

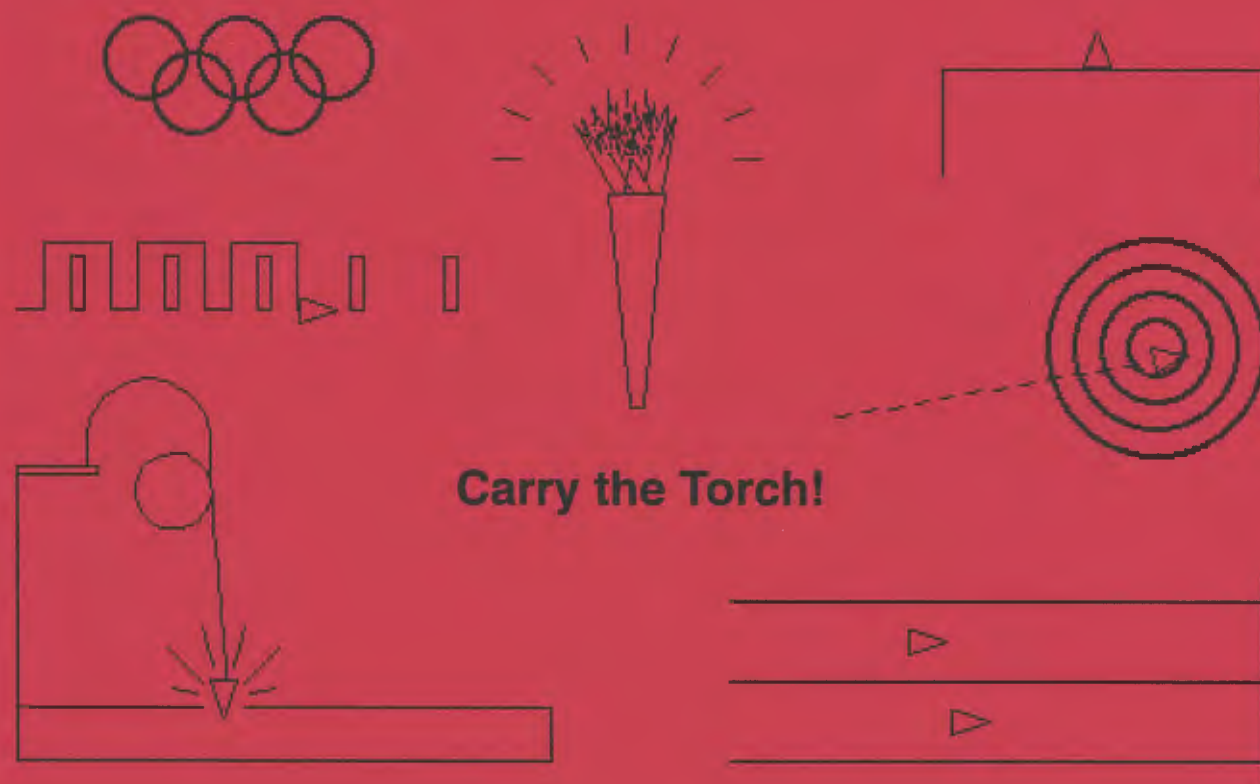


LOGO EXCHANGE

Volume 14 Number 4

Summer 1996

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



Carry the Torch!



In this issue:

The Historical Turtle
The Hudson River Project
Fun with Robotics
The Battle of Baltimore

Logo Teacher Training
Programming in *StarLogo*
An *LX* Retrospective
Summer Logo Workshops

Editorial Publisher

International Society for Technology in Education

Editor-in-Chief

Dorothy M. Fitch

Copy Editor

Jennifer Roland

Founding Editor

Tom Lough

International Editor

Dennis Harper

SIGLogo Officers

Chuck Priesen, President
Steve Senko, Vice-President
Hope Chafian, Secretary/Treasurer
Dorothy Fitch, Editor

Production

Kerry Lutz

Director of Advertising Services

Lynda Ferguson

Submission of Manuscripts

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.

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

Send surface mail to:

Dorothy M. Fitch
3 Derby Road
Derry, NH 03038

Send electronic mail to:

Internet: 71760.360@compuserve.com

Deadlines

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

Volume 15, Number 2	June 1, 1996
Volume 15, Number 3	Oct. 1, 1996
Volume 15, Number 4	Jan. 1, 1997
Volume 16, Number 1	Mar. 1, 1997

ISTE Board of Directors 1995-96

David Brittain, President
MGT. of America (FL)
David Moarsund, Executive Officer
Dennis L. Bybee, Associate Executive Officer

Executive Board Members

Connie Stout, Secretary
Texas Education Network (TX)
Terry Killion, Treasurer
Jefferson County RT School District (CO)
M.G. (Peggy) Kelly, Past-President
California State University - San Marcos
Terrie Gray, At Large
Chico Unified School District (CA)

Board Members

Kathryn Edwards Price
Apple Computer, Inc., Santa Monica (CA)
Peter Whollhan
Sts. Peter and Paul Schools, U.S. Virgin Islands
Paul O'Driscoll
Salem-Keizer Public Schools (OR)
Lynne Schrum
University of Georgia
Carla Schutte
Educational Technology Consultants (FL)
Gwen Solomon
U.S. Department of Education (DC)
Neal Strudler
University of Nevada - Las Vegas
Harnet Taylor
Louisiana State University
Brenda Williams
West Virginia Department of Education

Logo Exchange is published quarterly by the International Society for Technology in Education (ISTE), 1787 Agate Street, Eugene, OR 97403-1923, USA; 800/336-5191. This publication was produced using Aldus PageMaker®.

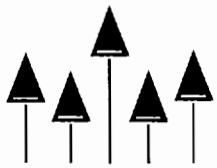
ISTE Members may join SIG/Logo for \$24.00. Dues include a subscription to *Logo Exchange*. Add \$10 for mailing outside the USA. Send membership dues to ISTE. Add \$4.00 for processing if payment does not accompany your dues. VISA, Mastercard, and Discover accepted.

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 the official policy of ISTE.

© 1996 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; 508/744-3350; FAX 508/741-2318. ISTE members may apply directly to the ISTE office for free reprint permission.

POSTMASTER: Send address changes to *Logo Exchange*, ISTE, 480 Charnolton St., Eugene, OR 97401-2626 USA. Second-class 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.



Logo Exchange

Volume 14 Number 4

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

Summer 1996

Contents

From the Editor:

Carrying the Torch *Dorothy M. Fitch* 2

Quarterly Quantum:

Let It Snow! *Tom Lough* 4

The Historical Turtle *Jim Muller* 6

Logo Workshops: Summer 1996 *compiled by the editor* 8

The Hudson River Project:

Computing the Water Quality Index using *MicroWorlds* *Orlando Mihich* 9

Reaction Time: Fun with Robotics! *LEGO DACTA* 15

Etchings: *LogoWriter* to *MicroWorlds* *John Gough* 18

The Artillery Battle of Baltimore *Jim Muller* 22

Windows on Logo:

The Last *LX* Column—A Retrospective Review *Glen L. Bull & Gina L. Bull* 25

To Glen and Gina Bull *from LX editors, past and present* 28

Logo: Search and Research

Parallel Programming in *StarLogo* *Douglas H. Clements and Julie Sarama* 29

Global Logo Happenings: *edited by Dennis Harper*

Constructing Change in Schools Through Teaching Training in Logo *José Valente* 34



From the Editor: Carrying the Torch

by Dorothy M. Fitch

In this issue we salute torchbearers, in both the figurative and literal senses. These torchbearers light the path ahead for the rest of us. They courageously guide us, shedding new light on familiar ideas and encouraging us to explore uncharted territory. The *American Heritage Dictionary* defines a torchbearer as: "1. One that carries a torch. 2. One...who imparts knowledge, truth, or inspiration to others."

LX's founding editor, Tom Lough, exemplifies both definitions of torchbearer. He has always carried a torch for Logo. Fourteen years ago it was his idea to start a newsletter for Logo users, which has evolved into this journal. He has been a guiding light for us all since then.

And now, Tom truly will be carrying a torch! Tom has been selected as a torchbearer to carry the Olympic flame through Hartford, CT for one mile on June 16. How exciting!

Not only that, but modest Tom, I have just learned, was a member of the USA Modern Pentathlon team at the 1968 Olympics in Mexico City! (The modern pentathlon includes horseback riding over jumps, fencing, pistol shooting, swimming, and running.) Having known Tom for so many years and just now finding out this exciting information was truly awesome, in the true sense of the word. As he wrote in an e-mail message, which he has allowed me to share:

I have not really kept my Olympic participation a secret; I just didn't proclaim it to everyone. I enjoy sharing my Olympic experiences with school groups and others. I take pleasure in encouraging people to dream and then to start on the path to realize their dreams. From my Olympic experience, I learned many valuable lessons, which I still try to apply in my everyday life. And, as you might expect, having represented our country in international competition has meant a lot to me.

