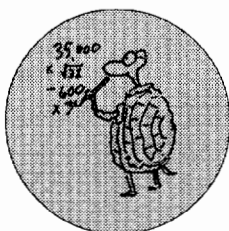
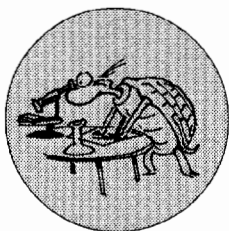
Logo Exchange solicits articles on all topics of interest to Logo-using educators. Submission guidelines can be obtained by contacting the editor. Opinions expressed in this publication are those of the authors and do not necessarily represent or reflect official ISTE policy.

© 1997 ISTE. All articles are copyright of ISTE unless otherwise specified. Reprint permission for nonprofit educational use can be obtained for a nominal charge through the Copyright Clearance Center, 27 Congress St., Salem, MA 01970; 508/750-8400; Fax 508/750-4470. ISTE members may apply directly to the ISTE office for free reprint permission.

POSTMASTER: Send address changes to *Logo Exchange*, ISTE, 480 Charnelton St., Eugene, OR 97401-2626 USA. Periodicals postage paid at Eugene, OR. USPS# 660-130. ISTE is a nonprofit organization with its main offices housed at the University of Oregon. ISSN# 0888-6970

This publication was produced using Aldus PageMaker®.

Contents



ARTICLES

- The Philosophy of Logo: Its Most Important Attribute
David D. Thornburg 5
- The Koch Snowflake Curve and Recursion
Don E. Ryoti 14
- Logo and Thinking Writing: Part I
Leslie F. Thyberg 20
- Teaching Recursion: A Cognitive Profile Perspective
Pam Gibbons 24

COLUMNS

EDITORIAL

- A Word for Logo
Gary S. Stager 2

QUARTERLY QUANTUM

- The L Word:
Don't Be Afraid to Use It
Tom Lough 3

LOGO NEWS

- Forging Tech-Savvy Families
Carolyn Dowling 4

TEACHER FEATURE

- Frank Corley
Marian Rosen 9

STARLOGO STARTERS

- Getting Started
Alan Epstein 10

FOR BEGINNERS

- Tips for Making Multimedia
with MicroWorlds 2.0
Gary S. Stager 12

LOGO ONLINE 17

IN THEIR OWN WORDS

- Stop the Internet,
I Want to Get Off
Brian Harvey 18

LOGO: SEARCH AND RESEARCH

- This Much We Know:
Part I, Mathematics
*Douglas H. Clements
and Julie Sarama* 28

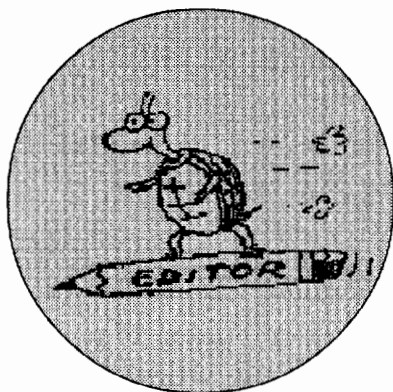
THE LAST WORD: COMMENTARY

- We Choose to Go to the Moon
Daniel E. Kinnaman 31

25 THINGS TO DO WITH LEGO AND LOGO

- Gary S. Stager* 32





A Word for Logo

It's alive! Yes indeed, Logo is still quite alive and an important part of the learning adventure for countless students around the world. According to IBM, more than 1.1 million students use LogoWriter or MicroWorlds in eleven Latin American countries. I'm writing this column from Australia, where I have returned to lead three weeks' worth of Logo workshops for hundreds of Australian teachers. Anywhere I am given a stage to talk about Logo, the room is full of interested educators.

Logo Exchange is here to share ideas among Logo-using educators and to maintain a presence for Logo amidst the noise of the educational computing community. Please share your copies with colleagues and encourage them to subscribe.

This is my first issue as the editor of *Logo Exchange*. My name is Gary Stager, and I currently teach at Pepperdine University's Graduate School of Education and Psychology. I've traveled the world as a Logo evangelist and have spent the past 15 years helping teachers use computers to create constructive learning environments for children. I was one of the founders of SIGLogo, as well as both a vice-president and president of the organization. For several years I was a contributing editor to this very publication. I thank SIGLogo for affording me this opportunity to edit one of the longest running journals dedicated to educational computing in the world. My mission is to share the breadth and depth of Logo with a growing audience of readers.

It is a great honor to edit the journal started 16 years ago by Tom Lough. Tom played a seminal role in the early success of Logo by creating a print-based community of practice. I first met Tom at Logo '85 and was struck by the way he welcomed me and others into the Logo community. Years later I admired the respect which he afforded a 10-year old student of mine. Under his leadership, *Logo Exchange* was fun to read cover-to-cover. I hope to recapture some of that spirit and make Tom proud of what his baby has become.

I must also thank Dorothy Fitch, who worked tirelessly to edit *Logo Exchange* over the past several years. She did a great job and I wish her well in her future endeavors.

A New Look

You may have noticed a new look to *Logo Exchange*. Special thanks to Peter Reynolds and the designers at FableVision Studios in Watertown, Massachusetts, who generously donated their time and energy in crafting the new, updated look of *Logo Exchange*. Peter is known best for his work as creative director at Tom Snyder Productions, where he helped create K-12 materials for more than 12 years. Meet Peter at www.fablevision.com. Thanks to S.M. Summerlight and Ron Richmond at ISTE for their layout expertise.

New Features

I am enormously grateful to David Thornburg for contributing the first in a series of feature articles written by stars of educational computing who have a

special warm spot in their heart for Logo.

Alan Epstein is a teacher educator who will be contributing the StarLogo Starters column for those of you interested in learning to use StarLogo. Carolyn Dowling is a writer and university administrator from Australia who will be contributing book reviews to each issue of *Logo Exchange*. Doug Clements and Julie Sarama continue to share the latest in Logo research with us, while Logo legend Brian Harvey will contribute special assignment articles on a regular basis. Dan Kinnaman is a respected educational leader who has spent several years writing commentary and news for *Technology and Learning*. His former post did not allow him to express his fondness for Logo, so I am pleased to feature Dan's thoughtful commentary now in *Logo Exchange*.

I am thrilled to bring back one of my favorite columns from the Tom Lough era, the Teacher Feature. Please share with us the story of an outstanding Logo-using teacher you admire.

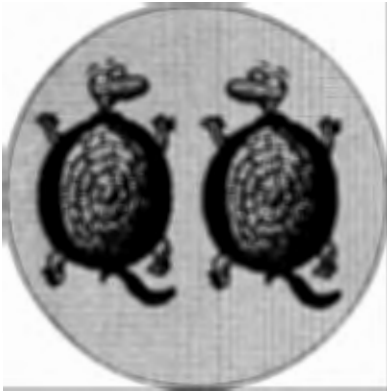
I hope you will enjoy reading this issue and find an article or two that inspires you. Remember—this is your journal! Please share research, kids' projects, teacher education materials, Logo games, programming challenges, letters to the editor and teacher features. Your contributions are always appreciated.

Enjoy!

Gary S. Stager
Gary



Gary Stager
logoexchange@moon.pepperdine.edu



The L Word:

Don't Be Afraid to Use It!

Not too long ago, I was discussing the features of a fairly well known software application with a fellow teacher. When asked about the programming features, I said something like, "Well, it is based on a powerful procedural language derived from Lisp, one of the most widely used programming languages in the field of artificial intelligence."

My colleague looked at me quizzically and then asked, "Hey, are you talking about Logo?"

For just a second, it made me feel like I was trying to recruit someone for a multilevel marketing scheme whose name I was not supposed to reveal until the very end. Then it really made me think, why was I so secretive or hesitant or apologetic or whatever I was about mentioning Logo by name?

Whenever I discover something about myself that I did not realize I was doing, it gives me pause. In this case, a *big* pause!

Luckily for me, Seymour Papert's article, "Educational Computing: How Are We Doing?" appeared in the special 25th anniversary issue of *T.H.E. Journal*. (Congratulations to Sylvia Chorp and her gang!) In the article, Seymour suggested the metaphors of the history of aviation and the evolution of cinema to stimulate thinking about how educational technology will lead to change. He then traced the evolution of Logo in a similar way.

Although Logo was first used mainly for writing programs to draw geometric graphics, later develop-

ments helped it evolve in different directions. Incorporating the Logo programming language into an environment with Lego building blocks, motors, and sensors opened the world of computer control and programming to many more children. Idit Harel's concept of working with children as designers of complete educational software packages provided opportunities for students to work with Logo for longer periods of time—and to produce something more substantive than just a set of procedures. Finally, the power and potential of the World Wide Web offers the opportunity for

Logo has not disappeared. It has not been thrown in the fad trash bin. It has not failed our children. . . . It has evolved and grown and spread and promoted the next exciting stages of educational technology and programming evolution.

students and teachers alike to interact with a virtual community whose collective knowledge about Logo (and other matters) greatly exceeds that of any one individual's.

By suggesting that the problem of whether Logo failed or succeeded is dissolved rather than solved, Papert pointed to what seemed to be the root cause of my hesitation to say the L word.

Logo has not disappeared. It has not been thrown in the fad trash bin. It has not failed our children. Rather, it has evolved and grown and spread. It has extended into many other promising areas of education and, by doing so, has promoted the next exciting stages of educational technology and programming evolution.

Not bad for a computer language "just for kids."

Please excuse me. I have to go now. It's time for my self-imposed exercises.

```
to exercise
repeat 100 [say "Logo wait 10]
end
```

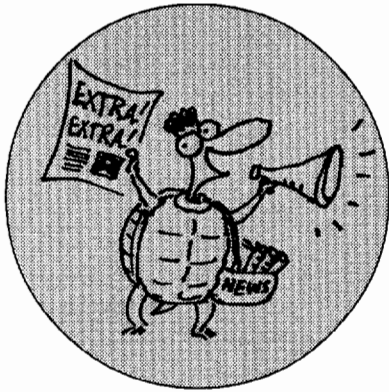
As always,

FD 100!

PS: A special welcome to our new editor, Gary Stager! I have known and worked with Gary for many years and know that his energy and enthusiasm levels are virtually unmatched, especially in Logo matters!



Tom Lough, Founding Editor
70020.223@compuserve.com



Logo News

Logosium '97 at NECC in Seattle

Logosium '97 was held at John Hay Elementary School in Seattle, Washington, on June 29. The event was sponsored by ISTE SIGLogo and the Logo Foundation. The Logosium '97 organizer was Marian Rosen.

Fifty participants represented 12 states and four countries. Some were new to Logo, while others have been using Logo for a long time.

The Logosium '97 schedule included a full day of activities. After a few introductions by Marian Rosen, Logosium's 40-minute concurrent sessions began. As you would expect at a Logo conference, a great variety of topics were explored, including: new implementations of Logo, games, integrating Logo into the classroom curriculum, math with Logo, language arts with Logo, art, inventions, randomness, and robotics.

The keynote address, "The State of the Turtle," was given by Logo Foundation Director Michael Tempel.

Indeed "The State of the Turtle" was very much on all participants' minds. Marian Rosen pointed out that much of what is good in educational technology really has its roots in Logo, which means it still represents some of the best in educational technology. In our search for higher standards in education, Logo is one of the first places to look.

—Dwight Harris
Seattle, Washington
dharris@cks.ssd.k12.wa.us

LCSI News Run MicroWorlds Projects on the Web!

A plug-in for the Windows 95 versions of Microsoft's Internet Explorer and Netscape's Navigator 2.5+ was released by LCSI at NECC '97 in Seattle. The new plug-in enables Internet surfers who don't own MicroWorlds to experience the interactivity and powerful animation capabilities of MicroWorlds 2.0 (Win 95) via the World Wide Web. Any project created using MicroWorlds 2.0 (Win 95) may be viewed on the Web using the plug-in. The plug-in is available for free downloading from LCSI's Web site (<http://www.lcsi.ca>).

Popular "Build-Your-Own" Feature of LCSI Site Licenses Will Continue

LCSI will continue to offer all registered site license holders of its MicroWorlds, Math Links, and Turtle Math software the opportunity to purchase copies of any of its software titles at the special rate of \$25 (U.S.) per copy (the award-winning My Make Believe series will be available for \$15 per copy).

LCSI is able to offer these prices to its best customers by eliminating expensive packaging and print documentation. All such purchases come in generic packaging with full documentation provided on CD. Best of all, software purchased under the LCSI "Build-Your-Own" feature may be resold by the site school to its students

and staff, and schools may purchase unlimited copies for any platform—regardless of the platform of the site license.

Since its introduction in December 1996, many North American schools have embraced this unique purchase option as an effective fundraising vehicle and an inexpensive way to get excellent educational software into the hands of their students.

Special Pricing for Trade-Up to MicroWorlds by Competitive Logo Users, Including HyperStudio

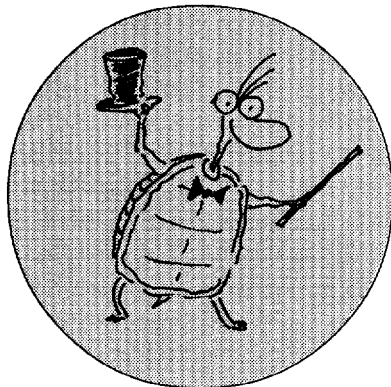
LCSI will offer a special trade-up price of \$599 (U.S.) for a single platform site license of MicroWorlds (DOS, Windows 95, or Macintosh) to owners of competitive commercial versions of Logo, including HyperStudio. This offer will be available for the 1997–98 school year directly only from LCSI.

For more information about any of the above items, contact LCSI at 800/321-5646 or by e-mail at info@lcsi.ca.

Terrapin News Terrapin Software and Harvard Associates Have Merged

Terrapin Software, makers of LogoPLUS for Macintosh and Apple II, and Harvard Associates, makers of PC Logo for Windows and DOS, have merged to offer a broader line of Logo and Logo-related products to their customers. The combined company offers

See LOGO NEWS (Page 7)



The Philosophy of Logo:

Its Most Important Attribute

by DAVID D. THORNBURG

I don't remember when I first heard about Logo or from whom I heard it—probably sometime in the 1970s when I was at the Xerox Palo Alto Research Center (PARC). Alan Kay and his team had developed Smalltalk as an extensible object-oriented language that was comprehensible by youngsters, and which allowed them to construct elaborate microworlds. Like Seymour Papert, Kay was a Piagetian constructionist in the finest sense of the word, and this philosophy emerged in Smalltalk's design just as it did in Logo's development thousands of miles to the east. As I walked into the learning research area at PARC, I would see kids of all ages making magical constructions on the Alto computers we had built from scratch. (The Alto was probably the first "personal" computer—almost certainly the first to incorporate anything like a graphical user interface. For readers who want to know what computing was like for us in 1973, leave your monitor, mouse, and keyboard on your desktop and stick the rest of a Macintosh in a two-drawer file cabinet.)

One thing that caught my eye at the time was Smalltalk's turtle graphics capabilities, because wonderful graphical images seemed popular with kids and grown-ups alike.

I left PARC shortly after the Apple II and Atari computers hit the scene. It seemed natural for someone to create a programming language better suited for kids than the rudimentary BASIC with which the machines were shipped. While Logo for PC's was being developed on the East Coast, a small group of us at Atari created Atari PILOT—an extension of the text-based PILOT language originally developed by Dr. John Starkweather at UC San Francisco. Atari PILOT was the first language (to my knowledge) that brought turtle graphics to inexpensive personal computers. By using a simple set of primitives, PILOT allowed youngsters to create powerful programs without them having to divert their attention from the task at hand to master the arcane mechanics of a traditional programming language.

When the Milton Bradley "Big Trak" was introduced, the world had a cheap programmable robot that used turtle graphic commands. Those of us who were interested in kids using computers to develop thinking skills were delighted! By taping a pen to the front of the Big Trak, "squirrels" and other familiar Logo constructions could be made on butcher paper.

The point is that, by the time I saw Logo, I had already been exposed to three other programming environments that supported the idea that technology could be a tool to help develop thinking and problem-solving skills.

The initial momentum behind Logo was great to see, yet in the early 1980's many of us were frustrated by the forced inclusion of BASIC programming into the "computer literacy" curriculum of most schools. We railed against the inelegance of programming in Dartmouth BASIC, but, deeper than that, we bemoaned the fact that BASIC did not

I hope that Logo's underlying philosophy continues to be the driving force for a pedagogical movement that embraces all Logo-like tools for the creation of interactive microworlds.

reflect the underlying philosophy of Logo—one that linked pedagogy to programming in a tight bond. Logo's central theme was that the journey was the reward. The act of creating a program from scratch and debugging it was where the learning took place. Once a Logo task was complete, the learner had accomplished two things: She had not only created a working program, but also, and more important, had developed or refined problem-solving strategies that could be applied to other tasks, whether computer-related or not.

In short, many of us believed then (and now) that the effective use of Logo in schools required a fundamental shift in our thinking about education—a move from the intellectual bulimia of binging and purging information to the realization that real learning takes place when the connections are forged through the learner's own constructions. This philosophical connection led Logo to be seen as a "religion" by some outsiders. Much the same reaction—almost like holy war—can be seen today when a few Wintel buffs are tossed into a room full of Macintosh users.

And yet Logo refused to die! In some countries where I worked (especially Brazil), Logo was—and remains—the dominant programming tool used by youngsters.

Now in the late 1990s, I'm asked if I remain a fan of Logo. "Yes"—with a special caveat. Many Logo users started with the language and got the philosophy second. In my case, it was the other way around. I had seen kids using computers in "Logo-like" ways years before personal computers entered the market and when Logo was an interesting LISP project at BBN.

In fact, I believe in the philosophy more than I believe in Logo.

Several tools allow students to construct computational microworlds in which concepts can be tested and modified until the system behaves in the manner desired. These tools are Logo-like in my view, and some might even be better for some tasks than Logo. If we accept Abraham Maslow's notion that every problem looks like a nail if the only tool we have is a hammer, then our children need all the tools they can get.

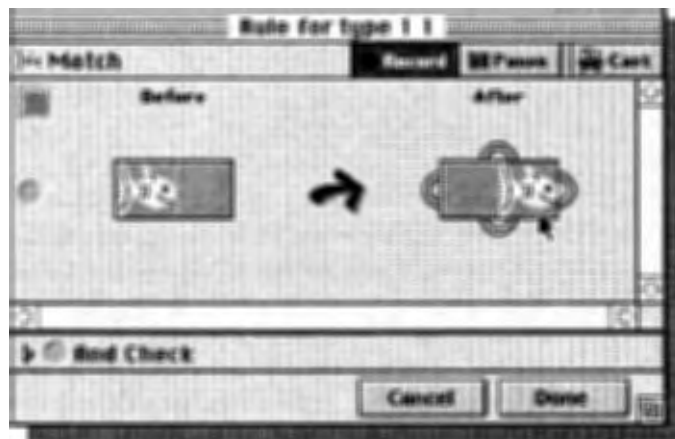
One of my favorites had its genesis in the early 1970s in the work of David Canfield Smith (also a PARC alumnus), who developed the concept of programming by demonstration. His research in this area continued at Apple and can be seen in the modeling language Cocoa (currently available in design release form at [\[cocoa.apple.com\]\(http://cocoa.apple.com\)\). For some tasks, Cocoa is much better than Logo. Like Logo, Cocoa embraces a constructivist philosophy, but their similarities end there.](http://</p>
</div>
<div data-bbox=)

Logo was always promoted as a language with no floor and no ceiling. In fact, Logo has a floor (albeit a low one), and until recently it also had a low ceiling. Logo's ceiling has been removed with the advent of StarLogo (<http://el.www.media.mit.edu/groups/el/Projects/starlogo>) at the expense of making the language more cryptic to the uninitiated. Cocoa's programming environment can be mastered by preliterate children, and yet it is powerful enough to be used for modeling predator-prey systems and other similarly complex microworlds.

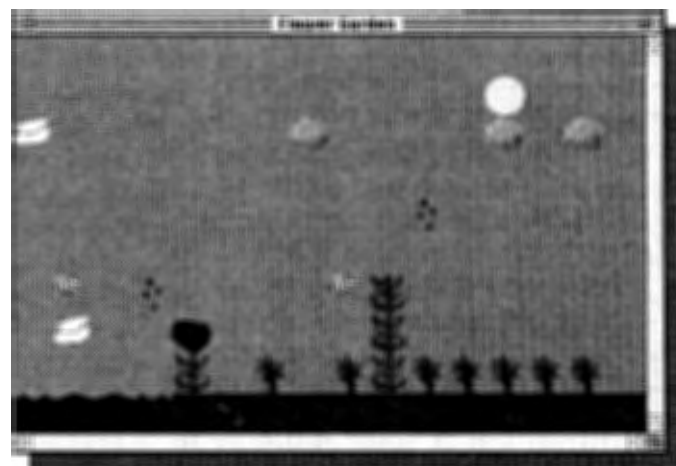
I see Cocoa primarily as a tool for crafting highly interactive and interconnected microworlds. A Cocoa world is built from objects that obey sets of rules. Once a variety of these objects are placed on the screen, they interact with one another and their environment according to their respective rules. Because the objects have one or more graphical representations, the user can see how the world works. If it does not behave as expected, then the user can double-click on any instance of an object to examine its rules and change them if needed.

Rule creation is performed by looking at an image of a "before" state and then creating an "after"

state that shows the status of an object after one tick of Cocoa's clock. For example, if you want to create an object that moves across the screen from left to right, you draw the object, select it with the "rule" tool, and drag the "after" image of the object one square to the right. Because each type of object can have many rules associated with it, a microworld with several types of objects can display quite complex behaviors typical of real systems.



For example, the following figure shows a snapshot of a richly animated flower garden world in which water evaporates and forms clouds. Clouds then produce rain that falls toward the ground. The rain sprouts weeds or flowers (depending on whether flower seeds are in the ground), causes a sprouted flower to grow or set a blossom, or causes a blossom to open. Bees pollinate open flowers, which then produce seeds, and so on.



The behavior of worlds created in Cocoa is easy to examine. Seymour Papert has often talked about the importance of being able to lift the hood to see how something works. This is the appeal of Logo (and of HTML, for that matter). Cocoa's worlds are easy to examine—even by those who did not create them.

So, what is Logo's future? I hope that it and its derivatives gain adherents. In so doing, however, I hope that Logo's underlying philosophy continues to be the driving force for a pedagogical movement that embraces all Logo-like tools for the creation of interactive microworlds—tools such as Stella, StarLogo, and Cocoa.

Logo is one of many wonderful tools for promoting the kinds of thinking that every learner will use for the rest of his or her life. Let's put the development of these thinking skills first and ensure that every tool that promotes these skills flourishes in its own way.



About the Author

Dr. David Thornburg is the director of the Thornburg Center and Senior Fellow of the Congressional Institute for the Future. Through his presentations he reaches more than 100,000 educators around the world every year. He has written numerous books, including one on the Big Trak, two on PI-LOT, seven on Logo, and several on the impact of emerging technologies on education. This article was written in Olinda, Pernambuco, Brazil. Additional copies of this article (in both English and Portuguese) can be downloaded from the Thornburg Center's Web site (<http://www.tcpd.org>).

David Thornburg
dthornburg@aol.com.

a wide range of software, curriculum materials, and robotics to enhance the Logo experience and can now provide cross-platform discounts on Logo for schools with multiple types of computers. Contact them at 10 Holworthy St., Cambridge, MA 02138 USA; ph. 800/774-LOGO or 617/547-LOGO (547-5646); fax 800/774-4610 or 774-4617; e-mail: info@terrapiinlogo.com; <http://www.terrapiinlogo.com>.

PC Logo for Windows Version 2

PC Logo for Windows Version 2 has been released. This new version of the popular Logo for Windows incorporates the many suggestions made by customers during the last two years as well as several new features that make the language both easier to use and more powerful. The new Turtle Center, Multiple Turtle Center, and built-in shape editor enhance traditional Logo programming by offering a graphical user interface for executing many commands. Users can choose between standard and expert modes and add their own primitives, menus, and online help to customize PC Logo for their own needs. PC Logo for Windows 2 is available on disk or CD-ROM and includes 6 Mb of example programs to illustrate the new features and inspire Logo exploration.

In-depth descriptions of Terrapin's Logo products, along with other Logo information, are available on the company's new Web site: <http://www.terrapiinlogo.com>.

MSWLogo

What is MSWLogo? It's a free version of Logo that is a joint effort between Brian Harvey (University of California, Berkeley) and George Mills (Softronics Inc.). MSWLogo's goal was to make the language current by enhancing the language that handles graphical user interfaces and event-driven programming. By enhancing the language, students can learn to build a

Windows application that looks and behaves much like other familiar Windows applications. With these enhancements, students can build multimedia games that use animation, sound, networking, and more.

However, MSWLogo does not do any of the actual work; users still must learn the same concepts any Windows programmer—but in the spirit and ease of Logo. Check out MSWLogo version 5.3 at <http://www.softronix.com>.

StarLogo

StarLogo 2.0 has many new features including:

- turtle shapes
- paint tools
- adjustable graphics window
- PPC and 68K native versions
- turtle breeds
- turtle monitors

New Features and Sample Projects

Download StarLogo 2.0 and read about new features at

<http://www.media.mit.edu/~starlogo/download>

For updated documentation and sample projects, go to:

<http://www.media.mit.edu/~starlogo>.

A StarLogo users mailing list is available through starlogo-users@media.mit.edu. For more information, check:

<http://www.media.mit.edu/~starlogo/community/>.

Dale Seymour Publications

Dale Seymour Publications publishes a terrific set of math "replacement unit" books for elementary school teachers called *Investigations in Number, Data, and Space*.

See LOGO NEWS (Page 30)



BOOK REVIEW

Forging Tech-Savvy Families

by **CAROLYN DOWLING**

The Connected Family: Bridging the Digital Generation Gap

Seymour Papert

1996, Longstreet Press (Atlanta)

ISBN: 1-56352-335-3

This book, with its associated CD and Web site, is a call to arms. It is an attempt to mobilize a group of people who until now have been relatively powerless in relation to the education of children, namely, parents. What has changed the equation that balances the relationship between child, home, and school is, according to Papert, the greatly increased presence of powerful computing technology in the home. This opens the potential for building a whole new learning culture within the family setting—a culture based on modes of learning rarely fostered within formal schooling environments, and one inclusive of family members of all ages.

In contrast to the openly polemical stance of *The Children's Machine*, this book's tone is more intimate, even "chatty," more collaborative, and, perhaps above all, a far more constructivist learning experience in itself. It would be simplistic to attribute this solely to the inclusion of material in other media (after all, do-it-yourself texts have a long and honorable history), but the CD and Web site certainly provide easy and enticing access to a range of illustrative activities; there are clear advantages to the older generation in having everything "there" and without the

need to forage for cardboard and glue. Through these means, along with a segmented, modular structure that, together with the introduction of "Hot Words," provides some foretaste of a hypertext environment, this more-than-book makes a serious attempt to exemplify its subject matter by having the "reader" undertake the types of experiences that are the core of the content. In this way the boundaries of reading, understanding, and learning are gently and most persuasively stretched. The delicate commercial question of how much the customer should be allowed to try before buying is generally well handled, given the slightly awkward situation that Papert is promoting a specific set of products (albeit with good reason) and is strongly associated with several of them. A pleasing feature of the "practice" materials is that enough features are usually provided to allow a reasonable depth of experience.

Mixing media, of course, is a tricky business, and the extent to which the three elements of the publication interact smoothly will differ by individual circumstance. Glitches and infelicities can occur for idiosyncratic reasons. In my case, the CD was able to take over my computer and not only turn it off, but also turn it on again; this was definitely "bad magic" and curiously out of keeping with the usual Papertian emphasis on the user controlling the technology! My failure to find the much vaunted "Leaf Menu" on the ConnectedFamily.com Web

page, and my consequent venting of frustration (much to the amusement of the *Logo Exchange* editor) resulted from a personal habit of never opening windows to full screen size. Apart from that, I had fun!

Despite very defensible claims to universality, particularly in relation to that attraction between children and computers that appears to operate over and above cultural, economic, and other circumstantial boundaries, in another sense this is a strongly "situated" work. A considerable amount of its argument and many of the details of its illustrations and examples are quite explicitly localized through reference to specific localities, school systems, educational practices, and even identifiable individuals. (And what, as one Australian reader asked me this morning, is America Online?) Although this is in some respects a strength, presumably in particular for American readers, it provides an unfortunate excuse for a wider readership to disengage from the issues under consideration on the grounds that, of course, "It's not quite like that for us."

This is a pity, given that we can strongly argue that the most interesting level of discussion transcends the particulars of place and time (to an extent) and perhaps also to some of the specifics of the technology. This is not just a book about choosing software: It is about the nature of thinking and learning; it is about the "point" of

See **BOOK REVIEW** (Page 27)



TEACHER FEATURE

Frank Corley

by **MARIAN ROSEN**

Frank Corley teaches mathematics to seventh through 12th graders and chairs the mathematics and computer science department at Priory School in St. Louis, Missouri. Each summer he teaches two courses—architecture and mathematical investigations—at the Missouri Scholars Academy, a program that brings the state’s “brightest rising juniors” to the University of Missouri at Columbia for a three-week residential program. Frank uses Logo—the pedagogy and the software—in all of his teaching.

I asked Frank how he “found” Logo. He said that in 1988 he was a computer science graduate student who had decided to return to secondary teaching when he happened to read Seymour Papert’s book, *Mindstorms*. The book caused him to “freak out!” He saw Logo as a philosophy of education that was constructivist, let students build their own knowledge, and was open-ended and exploratory. He also viewed Logo as a way to teach LISP, a language he loved, to younger students.

Frank suspects that most secondary math teachers erroneously think of Logo as a “language for little kids” and ignore programming in favor of technology tools. Although he has found other software and graphing calculators somewhat useful, he thinks that nothing has the flexibility of Logo. He notes that his students can write programs according to their individual style and can produce either graphic- or number-generated proofs to the problems posed. Any idea can be ex-



pressed mathematically, and the students are not limited by the preconceptions of a canned piece of software. For Frank, programming in Logo is a way to translate one’s ideas into a formal language. This is a kind of mental discipline that aids problem solving and has a general intellectual benefit.

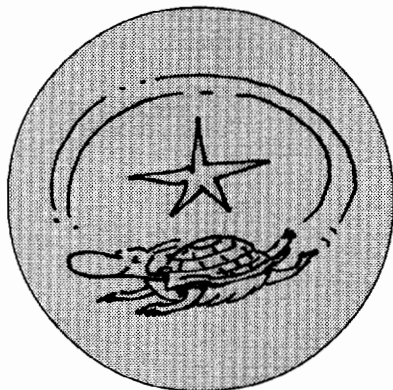
Frank’s ninth-grade geometry and summer mathematical investigations courses have the same basis. He poses a “difficult, big, open-ended” problem. Class members discuss various mathematical approaches that seem worth investigating. When they have refined several possible solutions, kids go to the computer to simulate the approach in which they are interested and look at the results from the computer. Then the students regroup to report what they’ve learned and to debate the strengths and weaknesses of each algorithm (program).

Here are three of Frank’s favorite student challenges.

- **Random Chords:** Given a random chord, what is the probability that its length is greater than the length of a side of an equilateral triangle inscribed in the circle?
- **Random Triangle Formation:** Given a line segment that has been broken into three parts by randomly picking two points on that line segment, what is the probability that the three parts can form a triangle?
- **Greater Than One:** Randomly choose a decimal between 0 and 1 and continue to do so until the sum of all the decimals is greater than 1. On average, how many decimals do you need to generate?

Frank poses the starter questions and helps direct the discussion, but otherwise the kids are in control. He has the confidence to run a discovery-based classroom and says, “I don’t feel I need to know the answer to everything.” As a matter of fact, he resists looking up the “correct” answer to the Random Chord problem because he thinks his reaction to the logic of the students’ thinking is better because he does *not* know. Students sometimes look up solutions on their own after class—and tease Frank about telling him—but so far his ignorance remains blissful.

See CORLEY (Page 32)



Getting Started

What is StarLogo?