This year, Tom has been instrumental in getting an online project called CyberTorch up and running. "CyberTorch: An Educational Project of Heroic Proportions" is a grass-roots project facilitated by the Global SchoolNet Foundation, a nonprofit organization established by Al Rogers of FrEdMail fame.

The purpose of the project is to provide a framework for schools, teachers, and students to incorporate the 1996 Olympic Torch Relay into their classroom activities. Across the country, teachers are developing math and science projects, geography lessons, language arts assignments, and physical education events that relate to the torch relay. Some students are interviewing local torchbearers, while others are designing and making mockup torches for them to carry in their running workouts.

Schools share their experiences via the CyberTorch online resources, or search the information for torch activity ideas of their own. What Logo-related projects can you think of for the CyberTorch project?

Check out the CyberTorch home page on the Web at <http://www.gsn.org/torch/>. You can also sign up for the torch-talk newsgroup of CyberTorch by sending an e-mail message to majordomo@gsn.org. Leave the subject line blank. In the body of the message, type the single line: subscribe torch-talk. Then send the message. Follow the route of the torch on the Coca Cola Torch World Wide Web page at <http://www.cocacola.com/olympics/torchrelay.html>.

Another Olympian feat is being a columnist for a computer education journal for 14 years! Glen Bull, one of Tom's original columnists, has done just that. He, too, is a torchbearer, having guided us in exploring Logo ideas, activities, and connections for so many years. Glen and Gina Bull have decided to step down from their time-honored position at the end of this year. Their last column is a fascinating and personal reminiscence about *Logo Exchange* and how Logo has

affected them. We wish them all the best and hope to hear from them again from time to time. We thank them here (and also at the end of their column) for their great contribution to the Logo community.

And Tom, as you run with the Olympic flame on June 16, we'll all be thinking of you—every step of the way.

repeat 5280 [fd 100!]

Happy Logo adventures!



Dorothy M. Fitch, *LX* Editor
3 Derby Road, Derry, NH 03038
Telephone: 603-425-2010 Fax: 603-425-6487
E-mail: 71760.366@compuserve.com

P.S. Tom tells me that the May/June 1984 issue of *Teaching and Computers*, published by Scholastic, Inc., includes a Logo article called "Light the Way to Logo With Logo Olympics" by Tom Lough, Steve Tipps, and Rebecca Poplin, pages 20–28. In the "Light the Torch" event, students use the turtle to draw steps leading up to the Olympic torch stand so that the flame can be lit. In the process, they learn something about the slope of a line. Dig out this article, if you can!

About the cover

The cover pictures offer ideas for designing your own "Turtle Olympics" events. These pictures were part of a Summer 1992 *LX* article by Dorothy Fitch. It seems appropriate and timely to share them again now. As the Summer Olympics approach, help your students design Logo activities to honor our world's athletes. Write a program for your favorite Olympic event. Draw a sporting scene. Use animation to show a turtle (or several turtles) in motion.



Quarterly Quantum: Let It Snow!

by Tom Lough

You may not believe this, but as I write this column, I am surrounded by snow. This winter in Connecticut was something else! At one point, we had nearly three feet of the white stuff on the ground! And more is on the way.

Naturally, with so much of it around, my thoughts turned to snow, and to snowflakes in particular. (And for those of you reading this summer issue of *LX* in warm or warmer temperatures, maybe some Logo snow might make you feel a little cooler!)

I recalled reading about the scientists and teachers working in the Antarctic this year. One of the research activities involved analyzing cores drilled out of deep holes in the ice. From an examination of the ice crystals, researchers hoped to learn something about conditions on earth when the ice was just snow on what was then the surface.

What a fascinating concept! Information frozen in time, embedded in the snowflakes.

I began to wonder how we might encode information in snowflakes drawn with the Logo turtle. Hmmm.

I started with the assumptions that every flake has six identical spokes and that all angles between snowflake parts are 60 degrees. After a lot of thinking and messing around, I finally set up a snowflake representation using a list of number lists.

Each number list corresponds to a spike on each of the six snowflake spokes. The numbers in the number list determine the size of the snowflake.

(Don't worry if this doesn't make much sense yet. Just wait until you try the procedures.)

Let's focus on some ideas for using the procedures in the classroom to help students understand how information can be encoded in designs such as snowflakes.

After you define the procedures (listed at the end of this column), you can draw a snowflake by simply typing **flake** followed by a list containing one or more lists of numbers. Here are a few examples. Notice that the lists of numbers are grouped by square brackets into a single list (called **spokelist** by the procedures).

```
flake [ [2 2] [4 1] ]  
flake [ [4 3 2 1] [7 5 3] [9] ]  
flake [ [5] ]
```

Students like to figure out how the numbers affect the size and design of the snowflake. For convenience, each number in a list is multiplied by a factor of 10 in the **spike** procedure. This allows students to enter small numbers (even decimal numbers, such as 2.4) and still get reasonably large results.

Eventually, the snowflake drawing activity can be turned inside out, often by a student who asks another, "Betcha can't guess my numbers by looking only at my snowflake."

When this happens, the students are doing some thinking similar to that of the Antarctic researchers. They are trying to make sense of what they are seeing and to extract information (or even estimates of information) from their observations. Unlike the polar scientists, students can test their hypothetical data by drawing a snowflake based on it and comparing it with the original.

After students get acquainted with the **flake** procedure, some of them will prove to be amazingly adept at estimating what the input list looks like.

What other activities and thinking processes can you develop around these ideas? I'd love to hear about them.

If you are interested particularly in the programming aspect, feel free to dissect and adapt

the procedures in any way you like. I would enjoy learning what you do with them.

Above all, keep cool! Until the next snow-storm, ▲

FD 100!

Tom Lough

Founding Editor

P.O. Box 394

Simsbury, CT 06070

E-mail: 70020.223@compuserve.com

P.S. As an extension, develop a procedure to generate randomly the required list of number lists for drawing a different snowflake each time. I would appreciate seeing a copy of what you did. If you would like a listing of this procedures I developed, send me a self-addressed stamped envelope and request "Snowflake Design" procedures. Overseas subscribers please send equivalent international postage-return coupons.

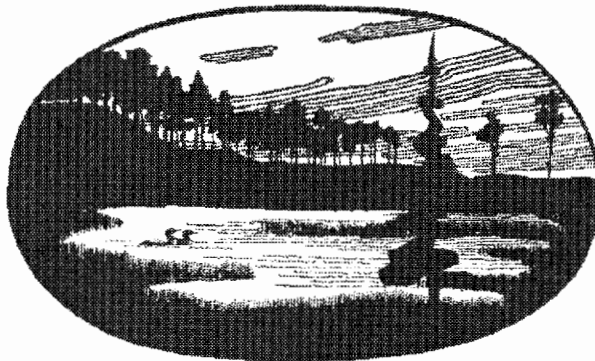
```
to flake :spokelist
repeat 6 [spoke :spokelist right 60]
end
```

```
to spoke :spokelist
if :spokelist = [ ] [stop]
spike first :spokelist
spoke butfirst :spokelist
end
```

```
to spike :spikelist
if :spikelist = [ ] [stop]
forward tenfirst :spikelist
left 60
spike butfirst :spikelist
right 120
spike butfirst :spikelist
left 60
back tenfirst :spikelist
end
```

```
to tenfirst :spikelist
output 10 * ( first :spikelist )
end
```

NECC '96—Call of the North



Minneapolis, MN • June 11-13, 1996

NECC '96 promises to bring you an exciting lineup of speakers from both the K-12 and Higher Education community, and the largest exhibition of educational technology products and services ever featured at a National Educational Computing Conference.

Conference highlights include:

- An exciting educational program—featuring practitioner-led pre-conference workshops and a wide variety of concurrent project, focus, poster, and paper sessions on topics including: Internet, WWW, multimedia, ethics, computer science, curriculum integration, and business/industry/school partnerships.
- "Flavors of Minnesota"—an opening reception co-sponsored by Compaq Computer Corporation that will feature displays and entertainment representing Minnesota's broad ethnic base.
- The biggest educational technology exhibition yet—nearly 700 booths will fill two domes of the beautiful Minneapolis Convention Center.

To receive a copy of the NECC '96 Advance Program, containing registration and housing information, please contact:

Donella Ingham, NECC '96/ISTE
1787 Agate Street, Eugene, OR 97403-1923
(PH) 541/346-2834, (FX) 541/346-5890
necc@oregon.uoregon.edu

NECC '96/TIES
2665 Long Lake Road, Suite 250
Roseville, MN 55113-2535
(PH) 612/638-8764, (FX) 612/638-8769,
necc96@ties.k12.mn.us

You can also access information contained in the NECC '96 *Advance Program*—and all previously published NECC '96 materials—by visiting NECC's home page at:
<http://isteonline.uoregon.edu/necc/necc96home.html>

For information on exhibiting at NECC '96,
Contact Paul Katz, Exhibits Manager
(PH) 541/346-3537, pdkatz@oregon.uoregon.edu

HOSTED BY: TECHNOLOGY AND INFORMATION EDUCATIONAL SERVICES (TIES)
SPONSORED BY: NATIONAL EDUCATIONAL COMPUTING ASSOCIATION (NECA)

NECC '96
Minneapolis



The Historical Turtle

by Jim Muller

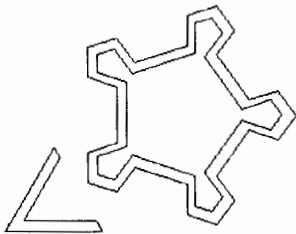
The day always seems to arrive in any Logo fan(atic)s life when you seem to run out of ideas. You've done every type of geometrical exercise and word game you can think of. There has to be something more than just another graphics project.

Geography is a subject that offers many possibilities. What about history?

While working with middle school classes some time ago, we looked at several famous battlefields from the early days of this country. In addition to the Alamo—what else would Texans study—we had a lot of fun with the Battle of Baltimore in the War of 1812.

The British brought a large naval force up the Chesapeake Bay and into Baltimore Harbor. They knew that their ships had more and larger guns than the rather small fort that guarded the entrance to the city.

Fort McHenry is small when compared to many other forts built in the 18th and 19th centuries. It is a star-shaped structure that sits on a point of land that juts out into Baltimore Harbor, which makes it the perfect spot to protect the city.



The barracks and other buildings line the inside walls of the fort. In the center is a large flagpole that made the fort and the battle famous. It was on that pole that the Star Spangled Banner flew.

Earlier in the war, the British had captured William Beanes, an American from Maryland. He was being held on one of the British ships

moving into Baltimore. Francis Scott Key and John Skinner, both from Washington, DC, received the permission of President James Madison to negotiate for the release of Beanes.

The British agreed to release Beanes. However all three men would have to remain on board the prisoner exchange ship until after the battle so they would not be able to warn the American patriots that they were coming.

The bombardment started on Tuesday, September 13, 1814, and continued all that day and almost all night. Key and his friends knew that Fort McHenry was no match for the huge guns of the British fleet. Sitting well back from the battle line, they paced the deck of their ship all night. Even when dawn broke, they did not know who had won the battle because the smoke and haze was so thick.

Suddenly there was a break in the smoke and haze and the Americans could see the American flag still flying over the walls of the fort. Key was so excited, he pulled an unfinished letter from his pocket and began to express his feelings.

Oh! Say can you see, by the dawn's early light,
What so proudly we hailed at the twilight's last gleaming?

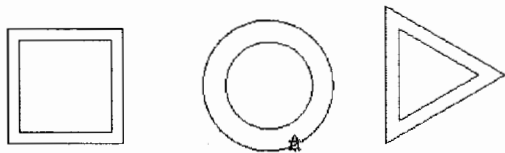
Key completed most of the poem within a few minutes. Shortly after that, the British gave up the fight and left Baltimore Harbor, leaving the three Americans behind. The poem was printed on handbills the next morning and distributed throughout the city as the citizens celebrated their victory. A few days later, it was sung to the tune of an English marching song that the Americans had adopted and named "Adams and Liberty." "The Star-Spangled Banner" became famous immediately and was sung at the Battle of New Orleans three months later.

"The Star Spangled Banner" is certainly an inspirational story for the children. But what

they couldn't understand was why the British lost the battle. They had many battle ships with more and larger guns than the fort. They had a large force of ground troops, many more than the rather small American force that was defending the city.

One of the things we talked about was the shape of the fort. It was shaped like a five-pointed star. The inner and outer walls of the fort were high and very thick. At each point of the star, there was a gun emplacement that allowed the gunners to cover every possible angle. Protected by high, thick walls and hidden in a small place that was difficult to hit, the guns were well protected.

In our Logo sessions, we had gotten to the point where we were discovering how to put shapes within shapes: squares spaced within squares, triangles spaced within triangles, for example.



Looking at the diagram of the fort, it became a challenge to recreate the star image.

```
to fort.mchenry
outside.wall
move
inside.wall
move2
ravelin
end
```

```
to outside.wall
repeat 5 [outside.corner]
end
```

```
to outside.corner
forward 46 left 72
repeat 3 [forward 25 right 72]
forward 25 left 72
end
```

```
to move
pu right 90 forward 8
left 90 back 3 pd
end
```

```
to inside.wall
repeat 5 [inside.corner]
end
```

```
to inside.corner
forward 53 left 72 forward 25 right 72
repeat 2 [forward 15 right 72]
forward 25 left 72
end
```

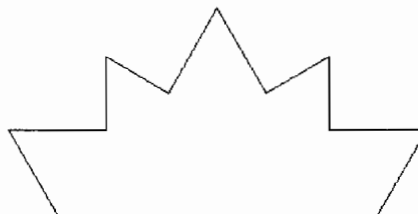
```
to move2
pu back 70 left 90 forward 20 pd
end
```

```
to ravelin
repeat 2 [forward 80 right 120]
forward 10 right 60 forward 60
left 120 forward 60 right 60 forward 10
end
```

Ravelin? What's a ravelin? A ravelin is a triangular fortification outside the main defensive structure; in this case, outside the fort.

There were actually several ravelin embankments that faced the harbor as a first defense against an invasion from the sea. It shielded men and guns from an advancing enemy.

One portion of the ravelin is shown with the diagram of the fort. Here's another to show what the actual fortification looked like.



Here's a challenge for you and your students: Add the actual ravelin shapes to the picture of the fort.

Those who want to take a look at the Battle of Baltimore from the British perspective and why they failed to silence the American guns can read the companion article starting on page 22 of this issue, "The Artillery Battle of Baltimore."

Jim Muller is the moderator of the Logo Forum on CompuServe. You can read more about him at the end of his other article in this issue, "The Artillery Battle of Baltimore." You can reach him via e-mail at 76703.3005@compuserve.com or on CompuServe at 76703,3005.

Logo Workshops: Summer 1996

Logosium — Minnesota

June 14 in St. Paul

The third annual Logosium in St. Paul, Minnesota will be a day of Logo workshops, discussions, and presentations hosted by the world renowned St. Paul Logo Project. This year, Logosium will include sessions conducted by students sharing their projects with other students and with adults.

Registration details appear on page 27.

Logo St. Paul — Minnesota

June 17–21 or August 19–23 in St. Paul

Over the past 14 years, the St. Paul Logo Project has provided a comprehensive professional development program for hundreds of elementary and secondary school teachers. A limited number of places has been set aside for people from outside the St. Paul Public Schools.

Registration:

\$490 per person; \$126 for university credit

For more information, contact:

Michael Tempel

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

Telephone: 212-765-4918; Fax: 212-765-4789

E-mail: michaelt@media.mit.edu

Summer at Spence — New York

June 24–28 in New York City

The Spence School, an independent school for girls located in the heart of New York City, just steps from Manhattan's Museum Mile, is a leader in educational technology. Join the Spence staff for a week of Logo exploration and creation.

Registration:

\$790 per person

For more information, contact:

Michael Tempel

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

Telephone: 212-765-4918; Fax: 212-765-4789

E-mail: michaelt@media.mit.edu

Stonington Retreat — Maine

June 30–July 6 on Deer Isle

Led by Laura Allen and Michael Tempel, the Stonington Retreat is an experiment in creating a reproducible learning experience for teachers interested in using computers and in thinking about their implications for the future of education. Each day will include a morning meeting, reflection and discussion, hands-on lab time, a local culture talk, and free time to explore Logo or the area.

Registration:

\$1,500–1,600 per person, includes tuition, bed and breakfast lodging, all breakfasts, three dinners, and a scenic boat ride.

For more information, contact:

Laura Allen

P.O. Box 20616, Cherokee Station
New York, NY 10021-0071

Telephone: 212-744-2447 or 617-354-2370

E-mail: laurallen@aol.com

Pepperdine University Summer Logo Institute & Conference

August 5–10 in Culver City, CA

Led by Gary Stager, the Summer Logo Institute (August 5–9, plus the Logo Conference) provides a unique collegial environment in which educators can learn Logo, enhance their programming skills and reflect upon the roles computers may play in the creation of constructive learning environments for children.

The Logo Conference (August 10) is an opportunity for those people unable to attend the Institute to recharge their batteries, learn Logo, share classroom project ideas, and attend minds-on workshops led by Institute participants and exciting guest speakers.

Registration options:

- 1) Institute and Conference, \$499
- 2) Logo Conference on August 10, \$75 (includes lunch)

Local hotel accommodations are available.