StarLogo is a relatively new variant of the Logo language that adds easily applied but powerful features that can be used to study dynamic systems. Examples might be insect colony behavior, economic relationships, or social interactions. Logo lovers who have used more traditional versions will still be able to program the usual turtle geometry applications. StarLogo, however, opens up a whole range of new exploratory possibilities.

StarLogo was developed as a tool for studying decentralized systems and emergent behavior and is geared toward K-12 students. Mitchel Resnick, a designer and researcher at the MIT Media Lab, observed the difficulty many people have in understanding that many systems do not have leaders. Bird flocks, the New York Stock Exchange, and traffic patterns are examples of decentralized systems whose "whole" is directed by the individual actions of their parts—there is no "orchestra leader" involved. In fact, individual behaviors can evolve from interactions with other individuals, and thus a form of artificial learning can take place.

In decentralized systems, aggregate—or whole group—behavior emerges from these individual interactions. Often this emergent behavior cannot be predicted easily from individual actions. For example, Resnick wrote a StarLogo program to simulate a highway traffic pattern.

Cars were given a simple set of rules that instructed them to increase their speed until they found themselves close to another car ahead, and then to slow down. The surprising result was that as the cars moved in one direction, a "wave" of slowdowns was observed to move IN THE OPPOSITE DIRECTION!

StarLogo affords its user the opportunity to develop, test, and analyze models derived from innumerable real world phenomena. Those looking for a tie-in to curriculum standards will note, for example, that the National Research Council's *National Science Education Standards* direct curriculum developers and implementers use extensive modeling as an integral part of the learning process. The standards not only highlight modeling and systems, but also offer descriptions and rationales for their use. StarLogo neatly fulfills these requirements with a modeling environment whose semantics are completely dictated by the user.

How StarLogo Differs from Other Logos

Several major innovations have been added to StarLogo to allow these kinds of explorations. First, a large number of turtles can operate at the same time. This idea is not new: Other Logo systems have employed multiple turtles and sprites. What is different in StarLogo is the language's parallel nature. When a user writes a procedure, it affects all turtles simultaneously. One can still pick out an individual

turtle or a set of turtles to which a command or procedure applies, but the default has all turtles running each procedure or command. A good illustration of this can be seen in the simple StarLogo program:

```
to move
clear-all
create-turtles 100
fd 30
end
```

After clearing the window, StarLogo creates 100 turtles, centered at (0,0), each with a heading distributed from 0 to 360 degrees. The "fd 30" then causes each turtle to move forward 30, instantly resulting in a circle of dots with radius 30.

Another innovation is the data that are available to each turtle. Because each turtle runs the same procedure, a good way to distinguish turtles and their behaviors is to change their individual data. Procedures can then branch according to the values held by the turtle data.

For example, we might want to simulate turtle aging. Each turtle might have a variable—say, "age"—that corresponds to how old it is. As the turtles forage, with each tick of the clock it increments its age variable by a little bit—say, two-tenths. In addition, each turtle changes color as it grows older, until it dies at age 100. You can try this set of procedures yourself. Enter them in the Procedures window and then enter "setup go < RETURN >" in the Command Center:

```

turtles-own [age]
to
  setup
  ca crt 100
  setage random 40 setc
  blue
end
to go
  move
  setage age + .2
  if age > 70 [setc red]
  if age > 90 [setc orange]
  if age > 100 [die] go end
to move
  rt random 50
  lt random 50
  fd 3
end

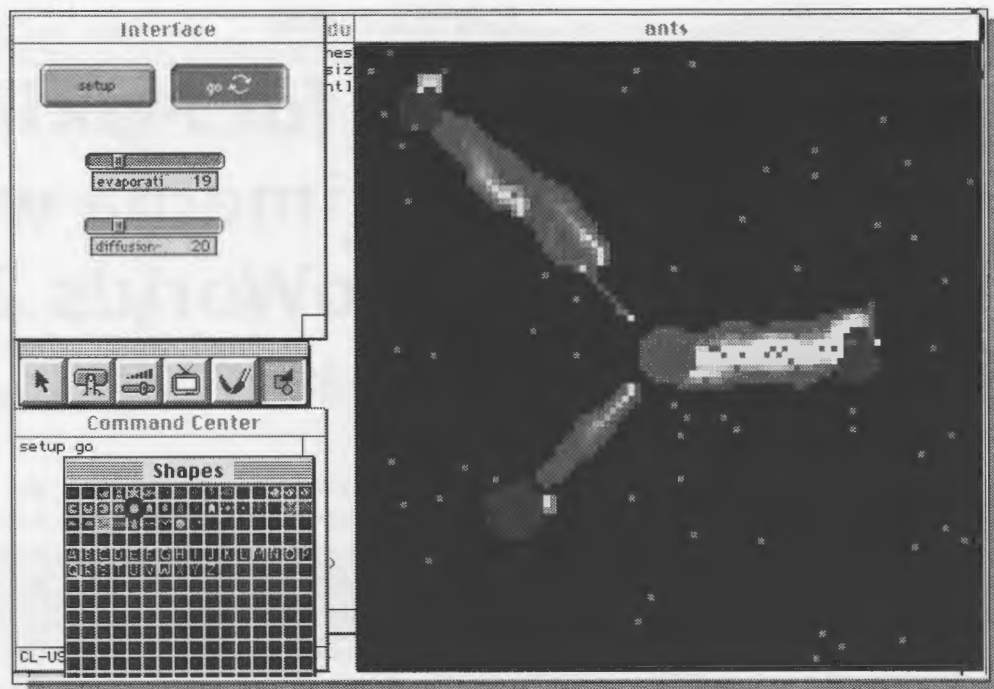
```

The “turtles-own” command sets up any variables to be used by all the turtles, while “setage random 40” initializes the age variable for each turtle with a number between 0 and 39. The “move” procedure keeps each turtle on its toes by randomizing its heading.

Although the turtles are running the same procedure, their data values for age help distinguish their individualized behaviors. If you are feeling especially energetic, change the program to cause the turtles to slow down as they age.

A further innovation is the introduction of patches. A patch resides in each location in the window. They can have color, other attributes, and—like turtles—their own data. Patches can also have procedures that act on them. Even more interesting is the way patches and turtles can sense one another. In other words, turtles can run procedures that test what is being done with the patch below them, and the patches can run procedures that test what is being done with the turtle(s) above them. In this way, the patches can easily model the environment on which the turtles act. We’ll explore using patches in the next article.

StarLogo provides a rudimentary but useful graphing facility for plotting the results of a simulation run. For example, a simulation of predators and prey



might produce time-related data for the population count, which could be plotted in real time. Eight plot pens are provided, and, in addition to time, the X-axis can represent other variable types.

Although not radically different from other dialects, the syntax of the StarLogo language appears simpler in practice. Procedures typically do not need arguments, and variables are declared as belonging to turtles, patches, or the entire program (globals). If all Logo vendors, commercial and shareware, could agree on a strict language standard, then I might take issue with some of the differences. In many ways, the StarLogo language is more accessible to students who use it.

How to Obtain StarLogo

StarLogo is free from the MIT research project that developed it. Web browsers can find it at:

<http://lcs.www.media.mit.edu/groups/el/Projects/starlogo/>

The package comes with reference documentation and a collection of diverse modeling examples. At this time, only a Macintosh version is available. A PC version is in development, but no release date has yet been set.

What's Next?

In next issue's column, we will write a program to simulate the behavior of termites collecting wood chips. See you then.

References and Further Reading

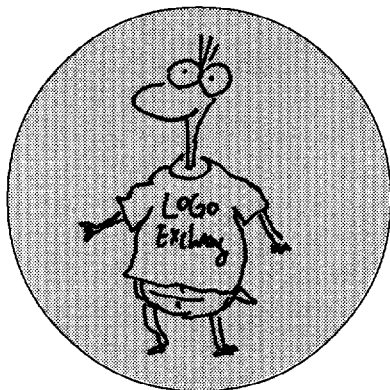
- National Research Council (NRC). (1996). *National science education standards*. Washington, DC: National Academy Press.
- Resnick, M. (1994). *Turtles, termites, and traffic jams*. Cambridge, MA: MIT Press.
- Resnick, M. (1996). *New paradigms for computing, new paradigms for thinking*. In Y. Kafai & M. Resnick (Eds.), *Constructionism in practice: Designing, thinking, and learning in a digital world*. Mahwah, NJ: Lawrence Erlbaum Associates.

LX

About the Author

Alan Epstein has been developing software since 1975, and most recently has been studying ways that computer technology can be effectively used in schools. He is most interested in robots, especially for children.

Alan Epstein
alan@metasoft.com



FOR BEGINNERS

Tips for Making Multimedia with MicroWorlds 2.0

by **GARY S. STAGER**

If I could only have one piece of software in a student laptop, classroom computer, or computer lab, it would be MicroWorlds. This program affords teachers the opportunity to meet curricular needs while respecting a student's right to a powerful open-ended software environment in which they can pursue personal interests while constructing their own knowledge. MicroWorlds 2.0 is now available for the Windows 95 system. It has all of the multimedia features (digitized sound, CD audio, video, transitions, and enhanced graphic tools) as its

MicroWorlds is a rich computational medium based on 30 years of Logo research. It provides students with an unparalleled intellectual medium and vehicle for self-expression.

Macintosh predecessor plus extra processing power. You can even publish simple projects on the World Wide Web.

MicroWorlds 2.0 may certainly be used to create the book reports and simple multimedia presentations that are so common in other hypermedia-authoring packages, yet it is far richer as a computational medium based on 30 years of Logo research. Micro-

Worlds 2.0 provides students with an unparalleled intellectual laboratory and vehicle for self-expression. Five-year-olds and their teachers can both explore sophisticated learning adventures with the same tool. In this column I have included several tips and hints for getting started with the multimedia features of the new MicroWorlds.

Tips for Understanding and Using MicroWorlds 2.0's Multimedia Features

The graphical user interface has been improved dramatically. The top eight buttons on the tool bar are for creating MicroWorlds objects, and the bottom six are for editing existing objects.

The three icons in the Command Center represent (from top to bottom) the command, turtle shape, and paint modes.

Turtle costumes (shapes) may now be any size at all. Double-click on a shape in the Shapes Center and use the plus and minus buttons to change the drawing area for a particular costume. Drag the frame in the preview window (top right) to move about in the editing window.

The most powerful new tool is the Eye Tool. It will edit any object on which it is clicked. You no longer need to use a specific tool to edit each type of object. While selected, the Eye Tool will also show invisible objects and hidden turtles.

Each object created with the the top eight tools on the tool bar, with

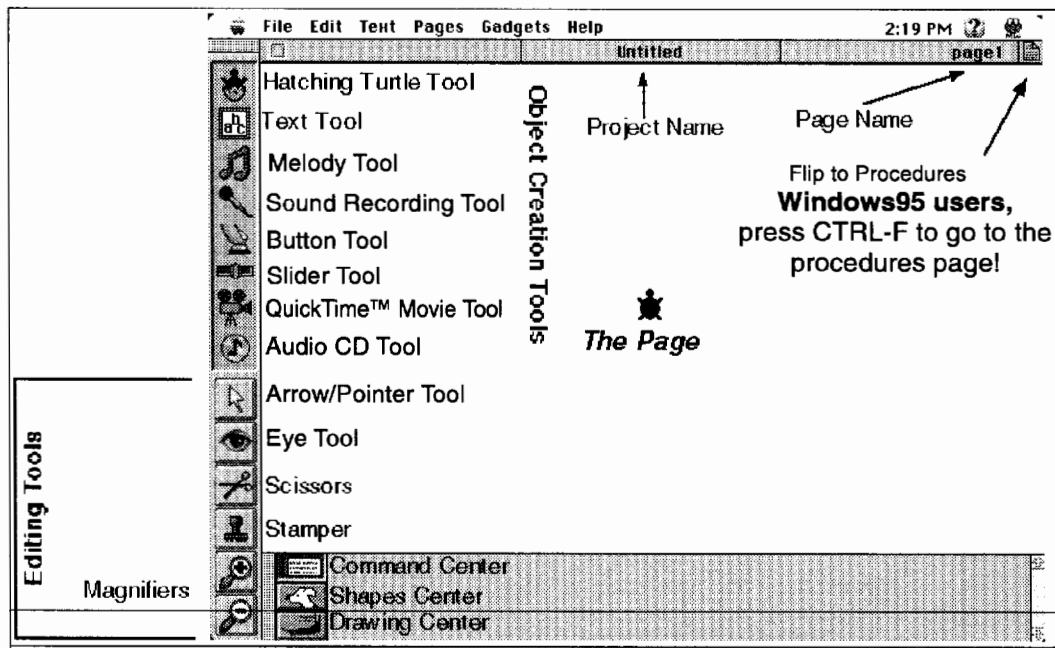
the exception of the text box and hatching turtle tools, can be executed merely by typing the name you gave them in the Command Center's procedure turtle and button instruction. For example, to play a video clip, melody, recorded sound, or CD-audio clip named *weasel*, type *weasel*. The object name behaves as if it were a defined procedure.

Typing the name of a text box reports its contents, so be sure to precede the name with a command to "catch" the output from the text box. For example, typing the name of a turtle, followed by a comma—as in *Weasel, fd 50*—tells the turtle named Weasel to go forward 50 turtle steps.

Check out the transitions in the Pages menu, but don't overuse them. MicroWorlds 2.0 allows you to set the screen transition for each page. The transition will appear when you go to that page.

If you don't like the standard button shape that MicroWorlds offers, then make your own! Any graphic imported into MicroWorlds and copied into the clipboard may be pasted into the Shapes Center as a turtle shape. Just put that new costume on a turtle and give it the instruction(s) you want it to run when it is clicked on. Voila! A button you can be proud of!

Melodies, recorded sounds, and CD-clip icons do not need to show unless you want them to behave like ugly buttons. Click the Eye Tool on them and set them to Invisible and they will still function whenever they are called on



to act. You may edit invisible objects by clicking on the Eye Tool. Every object on a page becomes temporarily visible.

You no longer need to use SNAPTEXT and UNSNAPTEXT to make a text box transparent. In version 2.0 you just click the Eye Tool on the text box and then on the Transparent button.

The Announce and Question commands now allow you to specify their screen position and size. This lets you move the Question and Answer dialog boxes to convenient screen locations. Just check how to use the SET primitive in the online Help. (Set lets you control all sorts of other things under program control, too.)

The File-Import menu item allows you to import text, sounds, graphics, video, and parts of MicroWorlds projects from other files. Try importing a stunning background picture or video clip you have saved on your hard drive.

Import an audio track from a CD and save it to your hard disk as a QuickTime file. This allows you to use multiple CD clips in one project without swapping CDs. Import sound from the File menu, give the new file a name, and then click on options to edit the track. Be sure to move any sound or video files with the MicroWorlds projects so they work properly.

Because most Web sites publish digital video in QuickTime format, you may need to convert these files to AVI if you are using Windows 95. You can download a conversion tool from

<http://www.intel.com/pc-supp/multimed/indeo/smartvid.htm>

or search for another. Mac users may download my free instructions on how to edit QuickTime video from

<http://moon.pepperdine.edu/~gstager/MoviePlayer.html>

In the Windows 95 MicroWorlds 2.0, you can explore the SAVEHTML command to learn how to create simple Web pages from your projects. Mac users need to download the HTML tools from <http://www.lcsi.ca>. Windows 95 users can actually share MicroWorlds projects on the Web by using the new MicroWorlds plug-in available at the LCSi Web site (above).

Explore the new online help and project book to find wonderful new surprises in MicroWorlds 2.0!

New Tools

Here are a few handy procedures you can add to your Logo toolbox. Playvideo takes the name of a QuickTime (Mac)/AVI (Win 95) movie as input, shows

the movie, plays it, and then makes it disappear. This creates the dramatic effect of a movie appearing and playing automatically.

```
Playvideo
"movie
```

```
to playvideo
:movie
set :movie
"visible? "true
run :movie
set :movie
"visible?
"false
end
```

Turtles and texts are procedures that report the names of every turtle or text box on the current page. You can type SHOW TURTLES to see a list of the turtles on that page, ASK BF TURTLES [fd 50] to have all but the first turtle in the list go forward, or use ZAP TURTLES to get rid of every turtle on the current page.