For more information, call: 310-568-5704

E-mail: GStager@Pepperdine.Edu

The Hudson River Project: Computing the Water Quality Index using *MicroWorlds*

by Orlando Mihich and Students

Introduction

The goal of this ongoing project is to help students develop into citizens who possess the knowledge, skills, and attitudes to engage in environmentally responsible behaviors. Booker T. Washington, Middle School 54, is located on the Upper West Side of Manhattan. The Hudson River, one of the most beautiful sites in New York, is within walking distance from the school. The question is: How is the Hudson River doing? The project began three years ago with the support of an enlightened administration and a yearly budget of \$250–\$300.

Project Description.

Once a week, a group of seventh- and eighth-grade students, "The Water Watch Team," visits the Boat Basin on the Hudson River, at the 79th Street Marina. Here, students record air and water temperatures, take pH readings, determine the water clarity and dissolved oxygen levels, and prepare a sample for the five day "biochemical oxygen demand" test. Sewerage pollution is tested using a Coli-count sampler. A water sample is brought back to school to determine levels of nitrates, phosphates, and solids.

In their work, students follow an "Internal Manual" compiled from different sources: the literature that comes with the suppliers' test kits, various electronic encyclopedias, and mainly, the "Field Manual for Water Quality Monitoring." The "Internal Manual" was written in part by students and contains a general section on the world's rivers, their history, the watershed, and some information on the utility and history of the Hudson River. For the most part, our "Internal Manual" conforms to the procedures outlined in the "Field Manual for Water Quality Monitoring." Other procedures requiring open-flame burners, filtration, and the use of sulfuric acid or other harsh chemicals have been modified. For example, the Coliforms

test is performed using the pocket-size Millipore Coli-count sampler. Prepared on location, the sampler is incubated at 35° C for 22–24 hours in the school's lab.

Following is a brief description of the importance of the nine tests.

1. *Dissolved Oxygen (DO)*. Dissolved oxygen is a measure of the health of a body of water. The absence of oxygen indicates severe water pollution.
2. *Fecal Coliform*. Fecal coliform levels are monitored because of the correlation between viruses that cause diseases and fecal coliform and pathogenic bacteria.
3. *pH*. Natural waters have a pH value from 5.0 to 8.5. Higher or lower levels are unsuitable for most organisms.
4. *Biochemical Oxygen Demand (BOD) [5-days]*. Biochemical oxygen demand is a measure of the quantity of the oxygen that disappears from the water because of the decomposition of organic matter.
5. *Temperature*. Thermal pollution from industries, urban runoffs, soil erosion, and deforestation affect the amount of dissolved oxygen and rate of photosynthesis.
6. *Turbidity*. Increased amounts of suspended solids reduce the transmission of light and cause waters to become warmer, with a consequent drop in oxygen levels.
7. *Phosphates*. The amount found in healthy water is generally small, not more than 0.1 ppm. Larger amounts of phosphates usually wipe out the river's fish population.
8. *Nitrates*. The presence of excessive amounts of nitrogen promotes plant growth and decay, which in turn increases biochemical oxygen demand.



9. **Total Solids.** High concentrations of suspended solids reduce water clarity, and contribute to a decrease in photosynthesis.

Students enter all test results into a *ClarisWorks* spreadsheet and generate charts for a discussion of results. Once all nine tests are completed, students use *MicroWorlds* to find the Water Quality Index.

Computing the Overall Water Quality Index

The Water Quality Index (WQI) is a standard index developed by the National Sanitation Foundation to compare rivers' waters. The WQI ranges from 100 for Excellent Water Quality to 0 for Very Bad Water Quality. To obtain the WQI number, it is necessary first to find the **Q-value**, the quality value, for each test. This is done with a series of nine "weighting curves," one for each test. Once found, the Q-value is multiplied by a "weighting factor" to obtain the **Total Q-value** for the test. The sum of all Total Q-values gives the **Overall WQI** for the testing site.

During the first year of testing, students computed the Q-value by consulting the charts provided in the "Field Manual for Water Quality Monitoring." *ClarisWorks* was employed to find overall WQI.

The second year, students recreated all nine diagrams in *LogoWriter*, and wrote a program to find the Q-value by positioning the turtle on the abscissa and typing "find" in the Command Center. At the same time we realized that several "weighting curves" could be changed and adapted to local conditions; e.g., Total Solids never exceeded 20 mg/L, so we changed the existing diagram's range of 0 to 500 mg/L to a range of 0 to 100 mg/L; the curve was redrawn accordingly. Now, we were able to position the turtle more accurately and achieve more precise results.

With the introduction of *MicroWorlds* it is now feasible to find the Q-value, the Total Q-value, and the overall Water Quality Index without switching to *ClarisWorks*.

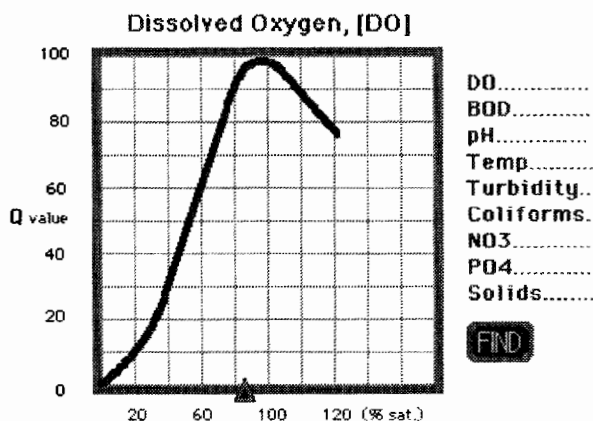
Discussion of results

For the school year 1994–1995, the WQI ranged from MEDIUM to GOOD. Fecal coliform bacteria exceeded 200 colonies/100 mL. The Hudson River's water does not meet the state standard for total body contact such as swimming. Dissolved oxygen levels were healthy, above 4 mg/L. Biochemical Oxygen Demand levels were higher than expected, indicating the presence of large amounts of oxygen-consuming microorganisms.

On May 18, 1995, the "Water Watch Team" students were awarded the New York Citywide and Boroughwide first prizes for their work in The Hudson River Water Watch Project as part of the Citywide "Team-Up-To-Clean-Up" Competition. In December 1995, "GREEN," the journal for the Global Rivers Environmental Education Network, published an article describing the students' work.

MicroWorlds as a programming tool

Following are sample pages for DO and WQI:

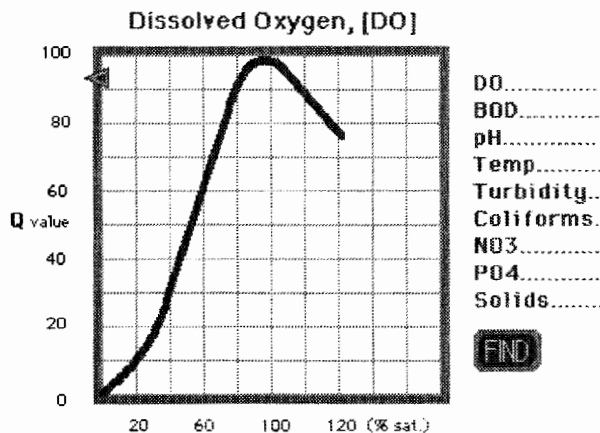


Students position the pointer for the % of DO on the abscissa, then click the FIND button. This turtle, t3, finds the Q-value for DO, using the procedure, **find.DO**.

Name:

Instruction:

Do it: ☒ Once ☐ Many Times



The pointer travels to the curve, turns toward the ordinate, and travels to it. The Q value shows in the active dialog box.

Q value * W.F.

DO.....
BOD.....
pH.....
Temp.....
Turbidity..
Coliforms..
NO3.....
PO4.....
Solids.....

93 * 0.17 = 16

Here, the Q-value is multiplied by the Weighting Factor (W.F.) to obtain the Total Q-value. To continue, students click the ARROW button. The new Biochemical Oxygen Demand page comes up. In the dialog box the Q-value for Dissolved Oxygen is displayed with a plus (+)

sign. The sum signs are added throughout with each new result. The last page computes the Overall Water Quality.

Dissolved Oxygen (DO) Procedures

```
to start.D0
; Positions t1 at the origin of the diagram;
  creates the reference turtle "zero.Q;
  creates the textbox "Q-D0.
```

```
t1, st
pu setpos [-185 -95]
setsh "pointer1
setsize 20
make "a pos
newturtle "zero
ht
pu setpos :a
newtext "Q-D0 [125 100] [120 155]
settc "125
setstyle "bold
end
```

```
to find.D0 ;button
; Prevents the accidental clicking of the
  "Find" button when the Total Q-value is
  already found, and when the Total Q-value
  is not yet found. Finds the Q-value.
if alert.D0 = 1 [alert.sound stop]
ifelse (ask "t1 [pos]) = :a [announce [In-
  troduce your result on the abscissa.]]
  [DO]
end
```

```
to D0
t1,
seth 0
if colorunder = 15 [seth 270 pu fd 10 go
  q.D0 stop ]
pu fd 1
DO
end
```

```
to Q.D0
; Rounds and adjusts the Q-value for the
  existing 20 x 20 grid; inserts the weight-
  ing factor; gives the Total Q-value.
talkto "Q-D0
insert round value / 2
insert char 32
insert "|* 0.17|
insert char 32
insert "="
insert char 32
insert round (value / 2) * 0.17
end
```



```

to alert.DO
talkto "Q-DO
output textcount "Q-DO
end

to move.value.DO
; Selects and copies only the Total Q-value.
top
select
cd
repeat 2 [cb]
cut
top
select
cd
copy
end

to end.DO ;button
; Prevents getting the next page before Q-
value is found; copies the Total Quality
Value; removes the textbox; gets and
starts the next page.
ifelse alert.DO < 1
[announce [Please, find Q-value first.]]
[move.value.DO remove "Q-DO getpage
"|Biochemical O2 Demand| start.BOD]
end

to go
setsh "pointer2
if colorunder = 55 [stop]
pu fd 2
go
end

to value
t1, output distance "zero
end

to alert.sound
; plays the Mac "sosumi" sound
sosumi
end

```

After all of the test results have been computed, the final page gives students the Overall Water Quality Index when they click the FIND button.

The Overall Water
Quality Index is:

Good

DO.....	16 +
BOD.....	5 +
pH.....	10 +
Temp.....	9 +
Turbidity..	6 +
Coliforms..	3 +
NO3.....	10 +
PO4.....	5 +
Solids.....	6

FIND

70



Water Quality Index (WQI) Procedures

```

to start.WQI
; Creates two new textboxes: "Total displays
the numerical sum of all "Total Q-values"
generated throughout the project, and
"|Water Quality| displays the water qual-
ity range, from "Excellent" to "Very Bad,"
according to the numerical value.
t1,
setsh "pointer2
pu setpos [0 0]
newtext "Q-WQI [125 100] [120 155]
settc "125
setstyle "bold
paste
newtext "Total [125 -55] [120 30]
newtext "|Water Quality| [-150 18] [140 40]
end

to find.WQI ;button
if alert.WQI = 1 [announce [Please,
continue.] stop]
settotal run Q-WQI WQI
end

to WQI
name total "result
talkto "|Water Quality|
if :result = 100 [print "Excellent]
if and :result > 90 :result < 100 [print
"Excellent]
if :result = 90 [print "Excellent]
if and :result > 70 :result < 90 [print
"Good]
if :result = 70 [print "Good]
if and :result > 50 :result < 70 [print
"Medium]
if :result = 50 [print "Medium]
if and :result > 25 :result < 50 [print
"Bad]
if :result = 25 [print "Bad]
if and :result > 0 :result < 25 [print
"|Very Bad|]
if :result = 0 [print "|Very Bad|]
end

```

```

to alert.WQI
talkto "Total
output textcount "Total
end

to end.WQI ;button
ifelse alert.WQI < 1 [announce [Please, find
the WQI Index.]]
[dolist [i [zero Q-WQI Total |Water
Quality|]] [remove :i]
Solids
getproject "WWT+1]
end

```

The last instruction, `getproject "WWT+1`, brings up a page with a picture of the Water Watch Team.



This project owes its continuing success to the enthusiastic involvement of **Michael Estrada** as student in charge, and the following students: Carolina Alcantara, Nestor Amaya, Socrates Brito, Marilyn Carabajo, Alex Estevez, Rufino Estevez, Lilliana Estrella, Tracy Fields, Laura Guzman, Loamy Hodge, Sam Lazarus, Alexandria Matos, Wilfredo Pichardo, Patrick Mohaney, Haydee Pimentel, Aaron Riccio, Wilmer Ruez, and Maria Solares.

The following page shows *ClarisWorks* sample charts with a brief discussion by the students.▲

Bibliography

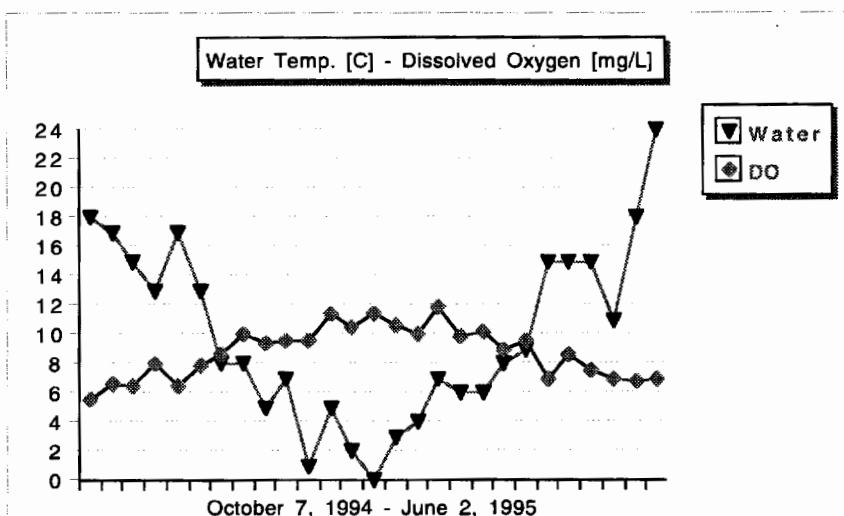
Mitchell, M.K., & Stapp, W.B. (1994). *Field manual for water quality monitoring*. (Available from GREEN, 721 E. Huron Street, Ann Arbor, MI 48103.)

Yoder, S. (1996). *MicroWorlds—Hypermedia project development and Logo scripting*. Eugene, OR: ISTE.

Orlando Mihich is a computer teacher at Booker T. Washington, MS 54, Community School District 3, New York City. He has taught science and Logo for the past 11 years. Prior to that, he was a chemical engineer in the surfactants and cosmetics industries. He can be reached by e-mail at Omihich@aol.com.

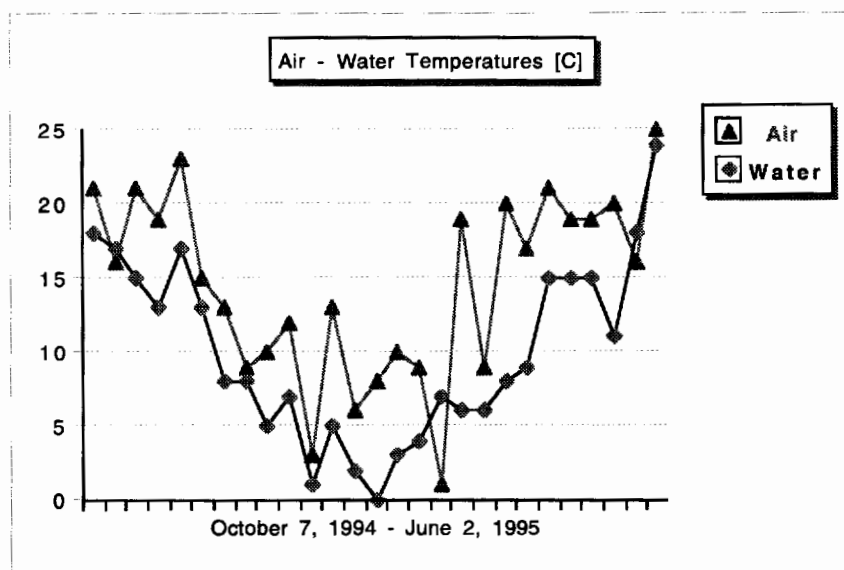
This chart clearly shows that gases, like oxygen, dissolve more easily in cooler water than in warmer water. In addition, healthy levels of DO, above 4 mg/L, were found for the Hudson River at the 79th Street location during the testing period.

By: Laura Guzman
Loamy Hodge



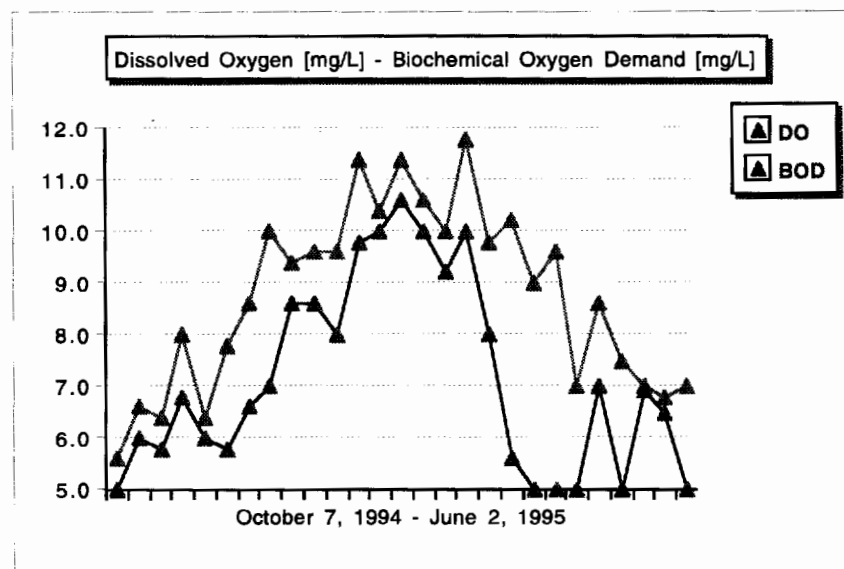
Water temperatures, in general, follow air temperatures, but water, a heavier fluid, responds less drastically to variations in temperature.