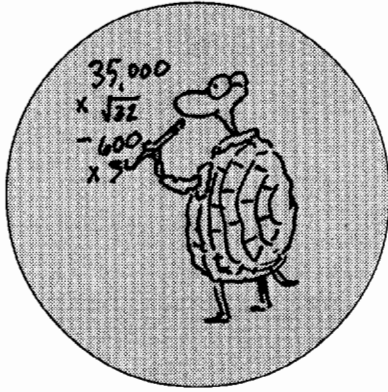
```
to turtles
output get first pagelist
"turtles
end
to texts
output get first pagelist
"texts
end
to zap :list
if empty? :list [stop]
remove first :list
zap bf :list
end
```

LX

About the Author

Gary Stager is the Editor of *Logo Exchange* and an Adjunct Professor at the Pepperdine University Graduate School of Education and Psychology.

Gary Stager
gstager@pepperdine.edu
<http://moon.pepperdine.edu/~gstager/home.html>



The Koch Snowflake Curve and Recursion

by **DON E. RYOTI**

Students find the Koch Snowflake Curve to be an interesting geometric shape. The process to generate comparable curves is straightforward. The Koch Snowflake Curve is the limit of a sequence of curves. Students can investigate the sequences of perimeters and areas for this sequence. From a mathematical standpoint, the curve is of interest because the sequence of perimeters diverges with no bound and the sequence of areas converges. This curve is also one of the simplest examples that illustrate self-similarity; parts of each "segment" are similar to the whole "segment." From a computer science standpoint, the procedure is a relatively simple example with multiple recursive calls. Students are motivated by how easy it is to create their own unique snowflakes, as we will see.

First, have students write the code for the drawing in Figure 1. Let the length of each segment be 30 turtle steps, and imagine that the two slanted segments are two sides of an equilateral triangle. Let the turtle start at A and end at B.



Figure 1

```
TO PATH
FORWARD 30
LEFT 60
FORWARD 30
RIGHT 120
FORWARD 30
LEFT 60
FORWARD 30
END
```

Two questions should be asked and answered:

1. What is the shortest distance from the start position to the end position?
Answer: 90

2. What is the distance that the turtle has moved?
Answer: 120

How could the procedure be changed so that the user can input the "shortest distance from the initial position to the final position?" The name "GEN" is an abbreviation for the word *generator*.

```
TO GEN :LEN
FORWARD :LEN/3
LEFT 60
FORWARD :LEN/3
RIGHT 120
FORWARD :LEN/3
LEFT 60
FORWARD :LEN/3
END
```

Second, have students write the code for the drawing in Figure 2. Each segment of Figure 1 is replaced with the pattern of the procedure TO PATH. Each of the four identical sections of Figure 2 is similar to Figure 1.

```
TO NEW :LEN
GEN :LEN/3
LEFT 60
GEN :LEN/3
RIGHT 120
```

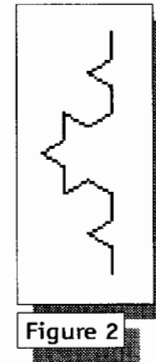


Figure 2

```
GEN :LEN/3
LEFT 60
GEN :LEN/3
END
```

Of course, a variety of other student responses are possible. A correct response could be entirely in terms of FORWARD, LEFT, and RIGHT;

these students are missing the modularity that is possible. An incorrect response would be to divide by 9 rather than 3; such students are having difficulties with variables. An incorrect response would be to repeat the GEN procedure four times; students omit turns and put in various types of turns. Examining these suggestions with students is useful to all. A repetition is involved, but it is more than what a REPEAT statement can do.

Some students who have studied recursion will realize that some recursive procedure will produce Figures 1 and 2 and the next ones in this sequence.

Third, present the concept of self-similarity. At the next stage, each of the segments would be replaced with a path that is similar to Figure 1. The segments of the new path are always one third of the segments in the previous path; the angles of the new path are always the same size as the angles in the previous path.

The desired procedure must include variables for the length and for the stage. Students typically understand

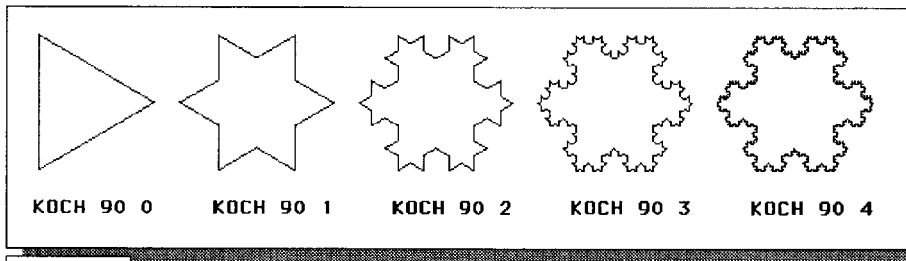


Figure 3

the need for recursion, that the value for stage will change by a decrease of 1, and that the length of the segments will change by a factor of one-third. But they wonder how to put it all together.

Notes:

- KSFSIDE is for Koch Snowflake Side.
- :LEN represents the (shortest) distance from the start point to the end point.
- :STAGE represents the number of levels of complexity.

```
TO KFSIDE :LEN :STAGE
IF :STAGE = 0 THEN FORWARD
:LEN STOP
KSFSIDE (:LEN/3) (:STAGE - 1)
LEFT 60
KSFSIDE (:LEN/3) (:STAGE - 1)
RIGHT 120
KSFSIDE (:LEN/3) (:STAGE - 1)
LEFT 60
KSFSIDE (:LEN/3) (:STAGE - 1)
END
```

Figure 1 corresponds with :STAGE = 1. What happens when :STAGE = 0? The answer is that the path of the turtle movements equals the shortest distance from the start point to the end point. A difficult-to-spot bug occurs when the person types (:STAGE-1) rather than (:STAGE - 1); in the former, there is no space between the "-" and 1.

Compare the procedures GEN and KSFSIDE. Each move (FORWARD) in GEN is replaced with directions to produce a figure similar to what is drawn with GEN. Each turn (RIGHT or LEFT) in GEN is kept the same.

Finally, we can utilize procedure KSFSIDE to form a closed figure. The procedure KOCH will generate the sequence of curves whose limit is the Koch Snowflake Curve. See Figure 3.

```
TO KOCH :LEN :STAGE
REPEAT 3 [KSFSIDE :LEN :STAGE
RIGHT 120]
```

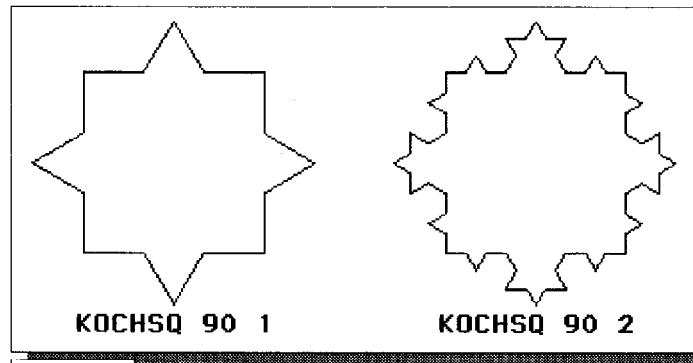


Figure 4

END

Some students are likely to think that it is necessary to use the triangle concept (REPEAT 3 with turn 120) because part of an equilateral triangle is drawn in the original path. Present the following procedures and curves. See Figures 4 and 5.

Both figures are drawn with :STAGE = 1 and :STAGE = 2.

```
TO KOCHSQ
:LEN :STAGE
REPEAT 4
[KSFSIDE
:LEN :STAGE
RIGHT 90]
```

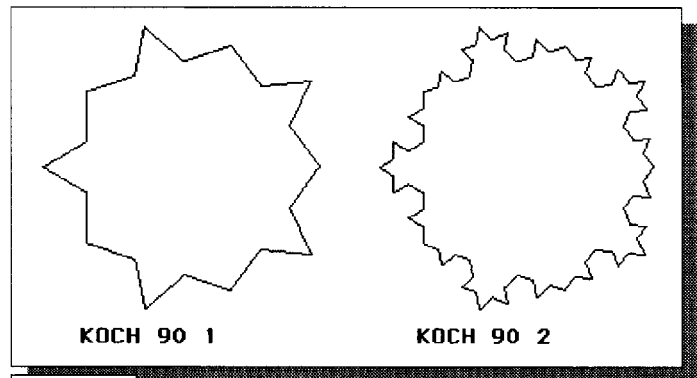


Figure 5

```
END
TO KOCHPENT :LEN :STAGE
REPEAT 5 [KSFSIDE :LEN :STAGE
RIGHT 72]
END
```

Now students can develop their own fractals. The major decision concerns how to use the generator. The segment (from start to end) does *not* have to be divided into thirds; it could be fourths, fifths, or whatever. The generator does *not* have to include two sides of a triangle; it could include three sides of a square or whatever. The generator's protrusion does *not* have to be positioned at the middle third; it could be the first third, third fourth, or whatever. The generator's

protrusion does *not* have to be a single thing: It could be several things; it also could be positioned to the left or the right. Finally, the closed figure does *not* have to be triangular; it

might well be pentagonal or whatever.

One simple example not only emphasizes the concept of self-similarity but also illustrates some variations mentioned in the previous paragraph. Contrast GEN4A and GEN4B.

There is only a slight variation in the code; the resulting figures are identical. See Figure 6.

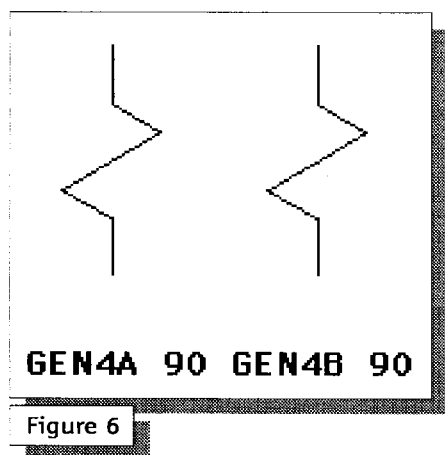


Figure 6

```
TO GEN4A :LENTO GEN4B :LEN
FORWARD :LEN/4 FORWARD :LEN/
4
LEFT 60 LEFT 60
FORWARD :LEN/4 FORWARD :LEN/
4
RIGHT 120 RIGHT 120
FORWARD :LEN/4 FORWARD :LEN/
2
FORWARD :LEN/4
LEFT 60 LEFT 60
FORWARD :LEN/4 FORWARD :LEN/
4
RIGHT 60 RIGHT 60
FORWARD :LEN/4 FORWARD :LEN/
4
ENDEND
```

When these procedures are transformed into the recursive procedures, the 4A procedure has six recursive calls and the 4B procedure has only five recursive calls. The figures at the next stage will be different. See Figure 7. The procedures to draw these are not included.

What is the stack for this procedure? (See Ryoti, 1996.) It is instructive to study the stack to better understand the concept of recursion.

KSFSIDE 90 0 creates a stack with one level. A single line segment of length 90 is drawn.

```
1 IF 0 = 0 THEN FORWARD 90
STOP 1
```

```
2 KSFSIDE 90/3 (0 - 1) *
3 LEFT 60 *
4 KSFSIDE 90/3 (0 - 1) *
5 RIGHT 120 *
6 KSFSIDE 90/3 (0 - 1) *
7 LEFT 60 *
8 KSFSIDE 90/3 (0 - 1) *
9 END 2
```

KSFSIDE 90 1 creates a stack with five levels. The result is Figure 1.

```
Level 5:
1 IF 0 = 0 THEN FORWARD 30
STOP 15
2 KSFSIDE 30/3 (0 - 1) *
3 LEFT 60 *
4 KSFSIDE 30/3 (0 - 1) *
5 RIGHT 120 *
6 KSFSIDE 30/3 (0 - 1) *
7 LEFT 60 *
8 KSFSIDE 30/3 (0 - 1) *
9 END 6
```

```
Level 4:
1 IF 0 = 0 THEN FORWARD 30
STOP 11
2 KSFSIDE 30/3 (0 - 1) *
3 LEFT 60 *
4 KSFSIDE 30/3 (0 - 1) *
5 RIGHT 120 *
6 KSFSIDE 30/3 (0 - 1) *
7 LEFT 60 *
8 KSFSIDE 30/3 (0 - 1) *
9 END 12
```

```
Level 3:
1 IF 0 = 0 THEN FORWARD 30
STOP 7
2 KSFSIDE 30/3 (0 - 1) *
3 LEFT 60 *
4 KSFSIDE 30/3 (0 - 1) *
5 RIGHT 120 *
```

```
6 KSFSIDE 30/3 (0 - 1) *
7 LEFT 60 *
8 KSFSIDE 30/3 (0 - 1) *
9 END 8
```

```
Level 2:
1 IF 0 = 0 THEN FORWARD 30
STOP 3
2 KSFSIDE 30/3 (0 - 1) *
3 LEFT 60 *
4 KSFSIDE 30/3 (0 - 1) *
5 RIGHT 120 *
6 KSFSIDE 30/3 (0 - 1) *
7 LEFT 60 *
8 KSFSIDE 30/3 (0 - 1) *
9 END 4
```

```
Level 1:
1 IF 1 = 0 THEN FORWARD 90
STOP 1
2 KSFSIDE 90/3 (1 - 1) 2
(creates level 2)
3 LEFT 60 5
4 KSFSIDE 90/3 (1 - 1) 6
(creates level 3)
5 RIGHT 120 9
6 KSFSIDE 90/3 (1 - 1) 10
(creates level 4)
7 LEFT 60 13
8 KSFSIDE 90/3 (1 - 1) 14
(creates level 5)
9 END 17
```

KSFSIDE 90 2 creates a stack with 21 levels; 77 lines are executed. The result is Figure 2. This stack is *not* included here (electronic or printed versions will be provided by the author on request).

With the stack and numbering, a variety of interesting numeric pattern questions can be asked:

1. Knowing the number of stages, how many levels are created?
2. Knowing the number of stages, how many lines are executed?
3. Knowing the number of stages, the turtle moves in which levels?

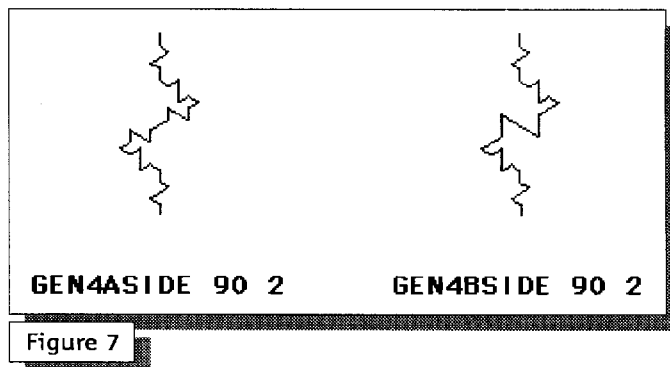


Figure 7

4. Knowing the number of stages, in which levels does the turtle turn?
5. For each stage, when a new level is created how many lines are on the stack?

References

- Abelson, H., & diSessa, A. (1980). *Turtle geometry*. Cambridge, MA: MIT Press.
- Horney, M. (1990). Fractals I: Making recursion visible. *Logo Exchange*, 9(1) 23-29.

Peitgen, H., Jurgens, H., Saupe, D., Maletsky, E., Perciante, T., & Yunker, L. (1991). *Fractals for the classroom: Strategic activities, volume one*. New York: Springer-Verlag.

Ryoti, D. (1996). Recursion and the stack. *Logo Exchange*, 15 (1), 27-30.



About the Author

Don E. Ryoti is a professor of mathematical sciences at Eastern Kentucky University. He has introduced Logo to

fourth through eighth graders and regularly teaches a 3-hour CSC Programming in Logo course, which is a requirement for K-4 and 5-8 mathematics emphasis education majors.

Don E. Ryoti
Wallace Building 313,
Eastern Kentucky University
Richmond, KY 40475-3133
matryoti@acs.eku.edu.

Logo Online Logo Online Logo Online Logo Online Logo Online

An impressive collection of Logo materials is available on the Internet. From time to time, *Logo Exchange* will point you in their direction. Below is a starter list for you to explore.

The Logo Foundation and the Global Schoolnet Foundation sponsor the Logo-I. Listserv. Subscribe to this listserv and join a global community of Logo users via email. No fancy browsers or high-speed data lines are required. To subscribe, send an email message to majordomo@gsn.org and put the words *subscribe logo-i* in the body of the message. All of the postings to the listserv are archived on the Web as well at

<http://archives.gsn.org/logo-i>

Speaking of the Logo Foundation, its Web site at

<http://el.www.media.mit.edu/groups/logo-foundation/index.html>

has links to numerous other Logo-related Web sites and a catalog of Logo materials available for purchase.

The massively parallel and wonderful StarLogo 2.0 for the Macintosh is available for download at

<http://www.media.mit.edu/~starlogo/download>

Check out the StarLogo Web site at

<http://www.media.mit.edu/~starlogo>

You will find updated documentation, more sample projects, user community

information, pointers to related work, and a comprehensive list of all new features. To join a list of StarLogo users, point your browser to

<http://www.media.mit.edu/~starlogo/community/>

Those of you who are interested in *Memo-Watchin*, *Turtle Math*, and *My Make Believe Castle/Titanium* his software will find LCSI product demo, support, and information available at

<http://www.lcsi.ca>

Terrapin Logo and PC Logo product information and other resources are available at

<http://www.terrapinlogo.com>

Windows users may be interested in MSWLogo 5.3, which is available for free download from

<http://www.softbooks.com>

Brian Harvey's Web page contains many papers written by Brian, information on his books, and free downloadable versions of Berkeley Logo. Go to

<http://http.cs.berkeley.edu/~bb/index.html>

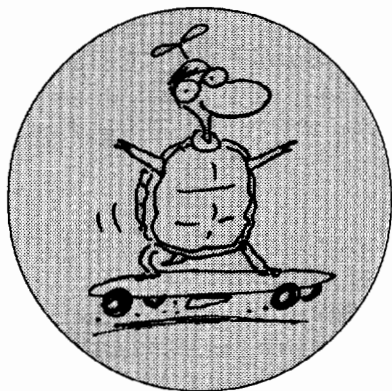
Berkeley Logo is available for Macintosh, Windows and Unix computers.

Gary Stager's Web site—

<http://www.pupperline.edu/~gstager/home.html>

—has a large collection of resources and links of interest to Logo-using educators and learners.





IN THEIR OWN WORDS

Stop the Internet, I Want to Get Off

by **BRIAN HARVEY**

About a month ago, Gary Stager asked if I was planning to attend NECC this summer. I said "No." I was discouraged by the extent to which the paper titles in the advance program focused on the World Wide Web. Shortly thereafter, Gary became editor of *Logo Exchange* and asked me to write an article about it.

Here are a few paper titles I've taken at random from the program:

- Internet Tapestries: Weaving a Web into the K-12 Curriculum
- ArtsEdNet: Bringing Art into the Classroom with the Web
- Virtual Museums: Constructing Meaning on the World Wide Web
- CyberSchool: Delivering High School Credit Classes over the Internet
- Evaluating World Wide Web Information: How Can We Trust What We See?

At least the last one doesn't sound wide-eyed.

I feel funny about writing this for two reasons. First, because I did not attend NECC, I'm only guessing about what I would have heard at these sessions. And second, I'm more accustomed to being the one who's enthusiastic about computer technology and trying to convince the skeptics.

I'm not against networks. I've been using the net since the early 1970s, back when it was the Arpanet rather than the Internet. Throughout my adult life, I've moved back and forth

between Massachusetts and California, and I use e-mail to keep in touch with friends on the other coast.

But I really don't like the World Wide Web. The trouble is that it uses a publishing metaphor rather than a conversational metaphor: Someone puts information on a Web page, and everyone else reads it. I'm much happier with the traditional, conversational use of the network—through e-mail and newsgroups (bulletin boards)—in which everyone talks to everyone else. I know that you can use your Web browser to get to e-mail and newsgroups, and I also know that it's the Web model that gets most people excited. It's a funny irony. Fundamentally, I don't like the Web because I think the publication model is elitist, but I'm also aware that it's equally elitist to think that I know how best to use the Internet and everyone else is wrong.

And, by the way, I'm not suggesting that the Web be dismantled. There are wonderful things out there. It's great that everyone can watch the Mars rover almost in real time. And I admit that I've bought a couple of import CDs through the Web. And I have my own Web page, too (no animations, but plenty of pictures).

The turning point for me came last summer. I was teaching a programming class for high school students as part of a summer program sponsored by the School of Education here at Berkeley. (The course used Scheme, a sort of Logo cousin.) One kid, a faster learner

than the others, complained of boredom, so I showed him one of my favorite programming projects, the Simplex lock problem. This concerns a kind of combination lock, different from most locks in that a combination can include pressing more than one button at a time. The problem is to work out the number of possible combinations.

Five minutes later, Nathaniel called me over to his workstation. On his screen was the Simplex Lock FAQ! (For those who've been on Jupiter recently, a FAQ is a document with "Frequently Asked Questions" and their answers.) I had no idea there was such a thing; he found it doing a Web search.

I suppose it's obvious that this wasn't the outcome I had in mind. The point wasn't merely to find out the answer, but to work out *how to compute* the answer. It was obvious to Nathaniel, too; showing me the FAQ instead of writing a program was mostly a joke. But the fact remains that he never did figure out the solution himself.

How much educational Web use has this same flavor? The worst thing about educational computing, starting with the development of machine-graded multiple-choice tests, has always been its tendency to make education more about knowing facts and less about solving problems.

I'm more sympathetic with those who want kids to publish their own Web pages. The *best* thing about educational computing has always been the empowerment of kids who discover

a new medium in which to express their ideas. Logo, of course, is part of that tradition. But even the creative use of the Web is problematic for me for two reasons.

First, the Web is like television: Its focus is on pictures, action, and animation. E-mail and newsgroups are almost entirely based on plain text. You can post a message to a newsgroup in seconds, and the message simply says what you want to say. But creating a Web page requires much more planning and implementation, especially if you want people to read and admire your page. More effort goes into the packaging than into the content. And Web publishing has quickly become a popularity contest in which the object is to be mentioned in someone's list of "hot" sites. Apart from blinking text, the thing I find most obnoxious on the Web is the convention of displaying how many people have read the page so far.

Second, although hypertext (links to other pages) can be extremely helpful, too many pages—and I think especially too many kids' pages—are entirely, or almost entirely, filled with links to other pages. There is an interesting mathematical reason why we should expect that: If N people are putting actual content pages on the Web, then there are 2^N possible pages of selected links to those N useful pages! The situation is just like that in the investment world, where more mutual funds are now available than there are underlying stocks. If there are N actual productive companies, then there are 2^N possible mutual funds. (And it's worth noting that all the people who run those mutual funds are, in the technical sense, unproductive; similarly, the people who put together lists of favorite Web pages aren't learning much.)

I don't want to be too simplistic about this. In its early days, Logo itself was the graphically exciting (drawing polygons!) alternative to text-limited programming languages in which one could do arithmetic speedily. More recently, the move from Logo to LogoWriter to MicroWorlds has been

a shift from a programming emphasis to an immediate-mode visual emphasis. And some authors of animated Web pages are motivated to learn to program in Java, an arena in which they do real problem solving—as in the Logo tradition. Perhaps we should get on that bandwagon: Declare Logo obsolete, and work on adding better heterogeneous list-processing primitives to Java.

But whether the tool is MicroWorlds or PageMill, it's really easy today to use the computer to paste clip art rather than do any actual computing. Gary suggests the metaphor of making a collage as opposed to creating one's own painting. That does capture a large part of the problem, but I think an even better thing to remember is the old MIT Logo Lab slogan about doing mathematics rather than learning about it. We used to think that we had important things to teach about algorithms, about recursion, about Total Turtle Trips, and so on. (Of course, when I say "computing" and "mathematics," I don't mean just numbers or even mainly numbers. You can also do math about words and sentences or even pictures.)

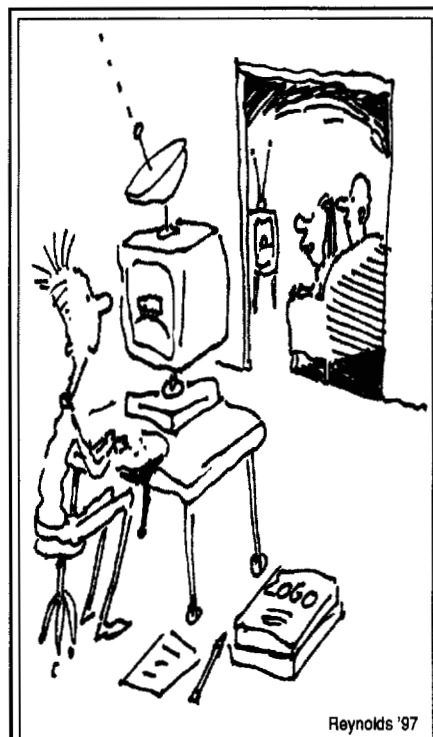
I'd probably like many of those Web-based curricula if I knew the details. Still, it worries me that technology in education seems to be so faddish lately. Last year the big idea was multimedia; this year it's the Web. Will it be something different next year? Are we focusing so much on the tools that we don't have time to think about the purposes?



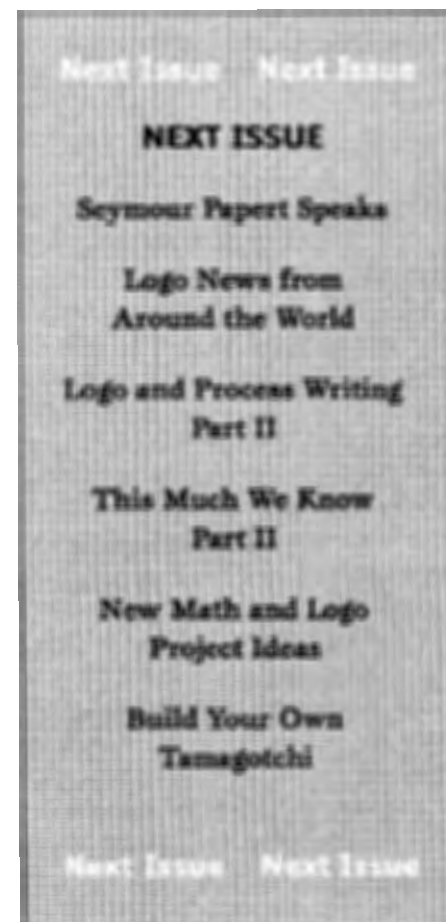
About the Author

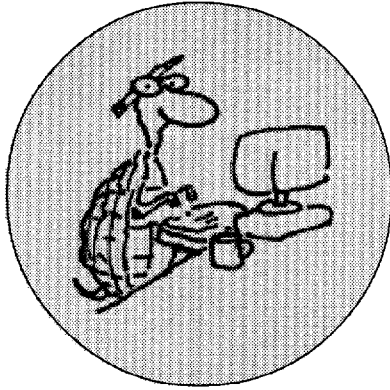
Dr. Brian Harvey is a lecturer in the Computer Science Division of the University of California, Berkeley. He is a veteran Logo developer and author of the three-book series, *Computer Science Logo Style*, and the developer of the UCB Logo.

bh@CS.Berkeley.EDU
UCB Logo may be downloaded
from <http://http.cs.berkeley.edu/~bh/index.html>



Junior controls his LEGO Logo marsrover while the folks watch TV.





Logo and Thinking Writing: Part I

by **LESLIE F. THYBERG**

Introduction

"Writing to learn is different from writing to show that you have learned what the teacher or the text has set for you to learn" (Countryman, p. 88). Despite the wonderful marriage of Logo and its word-processing capabilities, most applications of LogoWriter and MicroWorlds assignments at best show what the student has learned—or what we as teachers want and expect them to learn. They do not reveal whether the students has engaged in any real evaluation of his or her own learning, writing, and thinking. This may be one reason why Logo has not made the breakthroughs we expected of it. This probably is not the sole reason, but I would like to propose a different approach to looking at writing in which it is used in itself as a learning process.

In 1984, Dan Watt issued a clarion call when he put forth the notion of learning Logo in contrast to literacy and communication in a broader cultural context. It was his suggestion that because our culture is oriented toward an expectation of reading, children are, in effect, immersed in and surrounded by an environment that supports this expectation. I have thought long and hard about the implications of this. I have concluded that we are increasingly becoming a computer-using culture. But as this becomes the case, the questions about what a supportive environment is and how we immerse our children in the uses of technology become even more significant. We cannot take for granted that computers,

or even Logo, will be "good" for them. What may be more critical for us to examine are the methodology and the underlying philosophies that are being used for teaching.

My background is as that of a reading specialist and college-level language arts professor. In my courses, I make it a point to examine our belief systems and teaching perspectives. Our belief systems are a driving force in teaching—influencing everything from the instructional goals we set to the materials we select, the environment we establish, and the behaviors and practices we reward. In recent decades, the dominant approach to teaching has been from a skills perspective. Only recently has the skills perspective undergone any real challenge. A whole language perspective is making its way into the mainstream. Although this is generally a welcome trend, there are far too many instances of "overprocessing of the process approach" (or putting holes in the whole). This has to do with belief and perspective. "Materials in the hands of a teacher who holds a skills model are skills materials. Materials in the hands of a teacher who holds a whole language model are whole language materials" (Paul Crowley, International Reading Association, May 1989).

What bearing does this have on Logo? Specifically, I view Logo as a composition process and not as a separate curricular discipline.

Logo is more than a fancy way of teaching graphics or geometry. It involves the student in composing—in planning, in dealing with an audience, and in revising the program to deal with the inevitable bugs. A new breed of writers—writers of programs—will enter the classroom with this kind of composing experience. To the extent teachers can sense the similarity between these two modes of composing, they will have a powerful ally for the teaching of writing. (Newkirk, 1985, p. 40)

My approach is based on the proposition that there are two modes of composing: written composition and Logo composition, although such an approach is not limited to Logo. The process approach can be applied to any version of Logo or any Logo-like process (such as Peter Skillen's work with HyperCard in his "Thinkingland" project). Most educators "take as a given that depth and clarity of thinking enhance the quality of writing. What may not be so readily apparent is that writing is a learning tool for heightening and refining thinking" (Olson, 1992, p. 3).

Briefly, my approach draws on the notions of the process approach to writing. Donald Graves is one of the widely recognized "giants" in this domain. His model is one that makes writing "real" to children. Although there are variations, a certain form is generally followed in teaching the writ-

ing process. Writing is generally viewed as a function of three separate stages. These stages, in turn, have their own unique subsets. The three phases are *prewriting*, *writing*, and *postwriting*. The following is a brief discourse on the writing process. My proposition is to overlay the word *Logo* throughout and thus adopt a process-oriented approach to learning *Logo* and learning with *Logo*.

The Writing Process

The prewriting stage involves talking, reading, and sharing ideas that serve as catalysts to activate writing. The prewriting process creates a classroom atmosphere in which students dare to write. Before writing is undertaken, it needs to be justified as an inescapable product of the experience that has preceded it. In other words, students must first have a sense of purpose in writing.

Developing student skills at the precomposing stage includes emphasizing the notion of a sense of purpose to the writing activity itself. This can come from teacher-student conferences, maintenance of a writing folder or a journal, daily uninterrupted and sustained silent writing, having the teacher serve as a role model (by the writing that he or she does with and for the students), and the quality and character of the motivational activities that precede the composition experience.

The second stage is the actual writing process. Students are encouraged to simply start writing. They are told that words can be changed later. Gebhardt (1977, p. 676) refers to this phase as a process of "generating materials, drafting and revising drafts to make them more effective."

The third and final stage is that of postwriting or rewriting. Here the teacher challenges children to become effective critics of their own work. Children "must become actively involved in evaluating their own writing. . . . [They] should be helped to assume the responsibility for analyzing their own composition and diagnosing their

own writing problems if improvement is to take place" (Sager, 1977, p. 760). In this system, students are moved toward independence; the emphasis is process over and above product, particularly with respect to editing. Wuertenberg plans a series of writing experiences that are intended to move the student toward independence—that is, toward the time when he will serve as his own "editor" (Wuertenberg, 1977). Student editors edit their own and their classmates' writing. Specialized student editing committees form, depending on what particular skill is being emphasized; thus, there may be spelling committees, punctuation committees, run-on-sentence committees, or even adjective committees. These "committees" are trained in workshops by the teacher. Papers are passed from committee to committee until the editing is done. Students, in effect, expedite the "paperwork load" while acquiring an invaluable skill.