By: Alejandro Cruz
Ruez Wilmer



This diagram shows that, with several exceptions, most of the oxygen is used up in the five-day test by organic matter and microorganisms in the water.

By: Michael Estrada



Reaction Time: Fun with Robotics!

[Editor's note: This article was originally published in the premier issue (Spring 1989, Vol. 1, No. 1) of *LEGO DACTA CONNEXION*™, a newsletter for *LEGO® TC logo* Users. Reprinted with permission of LEGO Dacta, the educational division of LEGO Systems, Inc.]

To the Teacher:

Key ideas

1. Learning about the brain and voluntary muscle functions.
2. Practicing the scientific method.
3. Introducing the primitive **random**.

Teaching notes

A *stimulus* is something that causes a reaction. Light, sound, and electrical impulses are examples of stimuli.

The scientific method is often broken down into four or five parts. We will use four parts in this exercise: stating a hypothesis, designing and conducting an experiment, recording results, and reaching a conclusion.

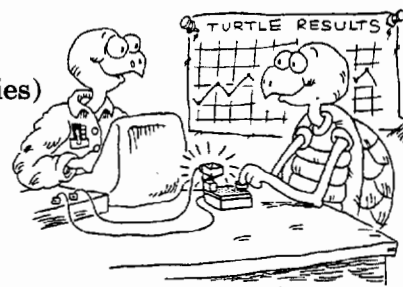
Students may need to be reminded that a hypothesis is an educated guess and that the experiment may disprove the hypothesis. Students should not be concerned about having their original hypothesis be "right." It is more important that experimental results become the basis for new ideas.

Your brain is the busiest, most complex organ in your body. It acts as a message center, sending and receiving messages through the nervous system to all parts of your body. Even the simplest task, such as standing, requires an incoming "balance" message from the ear and an outgoing command message to your legs and feet. The "message center" contains 10 billion neurons, each processing as many as 50,000 incoming and outgoing messages per minute, traveling 100 miles per hour through your body! Although everyone has a message center, we do not all react to stimuli at the same rate of speed.

In this activity, you are going to use the scientific method to examine how quickly people respond to a light stimulus.

Materials needed

- 1 *LEGO TC logo* system
(or other robotics system
with appropriate capabilities)
- 1 touch sensor
- 1 light brick
- miscellaneous elements
- Inventor's Notebook or
sheet of paper



Designing an experiment

Before you can set up an experiment, you need a hypothesis—a statement that you can test. For example: Because girls mature faster than boys do, they will react to a stimulus faster than boys will.

1. Record your hypothesis in your Inventor's Notebook or on a sheet of paper.
2. Build a reaction machine.

The machine pictured on the next page uses a LEGO light brick for the light stimulus. A touch sensor serves as the button that subjects will press to turn off the timer when they notice the light come on. Design your own reaction machine using a light and a touch sensor or an optosensor.

Students may invent different reaction machines. For example, they might replace the touch sensor with an optosensor and light brick, creating an "electric eye" to stop the timer.

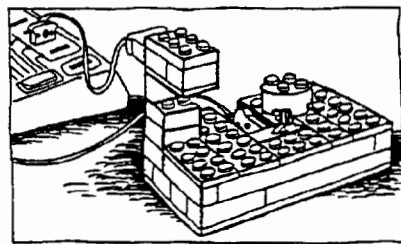
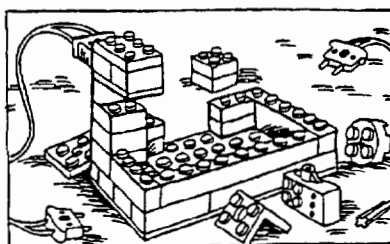
Including **wait random 250** in the procedure adds an element of surprise. Subjects are less likely to learn when the program starts and when the light will turn on.

If the student conducting the experiment wishes to control when the light turns on, another touch sensor, in addition to the touch sensor controlling the timer, can be added to the program. Plug another touch sensor into port 7 and try the following procedure.

```
to touch.light
  listento 7
  waituntil [sensor?]
  talkto 1 on
  resett
  listento 6
  waituntil [sensor?]
  off
  show timer/10
end
```

To make the experiment more impartial, assign a number to each subject. Have students record the name and number of each subject in their Inventor's Notebooks. Example: Jane-34; Sarah-56; Robert-23; and so on. Then response times can be recorded as Subject 34: 5, 4, 5.

To have a fair sampling of subjects for each experiment, it may be helpful for each team to state its hypothesis, set up its experiment, and have all class members act as subjects for each group in turn. Or, as a class, decide on a hypothesis to test. As the initial hypothesis is proved or disproved, each group can develop a new hypothesis.



3. Program your reaction machine. Here is one way to do it:

```
to light.react
  wait random 250
  talkto 1 on
  resett
  listento 6
  waituntil [sensor?]
  off
  show timer/10
end
```

The procedure **light.react** does the following:

1. waits a random amount of time between 0 and 25 seconds.
2. turns the light on.
3. resets the timer.
4. listens to the touch sensor.
5. waits until the touch sensor is pressed.
6. turns the light off.
7. displays the timer amount in seconds.

Conducting the experiment

1. Divide your subjects into appropriate groups. For example, if you are testing whether girls are faster than boys, split your subjects into two groups—girls and boys.

You may want to establish some rules for the test. For example, you may want to position subjects so that they cannot watch the operator. You may also have subjects place their hands a specified distance from the machine—perhaps on the table. *Remember: All subjects must follow the same rules or the experiment will not be accurate.*

2. Conduct tests on your subjects. Tell them to watch for the light to go on. As soon as they see the light, they must press the button.

Students should try at least two tests of response time. Data recorded must be accurate and reliable. The more trials run per subject, the more accurately and reliably data will reflect a subject's reaction time. Also, the more subjects used per experiment, the more accurately and reliably an experiment will reflect the general population.

Challenge

The following program can be used to test reaction time to sound:

```
to sound.reaction
wait random 250
tone 440 1
resett
listento 6
waituntil [sensor?]
off
show timer/10
end
```

Discussion

Is there an age when reactions are fastest? Ask the students to formulate a hypothesis for testing.

In addition to reacting to a stimulus, the brain is capable of learning. If the subjects practice, will their reaction times improve?

We depend on fast reactions to protect us from harmful stimuli in our environment. For example, fast reactions protect us from becoming badly burned when we touch a hot stove. What are some other examples?

While the subjects were involved in the experiment, their brains were sending messages to other parts of their bodies. What were some of those messages?

Recording the results

- As each subject completes the test, his or her reaction time is expressed in seconds on the screen. Write the results in your Inventor's Notebook or on a sheet of paper.

Hypothesis: Girls will have faster reaction times than boys.

Subject	trial 1	trial 2	trial 3	Avg.
Jane	4	5	5	4.67
Sarah	5	4	4	4.33
Girls group average (4.67 + 4.33) ÷ 2 = 4.50				
Robert	4	4	4	4.00
Tony	5	5	5	5.00
Boys group average (4.00 + 5.00) ÷ 2 = 4.50				

Drawing a conclusion

- Based on your data, was your hypothesis correct? In the example data, girls and boys have the same average response rates: therefore, our hypothesis seems incorrect. The fact that girls tend to mature faster than boys does not seem to affect reaction time. What other explanations could there be? Perhaps the sample size (2 girls/2 boys) was not large enough to draw an accurate conclusion. Perhaps maturity doesn't affect reaction time at the subjects' age level, or other factors (height, general health, degree of alertness, etc.) are more significant influences on reaction time.
- Formulate a new hypothesis to test your conclusions.
- Write your new hypothesis in your Inventor's Notebook or on a sheet of paper.

Girls and boys have the same response rate to a visual stimulus (light); therefore, they will have the same response rate to other stimuli.

Challenge

Vision is only one of your senses. Do you think reaction time will differ in a test of hearing? Change your experiment design so that you can test an auditory stimulus rather than a visual stimulus.