In the postcomposing phase, the students' writing is evaluated and published. This stage is perhaps the most difficult. Teachers must avoid the temptation of demanding too many alterations or revisions.

These three stages are typically subdivided into five components: (1) topic choice and (2) rehearsal are part of the prewriting experience, (3) composing is the actual writing phase, and (4) reading and (5) revision are part of the postwriting experience.

This style of instruction requires an active role for both teachers and pupils. Because it is a large demand for a lone teacher to process students' work, this model is designed to empower students by teaching them the art of proofreading, polishing, and editing their own and each other's work. Research has shown that when the process model is well implemented, writing experiences lead the students to acquire the ability to be independent, self-correcting, and self-editing. They accurately perceive themselves as effective writers.

Teaching the Process Model to Students

How does one teach students to become process-oriented? Bransford and Stein's "IDEAL Problem Solver" model is a clear and simple approach that synthesizes many of my views on how to get students involved in cognitive processes (Bransford, 1984). This model can be integrated easily into any classroom organization. Its five components are:

- I Identify the problem
- D Define and represent the problem with precision
- E Explore possible strategies
- A Act on a strategy
- L Look at the effects

The five steps of the "IDEAL" acronym correlate directly with the five steps of the writing process (topic choice, rehearsal, composition, reading, and revision). As part of the training process, children can list the steps of the writing and IDEAL process and make posters (or some other visual representation) of the process to display in the classroom.

Implementing a process-oriented environment can progress through three phases. In the first phase, it is the teacher's responsibility to guide the student orally through each step. In the second phase, it is the teacher's responsibility to refer the student to the model, giving him or her the main responsibility of working through the steps. In the third phase, the student can use the model independently.

Teaching Logo Programming the Same Way

Logo programming can be taught in the same way that students are taught to develop and extend their writing skills. The ingredients for such a "Logo-process approach" are: topic choice, rehearsal, composing, reading, and revision.

Topic Choice

Both writing an essay and generating a Logo project begin with topic selec-

tion. Ideas can be provided or originate from a central location—say, a Turtle Learning Center or writing center. They may overlap. A topic may be embedded in a previous project. The child may choose to collaborate with a peer or select a topic from a brainstorming session or a list labeled “future topics” that he or she keeps tucked away in a folder or writing notebook. Like writing, the topic choice is most likely to be effected by the kind of precomputing experience in which the student engages.

Rehearsal

Rehearsal refers to the conscious or unconscious preparation that writers and programmers make for what follows. Daydreaming, sketching, working with manipulatives, and playing turtle are all part of the rehearsal process. Rehearsal can include what Dan and Molly Watt (1986, p. 19) describe as “playful exploration.” Messing about and experimenting are part of the process of creating. Brainstorming—or “mindstorming” (or whatever experience precedes the actual writing or programming—should be so compelling that the student wants to write to express him- or herself. Drill and kill exercises and convergent, fill-in-the-blank style tasks and worksheets are less likely to bring broad-reaching results.

Composing

Composing includes the selection of information and mechanics in order to assemble parts into a whole. The world of writing offers many techniques to help generate stories. Some writers engage in what Peter Elbow calls “free writing,” which is the equivalent of stream-of-consciousness writing. This is akin to what we might call the “antiplanner”—the improviser or bottom-up style programmer. Still others might follow an outline or a model of some previously written story, perhaps in the style of another author.

Reading

Reading the written text is part of the postwriting process. In the programming process, “reading” refers either to seeing one’s program run or stepping through the program to find errors or check the appropriateness of program content. Many students who work with Logo tend to get in over their heads very quickly. They stumble across surprises ranging from recursion to simple typographical errors; these are excellent opportunities for informal teaching.

Revision

Revision, literally “seeing again,” ranges from a simple adjustment to fix a small bug to adding, deleting, or reorganizing subprocedures. The National Assessment for Writing found that children have a tendency to make minimal revisions in their compositions unless they are given constructive guidance and effective feedback. What occurs in Logo is probably comparable. When a student is frustrated with a graphic design, he or she often clears the screen or begins a different project instead of working to solve the problem.

It is here that a teacher who knows the student’s needs and interests can best intercede. I use a similar debugging format to that described by Dan and Molly Watt in their excellent text on *Teaching with Logo*. Some of their suggestions include having the student: describe the problem, re-create the bug, reexperience the bug in another way (by playing turtle or computer or sketching the desired outcome graphically), make a procedure tree, or use metaphors.

Introducing the process approach takes a lot of time but is a worthwhile investment. It is a way of helping students understand that the *process* is as important as the *product*. It is a mechanism for getting them to think about their thinking. This, after all, is one of the powerful notions embedded in Logo (Papert, 1980).

Strategies for Applying the Model and Integrating Logo and Writing

In my former classroom environment, I had a Logo center and a writing center. Both had the process steps posted in plain view, and both were designed to stimulate the users. Allow me to describe a specific application of the IDEAL and process models.

Early in a school year I might begin an introduction to a creative writing class by reminding the students that they will be using IDEAL to write a group story for their class newspaper. I would ask them what the first step of IDEAL is and then pause to give them an opportunity to reflect. The child I would call on would (hopefully) explain the first step: identify the problem. “Our problem is not having a topic to write about,” I would say and then ask if anyone had ideas for topics. Various students would then be called on while suggestions were placed on the board. Students would ultimately divide themselves into teams to work on the topics in which they were most interested. As students would work in subgroups to further define their project goals, I would confer with students who needed greater focus or assistance. Just as Wuertenberg suggests, student editors could be appointed to edit their classmates’ writing. Specialized student editing committees could be formed, depending on what particular skill is being emphasized (e.g., spelling committees, style or content committees, or procedure committees or other Logo-related “consulting” groups).

Suppose one group of students decided to compose madlibs. I might ask them for an outline of their paragraph’s structure. Once this was done, we would brainstorm phrases, characters, resolutions, and so on until all of the students were satisfied that they had defined their topic enough to begin writing. Such an activity as this can be done in Logo and still be creative writing. I would remind students to use “act” and “look” steps while writing.

There would be a lot of discussion as the students would begin to explore their ideas at the keyboard. One student might suggest that their description of the characters wasn't clear enough. Another child might read through the introduction and make the point that they had jumped too quickly into the action and that they should go back and clarify some of their earlier ideas. This group is acting on strategies, trying out ideas, and looking at the results. Knowing this, I would be certain to see that further revising was taking place in the context of the next class session, class meeting, or special interest group. Certainly, as the year progresses it is in the students' best interest to have more opportunities for autonomy and decision making. Students who need more direct guidance can and should receive it from the teacher.

It is also useful to create basic guidelines or standards to help the students perform effectively and cooperatively in a process-oriented setting. One example is a rule that I developed some years ago. This is the "Three Before Me" rule, which requires children to ask for help from three peers before they ask a teacher. Two other rules I employ are: (1) You are responsible for your own thinking and behavior, and (2) you must be willing to help anyone who asks.

The outcome of following the IDEAL and process models goes beyond enhancing students' self-concepts and helping them become effective and responsible self-starters. It creates a whole culture that is self-supporting. Students have the opportunity to be both teacher and student. This helps students learn how to deal with people who think differently, acquire patience, learn to be respectful of others, see classmates in a new light, and develop self-reliance, among other benefits.

With all of this in place to frame the paradigm, let's return to the notion of thinking writing. At the Tenth International Conference on Technology and Education, Dr. Papert stated, "Every lost opportunity to think is deadly."

Writing has the potential to free students of the assumption that the doing of Logo (or using it to do some other subject) is just a collection of right answers to questions or a way to please the teacher. Thinking writing—and this includes writing notes, lists, observations, and feelings in addition to term papers, lab reports, and essay questions—can and should expand students' thinking and learning and their ability to think about their learning. In other words, thinking about Logo as a means of communication frees us from the trappings of yet another content area to teach. Instead it allows us to use it when it is appropriate as a means of communication—whether applied to a math lesson, an art lesson, a social studies unit, or the language arts.

References

- Barell, John. (1991). *Teaching for thoughtfulness: Classroom strategies to enhance intellectual development*. White Plains, NY: Longman.
- Bransford, J., & Stein, B. S. (1984). *The IDEAL problem solver*. New York: Freeman.
- Cambourne, Brian. (1988). *The whole story: Natural learning and the acquisition of literacy in the classroom*. Richmond Hill, Ontario, Canada: Scholastic-TAB Publications.
- Carnegie Task Force on Teaching as a Profession. (1986, May 21). A nation prepared: Teacher for the 21st century. *Chronicle of Higher Education*, 1-45.
- Cooper, J. David. (1993). *Literacy: Helping children construct meaning*. Boston: Houghton Mifflin.
- Countryman, Joan. (1992). *Writing to learn mathematics: Strategies that work, K-12*. Portsmouth, NH: Heinemann.
- Elbow, Peter (1973). *Writing without teachers*. New York: Oxford University Press.
- Fulwiler, Toby (Ed). (1987). *The journal book*. Portsmouth, NH: Heinemann.
- Gebhardt, Richard. (1977, Sept.). The timely teetertotter: Balancing discipline and creativity in writing classes. *Language Arts*, 676.
- Goldenberg, E., & Feurzeig, W. (1987). *Exploring language with Logo*. Cambridge, MA: MIT Press.
- Goodman, Ken. (1986). *What's whole in whole language?* Portsmouth, NH: Heinemann.
- Graves, Donald. (1981). *Donald Graves in Australia—Children want to write* (R.D. Washe, Ed.). Australia: Primary English Teaching Association.
- Grinols, Anne Bradstreet. (1988). *Critical thinking: Reading and writing across the curriculum*. Belmont, CA: Wadsworth.
- Murray, D. (1968). *A writer teaches writing*. Boston: Houghton Mifflin.
- Newkirk, Thomas. (1985, Winter). Writing and programming: Two modes of composing. *Computers, Reading and Language Arts*, 2 (29), 40-43.
- Olson, Carol. (1992). *Thinking/writing: Fostering critical thinking through writing*. New York: Harper Collins.
- Papert, S. (1980). *Mindstorms*. New York: Basic Books.
- Routman, Regie. (1991). *Invitations: Changing as teachers and learners K-12*. Portsmouth, NH: Heinemann.
- Sager, C. (1977, Oct.). Improving the quality of written composition in the middle grades. *Language Arts*, 760.
- Soppeland, M. (1980). *Words*. Los Altos, CA: William Kaufmann.
- Thyberg, Leslie. (1988). *A Descriptive analysis of student behaviors and attitudes using Logo programming as a vehicle in the writing process*. Doctoral dissertation. University of Pittsburgh.
- Watt, Daniel. (1984). Creating Logo cultures. *Proceedings, Logo 84*. Cambridge, MA: MIT Press.
- Watt, D. (1983). *Learning with Logo*. New York: Byte Books/McGraw-Hill.
- Watt, D., & Watt, M. *Teaching with Logo: Building blocks for learning*. Menlo Park, CA: Addison-Wesley.
- Wuertenberg, Jacque. (1976). *Helping children write: From basic composition to creative writing*. Englewood, CO: Educational Consulting Associates.
- Wuertenberg, Jacque. (1977, Mar.). *Teaching writing*. Paper presented at the Holiday Inn, Rosemont, IL, for the Bensenville School District, under the sponsorship of Educational Consulting Associates, Englewood, CO.



About the Author

Dr. Leslie Thyberg is a teacher and university professor from Pittsburgh, Pennsylvania.

Leslie Thyberg

logoexchange@moon.pepperdine.edu



Teaching Recursion: A Cognitive Profile Perspective

by **PAM GIBBONS**

Introduction

The observations of this paper follow the results of an investigation into the impact of individual differences in cognitive-processing profiles on novice programmers' ability to understand and implement recursive repetition in a Logo programming environment (Gibbons, 1993). Luria's model of brain function (e.g., 1966, 1973, 1982)—particularly regarding the dichotomy of simultaneous and successive processing modes of the second functional unit of the brain—provided the investigation's theoretical basis. The study focused on the process of learners' construction of mental models of recursion. The implication of simultaneous and successive cognitive processing abilities in the determination of the nature and progress of mental model construction is now presented from the perspective of its import for future pedagogical decision making in the domain.

The Luria Model: Simultaneous and Successive Processing

The Luria model of brain function (1966, 1973, 1982) describes functional relationships between anatomically distinct regions of the brain and the interaction of these functional units with the physical and social environment. The model draws a clear distinction between the brain functions associated with routine, established activities and the conscious intellectual activity associated with solving problems not previously en-

countered (Crawford, 1986). Simultaneous and successive processing are described by Luria (1973, 1982) as two forms of "synthetic cerebral activity" occurring in the second functional unit of the brain; the model asserts that people are born with differences in such cognitive abilities, which are then influenced by an experiential and cultural knowledge base overlay.

Simultaneous processing refers to an ability to retain two or more items in mind while attempting to establish a relationship between them; it also can be described as the ability to perceive a number of elements as contributing to a whole (Biggs & Collis, 1982; Naglieri, 1989). Luria (1973) describes the ability to recognize a face or identify an object by touch as employing this form of processing. Biggs and Collis (1982) further distinguish the nature of simultaneous processing by noting that it is distinct from "reasoning" as defined by Jensen (1973), because, for example, the task of copying a figure in which one line is perpendicular to another does not involve reasoning, yet it is simultaneous. Luria (1973, 1982) describes simultaneous processing as leading to concept development and the establishment of a knowledge base of "matrices of community relationships."

In contrast, the mode of organization of perceptual stimuli in successive processing is sequential, characterized by temporally organized ordered series; it is also closely associated with imitative responses and motor planning.

Luria (1973) suggests that this type of processing leads to the development of "kinetic melodies": automatized operational routines that are not necessarily available for conscious consideration. Successive processing is associated with procedures required for subjects to provide a response, particularly responses that conform to some exact prescription: The memorization of lists of items or instructions, tunes, procedures, and protocols are likely to involve successive processing.

Biggs and Collis (1982) note that, as with simultaneous processing, successive processing is concerned with the organization and storage of data; however, in the case of the latter, perceived stimuli have no interrelationship other than that one precedes the other in time (just as B follows A in the alphabet for the reason that it is the order in which they are learned, there being no other relationship between them). Biggs and Collis (1982) further describe the nature of successive processing by noting that it is distinct from "memory" as defined by Jensen (1973): Successive processing is a much more specific concept than that described by the general notion of memory, which, according to Luria, results from both simultaneous and successive forms of processing.

Cognitive Processing Modes and Recursive Concepts

The concept of recursion demands a robust, surrogate mental model to sup-

port its selection as a repetition strategy and successful implementation (Gibbons, 1993). The nature of the recursive structure places high demands on simultaneous processing (i.e., the ability to keep several items in mind at one time and consider them as composing one entity). In its more complex forms, recursion will not be understood or implemented if a subject relies on a successive processing interpretation: The sequential impression of written nontail recursive procedures is contrary to the demands of the recognition of the simultaneous holding of data by several simultaneously existing recursive procedures. If learners are to develop sound systemic models that support semantic knowledge, then the use of simultaneous processing abilities should be fostered. A Lurian interpretation suggests that the invocation of simultaneous abilities will not only have an impact on the nature of cognitive outcomes (e.g., higher cognitive residue and systemic knowledge), but also will have a long-term incremental benefit for the abilities themselves: The more they are used, the better they will become and the more they will continue to be used—and continue to improve in the future.

As proposed by Shneiderman and Mayer (1979) and confirmed by the research of Oliver and Malone (1993), syntactic programming knowledge (linked to successive information processing) can be learned by exposition and practical application. In contrast, semantic knowledge (linked to simultaneous information processing) “is acquired through intellectually demanding and meaningful learning activities that cause the learner to assimilate and anchor information to existing knowledge and ideas” (Oliver & Malone, 1993, p. 552). The Luria model asserts that, in the absence of such activities, the simultaneous processing abilities demanded by higher order programming skills will atrophy as a result of failure to demand the invocation of such abilities early in a

learner’s educational experience. What begins as processing mode preference thus may eventually develop into an inadequacy in an information-processing-mode ability.

Pedagogical Implications

Other research in computer programming education (e.g., Lo & McNutt, 1991; Oliver, 1991; Oliver & Malone, 1993; Shneiderman & Mayer, 1979) supports the contention that teaching practice must reflect the dichotomy of simultaneous and successive processing demands. The consistent recognition of the divergent demands and the dual importance of semantic and syntactic knowledge parallels the manifestation of the two information-processing modes and confirms the inadequacy of the development of one ability at the expense of the other. Similarly, Norman (1993) encourages the development of reflective cognition (proposed to invoke simultaneous abilities) to at least the same level as experiential cognition (proposed to invoke successive abilities). Harvey (1985) recommends that multiple teaching strategies are needed to adequately address recursion: A Lurian interpretation suggests that the success of presenting both structural and evaluative explanations of recursion reflects the increased likelihood of one of the presented expositions matching the information-processing preference of all learners. It is further proposed to be indicative of the efficacy of the concerted activity of both processing modes in the ultimate development of a sound surrogate mental model.

Sequence:

Context, Forms, and Complexity

Much of the pedagogical conjecture associated with the concept of recursion relates to its initial instruction. Various propositions have been forwarded regarding the relative merits of introducing (a) recursion through graphic rather than textual examples, (b) iterative repetition before recursive repetition, (c) tail before nontail ex-

amples, or (d) recursive functions before recursive commands (e.g., Harvey, 1991; Levenick, 1990, 1991; Riordon, 1984; Troy & Early, 1992). It is proposed that both strengths and weaknesses will be associated with any single approach and that the relative efficacy of such will vary between individuals and contexts. The development of a universal model of recursion is likely to be facilitated, however, through exposure to a broad range of recursive types and contexts, and expository representations that consistently support the information-processing mode appropriate to a specific level of understanding.

Approaches that suggest the use of complex forms as a means of obviating the initial development of erroneous mental models, in moving too quickly and in demanding simultaneous information processing before the establishment of a sound knowledge base, are at risk of precluding the development of any mental model at all. The rapid adoption of systemic perspectives by novices is obstructed by a lack of incremental learnability (diSessa, 1985). The danger of dealing exclusively with tail recursive examples for any length of time, however, is reinforcement of erroneous “internal loop” models of recursion that, for high successive learners in particular, may be difficult to subsequently displace (Gibbons, 1993). Similarly, the introduction of tail or nontail recursive commands before recursive operations has the advantage of simplicity at the risk of failure to progress to a mental model supportive of powerful programming ideas (Harvey, 1991).

Although a turtle graphics environment promises to provide a relatively concrete representation of nontail recursion, it also poses unique difficulties to novices that can be explicated in terms of the Luria model. The visual nature of the effect of graphics procedures is linked to simultaneous processing abilities and a spatial or structural conceptualization of recursion. Textual (command) examples, however, pro-

duce screen displays that preserve a temporal or process representation of the procedure's execution: This helps in the evaluative interpretation of procedures that support novice programming activity. Textual examples may thus offer a level of cognitive support to novices in allowing the invocation of successive abilities until a sound knowledge base is established. Such a distinction reflects McDougall's (1992) observations regarding learners' perceptions of different types of recursion ("structure" and "process" microviews) and of a subject's preference for the use of a textual example when teaching a peer about embedded recursion despite the syntactic "overhead" of such an approach (McDougall, 1988). The Luria model implicates simultaneous and successive processing abilities in the development of spatial (structural) and temporal (process or evaluative) models, respectively.

Modes of Representation

The proposition of using language appropriate to the development of a structural model of recursion and invoking simultaneous processing abilities is asserted to be more adequately expressed in terms of the nomination of an appropriate form of representation with which to articulate such concepts. Verbal articulations are likely to limit instruction to sequential and evaluative interpretations associated with successive information processing. Walton (1983), however, emphasizes the diagrammatic representation of ideas to support simultaneous information processing; mental models researchers such as McGrath (1990), Booth (1989), and Carroll, Mack, and Kellogg (1988) similarly note the potential use of instructional metaphors to support mental model construction.

The simultaneous processing demands of recursion are thus proposed to be supported through diagrammatic representation and the use of appropriate instructional metaphors. This contention is consistent with the reported success of the use of noncom-

puter metaphoric introductions to recursive concepts (e.g., McDougall, 1985, 1988, 1989, 1990; Riordon, 1984). The notion of the use of domain-specific representations of mental models as cognitive supports for instruction is also suggested by Hancock (1988) and Bibby (1992). The exploitation of such representations are thus proposed as aids in expositions of recursion.

Conclusion

It is therefore asserted that, in making pedagogical decisions, conjecture about the sequence of introducing learners to different types and contexts of recursion is far less important than the insurance that learners are exposed to a variety of recursive forms and contexts. Such expositions also should be presented in a framework that both attends to the demands of syntax and provides a structural account of the semantics of the language. The contextualization of recursive procedures in terms that are consistent with a systemic understanding of the execution of any other type of procedure also reinforces the development of a universal and uniform model of recursion.

Moreover, the language used in any exposition of recursion must clearly support the development of nonlooping conceptualizations and ultimately facilitate the adoption of a structural and hierarchical interpretation of recursion. In further recognizing the importance of addressing simultaneous processing abilities in assisting learners to conceptualize recursion, the need to invoke nonverbal or linguistic representations is also stressed: To this end the inclusion of diagrammatic representations and instructional metaphors is recommended.

References

- Bibby, P. A. (1992). Mental models, instruction and internalization. In Y. Rogers, A. Rutherford, & P. A. Bibby (Eds.), *Models in the mind: Theory, perspective and application*, pp. 153-172. London: Academic Press.

- Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning: The solo taxonomy (structure of the observed learning outcome)*. London: Academic Press.
- Booth, P. (1989). *An introduction to human-computer interaction*. Hove, Sussex, UK: Lawrence Erlbaum Associates.
- Carroll, J. M., Mack, R. L., & Kellogg, W. A. (1988). Interface metaphors and user interface design. In M. Helander (Ed.), *Handbook of human-computer interaction*, pp. 67-85. Amsterdam: Elsevier.
- Crawford, K. P. (1986). *Simultaneous and successive processing, executive control and social experience: Individual differences in educational achievement and problem solving in mathematics*. Unpublished doctoral thesis. University of New England (Armidale, NSW, Australia).
- diSessa A. (1985). A principled design for an integrated computational environment. *Human-Computer Interaction*, 1, 1-47.
- Gibbons, P. E. (1993). *Recursion: An analysis of individual differences in Logo programming*. Unpublished doctoral thesis. University of Sydney.
- Hancock, C. (1988). Context and creation in the learning of computer programming. *For the Learning of Mathematics*, 8 (1), 18-24.
- Harvey, B. (1985). *Computer science Logo style. Volume I: Intermediate programming*. Cambridge, MA: MIT Press.
- Harvey, B. (1991). On recursion. *The Computing Teacher*, February, pp. 4-5.
- Jensen, A. R. (1973). *Educational differences*. London: Methuen.
- Levenick, J. R. (1990). Teaching recursion before iteration. *The Computing Teacher*, October, pp. 12-15.
- Levenick, J. R. (1991). On recursion: Jim Levenick replies. *The Computing Teacher*, February, pp. 4-5.
- Lo, B. W., & McNutt, L. (1991). The neglected dimension in programming courses. In *Proceedings of ACC '91—Mosaic*, pp. 387-404. Adelaide: Australian Computer Society.
- Luria, A. R. (1966). *The human brain and psychological processes*. Translated by B. Haigh. New York: Harper & Row.
- Luria, A. R. (1973). *The working brain: An introduction to neuropsychology*. New York: Basic Books.
- Luria, A. R. (1982). *Language and cognition*. New York: Wiley.

- McDougall, A. (1985). Teaching and learning about recursion (Abstract). In M. Palmgren (Ed.), *Logo85 Pre-Proceedings*. Cambridge, MA: MIT Press.
- McDougall, A. (1988). *Children, computers and Logo programming*. Unpublished doctoral thesis. Monash University (Caulfield, Australia).
- McDougall, A. (1989). Teaching about recursion in Logo: A review. In T. Dupé (Ed.), *Backup the future: Proceedings of the Australian Computers in Education Conference*, pp. 152–160. Canberra: Computer Education Group of the A.C.T.
- McDougall, A. (1990). Children, recursion and Logo programming: An investigation of Papert's conjecture about the variability of Piagetian stages in computer-rich cultures. In A. McDougall & C. Dowling (Eds.), *The Proceedings of the 5th World Conference on Computers in Education: WCCE/90*, pp. 415–418. Amsterdam: Elsevier.
- McDougall, A. (1992). Structure and process microviews: Partial understandings of recursion in Logo programming. In L. Nevile (Ed.), *Proceedings of Logo and Mathematics Education Conference LME5*, pp. 109–117. Hawthorn: Australian Council for Educational Research.
- McGrath, D. (1990). Using mental models to teach programming to beginners. *The Computing Teacher*, 17 (8), 11–14.
- Naglieri, J. A. (1989). A cognitive processing theory for the measurement of intelligence. *Educational Psychologist*, 24 (2), 185–206.
- Norman, D. A. (1993). *Things that make us smart: Defending human attributes in the age of the machine*. Reading, MA: Addison-Wesley.
- Oliver, R. (1991). The development of semantic programming knowledge in introductory programming teaching. In *Navigating the nineties: Proceedings of the Ninth Annual Australian Computers in Education Conference*, pp. 264–272. Brisbane: Computer Education Group of Queensland.
- Oliver, R., & Malone, J. (1993). The influence of instruction and activity on the development of semantic programming knowledge. *Journal of Research on Computing in Education*, 25, 521–533.
- Riordon, T. (1984). Helping students with recursion: Teaching strategies. *The Computing Teacher*, January, pp. 38–40.

- Shneiderman, B., & Mayer, R. E. (1979). Syntactic/semantic interactions in programmer behaviour: A model and experimental results. *International Journal of Computer and Information Sciences*, 8, 219–238.
- Troy, M. E., & Early, G. (1992). Unravelling recursion, Part I. *The Computing Teacher*, March, pp. 25–28.
- Walton, J. (1983). *Sequential and simultaneous information processing abilities and their interaction and instructional treatments in senior high school mathematics*. Unpublished doctoral thesis. University of New England (Armidale, NSW, Australia).



About the Author

Dr. Pam Gibbons teaches at the Australian Catholic University- Mount Saint Mary Campus, 179 Albert Rd, Strathfield. NSW. 2135 Australia.

Pam Gibbons

p.gibbons@mary.acu.edu.au

BOOK REVIEW / Continued from Page 8

learning; it is about what we accept as “knowledge” and its problematic relationship to reading and writing; it is about the structures and processes that characterize contemporary “developed” cultures and the interplay between the formal (school) and informal (family) contexts in which children grow and learn.

To followers of Papert's earlier work—and Logo enthusiasts in particular—much of the thrust of this work, its arguments, and the types of examples it uses will be familiar territory and probably well within the “comfort zone” of most educators. Parents, however, particularly if these ideas are new to them, might experience the type of discomfort felt by some teachers in relation to Papert's earlier writings. There are many implicit assumptions here about the nature of parenting and the role of both nuclear and extended families in con-

temporary society. These assumptions go well beyond surface observations about the type of learning that does or does not take place within families. There is a sense in which they parallel that deep critical consideration of the nature of schooling that was implicit in *Mindstorms* and brought into the open in *The Children's Machine*. As an educator I applaud *The Connected Family*. I endorse virtually all its points of view, and I particularly admire its valiant attempt to embed the practice of the philosophies in the form of publication. As a parent, though, I am less comfortable, particularly in relation to the apparently strict line of demarcation between the “old” and the “young” in almost every case but that of the author himself, who appears to occupy a uniquely privileged position as a mature but nevertheless childlike (in the best sense) user of technology.

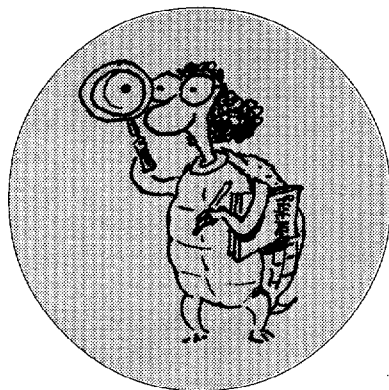
I hope this book is widely read. Although its benefits to parents who care deeply about how their children learn are clear (shouldn't that be all parents?), its most fertile audience might well be those young adults—perhaps but not necessarily with aspirations to a career in “education”—who as yet have no personal histories as either educators or parents and so have no cause to reject any of the book's central tenets in the dubious interest of justifying their current attitudes and practices!



About the Author

Dr. Carolyn Dowling is an Associate Professor and Head of the School of Arts and Sciences at the Australian Catholic University. She is the author of Logo books and numerous articles on technology and learning.

Carolyn Dowling
Australian Catholic University
412 Mt Alexander Rd.,
Ascot Vale, Victoria 3032
AUSTRALIA
c.dowling@mercy.acu.edu.au



This Much We Know: Part I, Mathematics

BY DOUGLAS H. CLEMENTS AND JULIE SARAMA

We believe Logo benefits learners. We believe this as teachers, and we believe this as researchers. Different teachers have different opinions. As disciplined inquiry, however, research goes beyond opinion. What does research say? We will answer this question by summarizing research we have reviewed in *Logo Exchange* for more than a decade. This body of research is so large that we must focus on main findings in a few areas. This column features mathematics. The next one will feature higher order thinking, creativity, language arts, and social-emotional development, and finish by asking, "Where do we go from here?"

Logo Aids Learning and Teaching Mathematics

"Logo programming, particularly turtle graphics at the elementary level, is clearly an effective medium for providing mathematics experiences. . . . When students are able to experiment with mathematics in varied representations, active involvement becomes the basis for their understanding." If teachers are active and reflective, then Logo helps students learn mathematics.

Turtle Graphics: A Good Way to Learn Geometry and Develop Spatial Sense

Writing a sequence of Logo commands—a procedure—to draw a geometric figure "allows, or obliges, the student to externalize intuitive expectations. When the intuition is translated into a program it becomes more

obtrusive and more accessible to reflection." In other words, students must analyze the visual aspects of the figure and reflect on how they can build it from parts. For example, in one study, 656 K-6 students who used Logo showed dramatic growth on correctly classifying squares (as a type of rectangle) compared to 644 control students. The Logo group outperformed the control group on the parallelogram items, so they were not overgeneralizing, that is, believing that all parallelograms are rectangles. Logo students did learn to apply the property "opposite sides equal" to the class of squares. They understood that the property "opposite sides equal" is not inconsistent with the property "all sides equal in length." Most students could apply both properties to the class of squares, demonstrating flexible consideration of multiple properties that may help lay the groundwork for hierarchical classification.

Effects of nonstructured programming in Logo on arithmetic skills appear small. However, if used in reflective ways, Logo can aid growth in arithmetic skills equally to CAI drill. A primary grade teacher of a "below average" class reported that she had written multiplication tables on the board. The students saw the pattern immediately, using the Logo command REPEAT to describe what they noticed. "This is the first time this has ever happened in my class," she said. "I am sure it is because of their Logo work. They are looking for patterns in

things more than they used to do" (Carmichael, 1985, p. 286).

Logo: An "Entry" to the Powerful Tool of Algebra

Logo is an environment in which some students perceive the use of formalizations such as variables as natural and useful. Again, however, we find that students' ability to generalize their Logo-based notion of variable may depend to a great degree on the depth of their Logo experience and the instructional support given them.

Logo Environments: Facilitating the Growth of Proportional Reasoning

On a geometric proportion task, students used additive strategies on paper-and-pencil tasks, but none used such strategies on the related Logo tasks. The reason lay in the interaction between students' formalization and computer feedback. They formalize proportional relationships algebraically as Logo programs. They receive graphical feedback regarding their mathematical intuitions. With pencil and paper, the formalization is less salient; the feedback is absent.

In general, Logo programming can help students build stronger connected knowledge (rather than mechanical chains of rules and terms) for mathematical topics. Thus, it is strongly compatible with mathematics standards.

Several unique characteristics of Logo facilitate students' learning.