Etchings: *LogoWriter* to *MicroWorlds*

by John Gough

I recently came across Web-A-Sketch, an entertaining Internet site on the World Wide Web (<http://www.digitalstuff.com/web-a-sketch/>) that offers simple instructions and an interactive graphics screen. By pointing and clicking with the computer mouse, you can make a drawing of connected line segments that join from point to point wherever you click.

This idea was based on the popular children's toy "Etch-A-Sketch" (from Mattel), which has two knobs. Rotating one knob moves a "graphic cursor" vertically, drawing a line; turning the other knob moves the cursor horizontally. Turning both knobs simultaneously combines the two directions of movement to produce diagonals and even curves. Web-A-Sketch on the Internet can make only straight lines, but is otherwise similar to this toy.

A project for Logo

Probably the mouse clicking was the mental trigger for me: I suddenly thought this would be fun to do with *LogoWriter*. Because I am still exploring *MicroWorlds*, I thought I would try it first with *MicroWorlds*, believing that any *MicroWorlds* procedures would run, with simple modifications, in *LogoWriter*. But I immediately ran into several interesting difficulties.

How could I program the mouse in *MicroWorlds*?

My Logo experience began a little before the introduction of *LogoWriter*. Soon I began to feel confident with *LogoWriter* on either Apple IIe or IBM—that is, versions of *LogoWriter* with no mouse. A little later I had great fun exploring the Logo-like language HyperTalk for programming in Macintosh's *HyperCard*, which uses the mouse a great deal. Next came Macintosh *LogoWriter* with a mouse, about 250 colors, a microphone, and so on. Perhaps I have been a bit slow to extend my keyboard-programming skill with *LogoWriter* to include mouse programming

in the Macintosh version. But I felt confident I could make a *LogoWriter* mouse do more or less what I wanted, at least at the simple level of Web-A-Sketch's point-and-click, even though I knew *LogoWriter*'s mouse was extremely primitive compared to *HyperCard*'s ultrasensitive little mouse. But *MicroWorlds*?

Encountering a mouse problem

The only *MicroWorlds* mouse-related primitive, **mousepos**, reports the current position of the mouse on the screen. I have not (yet) found a *MicroWorlds* equivalent of the *LogoWriter* primitive, **button?**, which reports true if the mouse has been clicked and false if the mouse has not been clicked. But this is the essential ingredient in any point-and-click: We want to know *where* the mouse is currently located after moving (or "pointing") and we want to know *when* the mouse has been clicked. Contrast this with the extremely sensitive mouse of *HyperCard*, which sends signals about whether the mouse button is up, down, or still down; whether the mouse is entering, leaving, or still in a location; and so on.

The *MicroWorlds* mouse can do virtually nothing except manipulate ordinary screen objects such as the cursor and menu bars; click on buttons, text boxes, and tool palette options; and drag on slider bars. It cannot be programmed. Let me put this another way: A mouse click is not an event that *MicroWorlds* recognizes. So, no "Web-A-Sketch" in *MicroWorlds*.

Reverting to *LogoWriter*, it was easy to make a point-and-click, dot-to-dot drawing program that uses the mouse. Notice in **checkpos** that clicking anywhere below a y-coordinate of -80 stops the drawing, and clicking anywhere to the right of an x-coordinate of 200 erases the current drawing. The intermediate procedure **begin** lets the user choose any starting point, and from then on **continue.on** lets the user click anywhere, with the turtle drawing from point to point as the user moves and clicks the mouse.

```

to start.sketch
  pu setpos [-100 -95]
  pd label [Click here to stop]
  pu setpos [200 0]
  pd label [ERASE!]
  pu setpos [-200 90]
  pd label [Move the mouse and click to draw
    lines]
  pu setpos [0 0]
  begin
end

to begin
  ifelse button?
    [checkpos make "pos1 mousepos
      pu setpos :pos1 continue.on]
    [wait 5 begin]
end

to continue.on
  ifelse button?
    [checkpos pd setpos mousepos]
    [wait 5]
  continue.on
end

to checkpos
  if (last mousepos) < -80 [stopall]
  if (first mousepos) > 200 [cg begin]
end

```

This *LogoWriter* program can be extended by specifying locations (a *LogoWriter* equivalent of a "button") where the mouse can be clicked to make the turtle lift or lower the pen, change pen color, make selected letters appear at the place clicked, or print the resulting picture.

An alternative solution?

Having been blocked in a mouse-driven point-and-click drawing program for *MicroWorlds*, I wondered if I could draw in Etch-A-Sketch style using arrow keys instead of the mouse. But again there were difficulties. Easily written in *LogoWriter*, the following procedures allow Etch-A-Sketch drawing with keyboard input using arrow keys.

```

to etch
  cg cc
  show [Use arrow keys, U for up, D for down,
    P to print, Q to quit]
  sketch readchar
end

to sketch :input
  if 28 = ascii :input [setx xcor - 2]
  if 29 = ascii :input [setx xcor + 2]
  if 31 = ascii :input [sety ycor - 2]
  if 30 = ascii :input [sety ycor + 2]
  if :input = "U [pu]
  if :input = "D [pd]
  if :input = "Q [stopall]
  if :input = "P [printscreen stopall]
  sketch readchar
end

```

Incidentally, the turtle begins drawing from the centre of the screen. It would be possible to incorporate a version of **begin** that would let the user shift the turtle "cursor" to a starting point, then begin drawing from that point. Alternatively, pressing U, then using arrow keys to move, then pressing D, moves the turtle to any desired starting point without drawing.

The following short procedure **check.a** helped me find the ASCII codes I needed for the arrow keys.

```

to check.a
  make "which readchar
  print ascii :which
end

```

In *LogoWriter*, virtually all the keys are individually accessible through either **key?** or **readchar**. The use of ASCII codes allows access to keys that do not normally type text. The mouse, as a programmable object, sends messages to the command **button?** in the same way that the keyboard sends messages to the command **key?** or **readchar**.

Reading characters in *MicroWorlds*

Despite the similarities between *LogoWriter* and *MicroWorlds*, things were not immediately so easy in *MicroWorlds*. I had deliberately made the arrow-key-driven *LogoWriter* program using primitives I knew existed in *MicroWorlds*, replacing **rg** and **print**, which I would normally have used, with **cg** and **show**. After careful cutting and pasting from *LogoWriter* to

MicroWorlds flipside, the program, quite simply, refused to run in *MicroWorlds*. These *LogoWriter-MicroWorlds* procedures try to do Etch-A-Sketch drawing with keyboard input using arrow keys. But the program stops when it gets to the **sketch readchar** line.

Why? It seemed that **readchar** was refusing to work. I looked at the Vocabulary entry for **readchar** in the menu bar option Help, but at first was none the wiser. I looked for alternatives, such as using the built-in *MicroWorlds*' dialog box commands **question** and **answer**, which allow a user to type any text input. I discovered that in *MicroWorlds*, the command **question** is not sensitive to nonalphanumerics, such as arrow keys. When **check.alphanumerics** runs, it is impossible to get any response when pressing nonalphanumeric keys, such as arrow keys, the Tab key, the Return key, and the Space bar.

```
to check.alphanumerics
question [Type a character, and press
Return]
make "which first answer
show :which
show ascii (first :which)
end
```

The procedure **check.ascii**, a *MicroWorlds* version of **check.a**, was no more successful.

```
to check.ascii
show [Press any key]
make "which readchar
show :which
show ascii :which
end
```

I even pasted **getinfo** from the Help Vocabulary entry for **readchar** and it didn't work. The gnashing of teeth was very audible.

```
to getinfo
show [Type your choice]
make "answer readchar
show :answer
end
```

In desperation I tried to make sense of the fine print in the online Vocabulary Help for **readchar**:

You must click on the background of the page (outside of a text box, the Command

Center, or Procedures page) for **readchar** to recognize the character typed.

So I tried running **getinfo** again, reached the point where **show** told me to "Type your answer," moved the mouse to the upper part of the screen and clicked, and then typed my choice—BINGO!