- The commands and structure of the computer language are consistent with mathematical symbols and structures.
- Logo experience encourages students to view and describe mathematical objects in terms of the actions or procedures used to construct them.
- Logo promotes the connection of symbolic with visual representations, supporting the construction of mathematical strategies and ideas out of initial intuitions and visual approaches.
- The turtle's world involves measurements that are visible yet formal quantities, helping to connect spatial and numeric thinking.
- Logo permits students to outline and then elaborate and correct their ideas. Logo helps document student actions, leading the mathematical symbolization.
- Logo encourages the manipulation of screen objects in ways that facilitates students viewing them as mathematical objects and thus as representatives of a class.
- Logo demands and so facilitates precision and exactness in mathematical thinking.
- Logo provides a mirror of students' mathematical thinking. For teachers who are willing to work with and listen to students, such environments provide a fruitful setting. They help take the student's perspective and reveal previously unsuspected abilities to construct sophisticated ideas if given the proper tools, time, and teaching.
- Because students may test the ideas for themselves on the computer, they aid students in moving from naive to empirical to logical thinking and encourage them to make and test conjectures. Thus, Logo facilitates students' development of autonomy in learning (rather than seeking authority) and positive beliefs about the creation of mathematical ideas.

- Logo permits students to manipulate embodiments of certain mathematical ideas. Serving as a transitional device between concrete experiences and abstract mathematics, it may facilitate students' elaboration of the schemata for those ideas
- Logo is an environment in which students can use mathematics for purposes that are meaningful and personal for them.

When considering these benefits, one must keep in mind that the teacher's role is critical. Teacher mediation involves multiple actions. Teachers must be involved in planning and overseeing the Logo experiences to ensure that students reflect on and understand the mathematical concepts. They need to (a) focus students' attention on particular aspects of their experience, (b) elude informal language and provide formal mathematical language for the mathematical concepts, (c) suggest paths to pursue, (d) facilitate disequilibrium using computer feedback as a catalyst, and (e) continually connect the ideas developed to those embedded in other contexts. Teachers must provide structure for Logo tasks and explorations to facilitate desired learning. To accomplish all this, teachers need specifically designed Logo activities and environments.

The best use of Logo may involve full integration into the mathematics curriculum. Too much of school mathematics involves exercises devoid of meaning. Logo is an environment in which students use mathematics meaningfully to achieve their own purposes.

Finally, we need continuing research and development to expand our knowledge of what students and teachers learn in various Logo classrooms. Standardized tests do not measure many concepts and skills developed in Logo.

References

- Barker, W. F., Merryman, J. D., & Bracken, J. (1988, April). *Microcomputers, math*
- CAI, Logo, and mathematics education in elementary school: A pilot study. Paper presented at the meeting of the American Educational Research Association, New Orleans.
- Battista, M. T., & Clements, D. H. (1988). A case for a Logo-based elementary school geometry curriculum. *Arithmetic Teacher*, 36, 11-17.
- Battista, M. T., & Clements, D. H. (1991). *Logo geometry*. Morristown, NJ: Silver Burdett & Ginn.
- Butler, D., & Close, S. (1989). Assessing the benefits of a Logo problem-solving course. *Irish Educational Studies*, 8, 168-190.
- 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.
- Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, 20, 450-467.
- Clements, D. H., & Battista, M. T. (1992a). *The development of a Logo-based elementary school geometry curriculum* (Final Report: NSF Grant No.: MDR-8651668). Buffalo, NY/Kent, OH: State University of New York at Buffalo/Kent State University.
- Clements, D. H., & Battista, M. T. (1992b). Geometry and spatial reasoning. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 420-464). New York: Macmillan.
- Clements, D. H., & Meredith, J. S. (1993). Research on Logo: Effects and efficacy. *Journal of Computing in Childhood Education*, 4, 263-290.
- Clements, D. H., & Sarama, J. (in press). Research on Logo: A decade of progress. *Computers in the Schools*, 14(1-2).
- Hoyles, C. (1992). School interface and curriculum change. In S. Dawson & R. Zazkis (Eds.), *Proceedings of the Six International Conference for Logo and Mathematics Education* (pp. 3-25). Vancouver, BC, Canada: Simon Fraser University.
- Hoyles, C., & Noss, R. (1987). Synthesizing mathematical conceptions and their formalization through the construction of a Logo-based school mathematics

- curriculum. *International Journal of Mathematics Education, Science, and Technology*, 18, 581-595.
- Hoyles, C., & Noss, R. (1992). A pedagogy for mathematical microworlds. *Educational Studies in Mathematics*, 23, 31-57.
- Lehrer, R., & Smith, P. C. (1986, April). *Logo learning: Are two heads better than one?* Paper presented at the meeting of the American Educational Research Association, San Francisco, CA.
- McCoy, L. P. (1996). Computer-based mathematics learning. *Journal of Research on Computing in Education*, 28, 438-460.
- Miller, R. B., Kelly, G. N., & Kelly, J. T. (1988). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology*, 13, 348-357.
- National Council of Teachers of Mathematics. (1989). *Curriculum and evaluation standards for school mathematics*. Reston, VA: The Council.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Salem, J. R. (1989). Using Logo and BASIC to teach mathematics to fifth and sixth graders. *Dissertation Abstracts International*, 50, 1608A. (University Microfilms No. DA8914935)
- Tanner, H. (1992). Developing the use of IT within mathematics through action research. *Computers and Education*, 18(1-3), 143-48.
- Wilson, D., & Lavelle, S. (1992). Effects of Logo and computer-aided instruction on arithmetical ability among 7- and 8-year-old Zimbabwean children. *Journal of Computing in Childhood Education*, 3(1), 85-91.
- Yelland, N. (1995). Mindstorms or a storm in a teacup? A review of research with Logo. *International Journal of Mathematics Education, Science, and Technology*, 26(6), 853-869.

Acknowledgment

Time to prepare this material was partially provided by a National Science Foundation Research Grant, "An Investigation of the Development of Elementary Children's Geometric Thinking in Computer and Noncomputer Environments," NSF MDR-8954664. Any opinions, findings, and conclusions or

recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.



About the Authors

Dr. Douglas H. Clements, Professor at the State University of New York at Buffalo, has studied the use of Logo environments in developing children's creative, mathematics, metacognitive, problem-solving, and social abilities. Through a National Science Foundation (NSF) grant, he developed a K-6 elementary geometry curriculum, *Logo Geometry* (published by Silver Burdett & Ginn, 1991). With colleagues, he is working on the previously mentioned NSF research grant and finishing a second NSF-funded project, "Investigations in Number, Data, and Space," to develop a full K-5 mathematics curriculum featuring Logo. With Sarama, he is coauthoring new versions of Logo for learning elementary mathematics. One, *Turtle Math*, is currently available from LCSi.

Dr. Julie Sarama is an Assistant Professor at Wayne State University. She previously taught secondary mathematics and computer science, gifted math at the middle school level, and mathematics methods courses. She is coauthor of several investigations units and of *Turtle Math* and is currently designing and programming new versions of Logo and other computer microworlds.

Douglas H. Clements
SUNY at Buffalo
Dept. of Learning and Instruction
593 Baldy Hall,
Buffalo, NY 14260
Clements@acsu.buffalo.edu

Julie Sarama
Wayne State University
Teacher Education Division,
Detroit, MI 48202
Sarama@acsu.buffalo.edu

The units in *Investigations in Number, Data, and Space* tasks teachers to think in new ways about mathematics and asks how students best learn math. Some units add another challenge for teachers: Think about how computers might support and enhance mathematical learning.

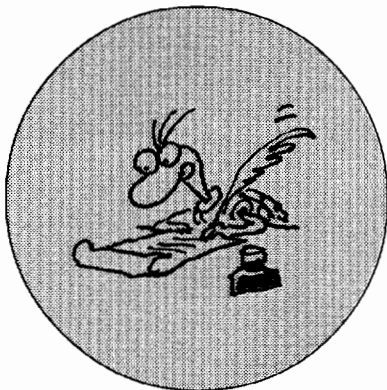
This innovative book series includes Geo-Logo, a special geometry-oriented version of Logo developed by *Logo Exchange's* own Doug Clements and Julie Sarama.

For more information, investigate the Investigations Web page at:

<http://www.awl.com/dsp/MainPages/Invest.html>



Send Logo News to
logoexchange@moon.pepperdine.edu



We Choose to Go to the Moon

by DANIEL E. KINNAMAN

We choose to go to the moon. We choose to go to the moon. . . . Not because it is easy, but because it is hard. . . . Because there is new knowledge to be gained. We shall send to the moon—240,000 miles away—from the control station in Houston, a giant rocket more than 300 feet tall, made of new metal alloys, some of which have not yet been invented . . . on an untried mission . . . on the greatest adventure.

These words were spoken 35 years ago by President John F. Kennedy at Rice University in Houston as part of a speech exhorting the United States to increase its commitment to space travel and exploration. Although not his intended subjects, his words also have a lot to say about what schools should be like and about Logo. Let's take a look at several of these phrases in light of schooling, learning, and Logo.

Not because it is easy, but because it is hard. This is true of Logo, too. You cannot embrace it passively. It isn't push technology. It only comes to life through intellectual engagement. Logo is powerful—not because it is easy, but because it is deep. As it continues to evolve in sync with Moore's law, it provides an ever more powerful multimedia studio for creative self-expression; a cyberlaboratory for testing ideas; a workshop for building tools, models, and systems.

Logo is an intellectual playground, but it isn't easy. For example, I can write recursive procedures that pro-

duce fun and interesting results (even desired and predicted results), but I still have difficulty explaining (and understanding fully) exactly how recursion works.

Logo is fun, but it is hard. It can stimulate creativity and encourage critical thinking, reflection, and conjecture. It often generates surprise, unexpected insights, and satisfaction. It is powerful—but it isn't easy, and it isn't passive entertainment.

When I write words such as these, my thoughts go beyond Logo to school in general. Isn't this what it should be about fundamentally—stimulating, challenging, engaging, and intriguing—hard fun? Henry David Thoreau warned that school should not be structured so that it makes a straight-cut ditch of a free meandering brook. Logo can help us overcome any such tendency.

Because there is new knowledge to be gained. And it can help us to remember that the universe of human knowledge is ever-expanding, and that there is still far more to learn than is now known. As Seymour Papert likes to point out, the traditional school curriculum generally covers only about one billionth of the knowledge in the universe, and we spend an inordinate amount of time arguing about which billionth that should be.

JFK was correct to emphasize the importance of constructing new knowledge from what is known. This doesn't dismiss the need to learn some basic facts via declarative presentation,

but it gives them more purpose. To get to the moon required using what was known to create new and additional knowledge and application. Sounds like Logo.

. . . made of metal alloys, some of which have not yet been invented. . . . We need to live and learn with the expectation that new ideas can be formulated, new processes can be designed, and new products can be developed and built. As Jerome Bruner wrote in *The Process of Education*, "Learning should not only take us somewhere; it should allow us later to go further more easily."

. . . on an untried mission . . . on the greatest adventure. When I first started teaching, I didn't know a lot of education jargon and I had no experience living on the teacher's side of the desk. What I knew intuitively was that to succeed I had to find ways to invite my students to become active participants on various academic adventures. This requires the acknowledgment that not all of the answers are yet known; that much uncharted and unexplored territory remains. The tremendous challenge in this was to find vehicles for the missions my students would take. That's why I got involved with technology, and with Logo. It had nothing to do with wanting to be part of a separate and exclusive culture. Logo simply showed itself to be a great vehicle for productive and enjoyable academic adventure.

To be sure, Logo has gone through an interesting evolution: from the early

speculation of its potential to reform schooling, to the widespread enthusiasm of the first International Logo Conference at MIT in 1984, and through all of the debate as to its relative importance during the commercial growth of the "educational software industry."

Although not exactly what any of us probably envisioned it would be, Logo is still here, of course, and still evolving. It's alive, and its influence should grow—but not because we need another miniculture. We should choose to keep Logo alive because of the sentiments noted in Kennedy's speech. After all, in less than a decade after the speech, there were nine manned missions to the moon. To the extent that Logo can help bring these sentiments to life in schooling, school can be a place in which many more learners experience that one small step, one giant leap sense of involvement and accomplishment.



About the Author

Daniel E. Kinnaman, a contributing editor to *Logo Exchange*, has more than 15 years experience as a classroom teacher and district-wide director of education technology. He also has taught graduate teacher education courses in conjunction with colleges and universities nationwide. He is executive editor of *Curriculum Administrator* magazine and serves as a consultant to Compaq Computer Corp., Safari Technologies, and school districts and education service centers throughout the United States.

Daniel Kinnaman

kinnaman@uconnvm.uconn.edu

CORLEY / Continued from Page 9

Approximately one-third of Frank's Priory students and all of his summer students have never seen Logo before they enter his class. This has not been a big problem. Students are asked to talk out their ideas and write them down in a sequential way. As the class discusses the ideas and their inherent mathematics, Frank then translates those ideas into Logo syntax. He has found that the majority of students make the transfer and develop the skill to program independently in Logo.

This summer's Greater Than One problem "gave rise to my most satisfying educational experience ever," Frank said. The kids labored hard, working out multiple proposals and finally agreeing on an answer they found the most likely. They moved to the computers and began to test their hypotheses. One student, Ryan, wasn't satisfied and quietly went in his own direction, working through the break. Suddenly, Ryan jumped up, walked to the board and started to explain and teach his idea. He had independently come up with a solution to the problem that just happened to be the formula for the volume of a four-dimensional pyramid. Bryan's approach, Frank's approach, and the class's approach were all totally different and produced the same results. The class burst into spontaneous applause! It doesn't get any better than that.

Frank Corley may be reached at: frank@priory.org



About the Author

Marian Rosen is an amazing Logo teacher at Conway School in Ladue, Missouri. She has led countless teacher workshops and graduate courses in educational computing. Marian is also a past president of SIGLogo.

Marian Rosen

mbrosen@icon-stl.net

25 Things to Do with LEGO and Logo

BY GARY STAGER

1. Plan and construct a "City of the Future"
2. Construct and give behaviors to a new LEGO animal
3. Construct and choreograph a LEGO ballerina
4. Construct and program a LEGO athlete
5. Construct and program to perform a job you dislike doing
6. Construct and program a LEGO supermarket scanner
7. Construct a LEGO toy factory
8. Construct a LEGO musical instrument or play a musical instrument with Control Lab
9. Construct and program a machine to send Morse code
10. Build a better LEGO mousetrap
11. Have a LEGO truck and tractor pull
12. Design, construct and program a LEGO kinetic sculpture
13. Invent a new type vehicle
14. Construct a machine for sorting & routing luggage at an airport
15. Design, construct & program a house of the future
16. Monitor the behavior of a gerbil or hamster with Control Lab
17. Construct a dog-walking machine
18. Create a robot arm that will pour you a drink
19. Invent a LEGO vending machine
20. Build and program a LEGO slot machine or roulette wheel
21. Construct a LEGO toll collector
22. Design and program a LEGO fax machine
23. Control a science experiment
24. Invent something never thought of before
25. Think of another 25 things to do with LEGO and Logo!



**No Hype.
No Hoopla.**

Just One Great Deal.

**Logo Lovers:
Trade-up to a**

MicroWorlds™

Site License For Just \$499†

trade-up@lcsi.ca
1.800.321.5646
<http://www.lcsi.ca>

† All registered owners of commercial Logo-based programs qualify. Price in US funds. Offer valid until July 31, 1997.
Not for resale.

© Logo Computer Systems Inc. (LCSI), 1997

ISTE BRINGS THE WORLD OF TECHNOLOGY CLOSER TO YOU.

By drawing from the resources of committed professionals worldwide, ISTE provides support that helps educators like you prepare for the future of education.

As an ISTE member, you benefit from a wide variety of publications, national policy leadership, and our work with Teacher Accreditation.

You also enjoy exciting conferences, global peer networking, and graduate-level Distance Education courses.

So if you're interested in the education of tomorrow, call us today.



International Society for Technology in Education
Customer Service Office
480 Charnelton Street, Eugene, OR 97401-2626 USA
Phone: 800/336-5191 (US and Canada)
Phone: 541/302-3777 (International)
Fax: 541/302-3778
America Online: ISTE CompuServe: 70014,2117
Internet: cust_svc@ccmail.uoregon.edu
World Wide Web: <http://isteonline.uoregon.edu>

WE'LL PUT YOU IN TOUCH WITH THE WORLD.