An awkward solution

The command **readchar** *does* work in *MicroWorlds*, but only if you click first with the mouse in the Front of the page. That is, you can't just run the command **readchar**. You have to shift the current location of activity from the Flipside, the Command Center, a textbox, or anything else to the Front of the page. By contrast, in *LogoWriter*, **readchar** works as soon as a key is pressed, regardless of where things were most recently active. So the instruction provided by the line

```
show [Type your choice]
```

should really read

```
show [Click near the turtle, then type your
choice].
```

Requiring a user to click before typing, seriously interferes with smooth, simple user input. It is at least as potentially disruptive as requiring a user to press the Return key after typing any input, as *LogoWriter* requires for the commands **readlist** and **readlistcc**. It also makes the Command Center unavailable for user interaction, and, similarly, textboxes and other objects that should be simply keyboard reactive.

```
to draw.keyboard
cg show [Click near the turtle. Use arrow
keys to move]
pd draw
end
```

```

to draw
make "input readchar
if 30 = ascii :input [sety ycor + 5]
if 31 = ascii :input [sety ycor - 5]
if 29 = ascii :input [setx xcor + 5]
if 28 = ascii :input [setx xcor - 5]
if :input = "U [pu]
if :input = "D [pd]
if :input = "S [stopall]
draw
end

```

Finally I used the click-first *MicroWorlds* version of **readchar** to create a procedure that mimics *LogoWriter*'s primitive command **readlist**, allowing a user to type some input, which is then reported as a list.

```

to readlistMW
show [Click near the turtle. Type a message.
Press Return at the end.]
make "thislist []
output r1
end

to r1
make "input readchar
ifelse 13 = (ascii :input)
[output :thislist]
[make "thislist sentence :thislist
:input]
r1
end

```

I am still struggling to cope with my personal transition from *LogoWriter* to *MicroWorlds*, feeling again and again that valuable and simple *LogoWriter* features that offer flexibility and efficiency in programming have been bungled in the making of *MicroWorlds*. *MicroWorlds* offers many hypermedia-type features within a Logo environment, but it is not simply *LogoWriter* bigger and better. And for the work I do—teaching programming in a Logo environment at simple and advanced levels—I am not yet convinced that *MicroWorlds* is an improvement. The lack of mouse programming and the clumsy click-and-type version of **readchar** both show this clearly. ▲

John Gough lectures in mathematics and computer education (*LogoWriter* and *HyperCard* especially) at Deakin University and is particularly interested in connections between mathematics and language. He can be reached at Deakin University SDS, 221 Burwood Hwy, Burwood, Victoria 3125, Australia. E-mail: jugh@deakin.edu.au

[Editor's Note: I contacted the technical department at LCSI about the missing **button?** primitive and the functioning of **readchar** and received the following reply explaining their design decisions.

Our design department decided to get rid of the **button?** primitive, for the following reasons:

- a) They say that you can do everything you used to do with **button?** by detecting clicks on buttons, turtles and colors.
- b) It's hard to figure out what happens when using **button?** in a mouse-driven environment. When the user clicks the mouse, say on a slider, and the **button?** is pending, do you activate the slider? And will **button?** output true?

In response to my query about **readchar**:

Yes, you still have to be outside of an editable area for **readchar** to work, and for the same reason as b) above. When in a text box, should **readchar** still show the key pressed?]



The Artillery Battle of Baltimore

by Jim Muller

Undoubtedly the most difficult job I have ever faced is trying to stay one step ahead of a child's imagination. This is what led the students with whom I worked into such subjects as geography, history, and science.

We all use the turtle to draw just about everything. But did you ever stop to think that for the turtle to draw, it has to be put in motion? Maybe we can learn something from how the turtle moves, as well as from what it does when it is moving.

The Battle of Baltimore caught the attention of different age groups. Younger children got caught up in the shape of Baltimore's Fort McHenry as described in "The Historical Turtle" in this issue. Some older students wondered why the British fleet had so much trouble moving beyond the small, rather poorly defended fort that guarded Baltimore Harbor.

What was the problem?

In trying to answer this, we got the turtle involved with not only the history of the battle, but also the physics of artillery fire control. Studying fire control led to the development of artillery games, but that's another story.

The British ships had more guns than the fort. They also had longer range guns. They could stay outside the range of the American guns and yet still reach the fort. You would think this would give them a decided advantage. But when you look into the problems of aiming a cannon on a rolling ship and trying to hit a small target some distance away, the problems become more real. And then there was the wind, which added to the confusion.

Today we have computers to do these precise calculations for us. In fact, it was this type of calculation that led to the development of ENIAC, the first digital electronic computer, which recently celebrated its 50th anniversary. The U.S. Army spent many hours trying to figure out

artillery fire control equations during World War II. Two professors at the Moore School of Engineering at the University of Pennsylvania designed the Electronic Number Integrator And Computer to do those calculations in seconds. However, until a Princeton University professor, John Von Neumann, came up with the concept of programmable memory, ENIAC had to be rewired every time it was to calculate another equation. So it's difficult to say just how much time it really saved.

One hundred and thirty-two years before ENIAC, the British gunners had to guess when firing their cannon. How far away were they? How hard was the wind blowing? How high should they aim? How much was the ship rolling? How much gunpowder should they use?

Cannon fire physics

The best we can do in a short article is to gloss over cannon fire physics. You might want to look into the laws of motion and how these can be represented through turtle motion.

To hit a target with a cannon ball, you need to know the velocity of the ball, the angle at which it was fired, and the impact of gravity on the ball as it moves through its trajectory. These three variables can be used in the following procedures.

```
to fire
cs ht
water
print [How many pounds of powder for this
      round?]
make "pwr readnum * 5
print [What's the angle?]
make "angle readnum
cannon :pwr :angle 1
end
```

The first task is to draw the terrain. In this case it is flat, because we are firing over water.

```
to water
pu setxy [250 -50] pd
setxy [-250 -50]
end
```

In different versions of Logo, **setxy** takes a list as input. In others, it takes the two values, *x* and *y*. **Setpos** is another primitive that does essentially the same thing and can be substituted where needed.

The next task is to specify how much powder is to be used. In reality, we need to define a variable that specifies the velocity of the round as it moves toward the target. If you translated real velocities into pounds of powder, you'd blow the cannon apart. Therefore, we're using a somewhat arbitrary variable to represent more accurately the amount of powder required.

The following procedure can be used in those versions of Logo that don't already have a **readnum** or **readnumber** primitive.

```
to readnum
op first readlist
end
```

The **cannon** procedure works with the **trajectory** procedure to move the cannonball through space. It combines the velocity of the cannonball, the angle at which it was fired, and the gravitational constant. Because the British ships were at sea level, the gravity is considered to be 1. You can, of course, alter that variable for other situations.

```
to cannon :pwdr :angle :grav
make "sx :pwdr * cos :angle
make "sy :pwdr * sin :angle
traject :sx :sy
end
```

The trajectory procedure, **traject**, moves the turtle through space.



```
to traject :sx :sy
setxy list xcor + :sx ycor + :sy
if ycor < -40 [stop]
traject :sx (:sy - :grav)
end
```

Exploring friction

All of this is very nice for a physics experiment. But it's not very realistic. The British also had to figure on the wind factor as well as the impact of friction as it moved through the air.

What does air have to do with this? Take a piece of cardboard and hold it facing in the same direction that you are facing. Next, try to push that cardboard forward. You can feel the resistance of the air. That's friction!

Now turn the cardboard so that the edge is facing the same direction that you are facing. It moves through the air more easily this time because of the lack of friction holding it back. The surface of the edge is much smaller than the surface of the flat side.

So let's add a friction factor to the **fire** procedure to make this a bit more realistic.

```
to fire
cs ht
water
pr [How many pounds of powder for this
round?]
make "pwdr readnum * 5
pr [What's the angle?]
make "angle readnum
make "fric 0.05
cannon :pwdr :angle 1
end
```

Then we have to change the trajectory procedure to include the friction variable. In this case, we're making the reasonable assumption that friction is proportional to velocity.



```

to traject :sx :sy
setxy list xcor + :sx ycor + :sy
if ycor < -40 [stop]
traject (:sx - :fric * :sx) (:sy - :fric *
:sy - :grav)
end

```

Other enhancements

Because you've now added an air friction factor, why not also add a wind factor that would blow the cannon ball to the left or right in proportion to its velocity?

You've drawn the terrain as water. To get closer to the problems the British had to face, why not draw the fort over to the right and establish target coordinates? Then add a conditional test to the **traject** procedure to see if the cannonball crosses those coordinates.

Add a rolling ship factor. As the ship rolls, the angle of the cannon changes slightly. Add commands that randomly change the angle within a small range to simulate a ship rolling in a choppy sea.

The guns of Fort McHenry were well entrenched in the corner fortifications. Protected by stout earthen walls covered with heavy stones, they were extremely difficult to hit.

On the other hand, the Star-Spangled Banner was a huge 50-foot flag with 15 stars and 15 stripes. You can call it luck, providence, or whatever you like. Even though it was heavily damaged, it proudly flew through the dawn's early light on that September morning in 1814. It now hangs in the Museum of History and Technology in Washington, DC. ▲

Jim Muller has had a lifelong interest in translating various technologies into understandable and persuasive programs. In 1981, Muller and his son organized the first Logo users group, the Young Peoples' Logo Association, which eventually grew into a worldwide 6,000 member organization. Among other things, the YPLA published *1,2,3 My Computer and Me* and *The Turtle's Sourcebook*. Both of these books are now being expanded for use with current versions of Logo. In 1985, the YPLA merged with CompuServe, where it became The Logo Forum. Today, Muller is a computer training and marketing consultant in the Dallas/Fort Worth metroplex. You can reach him via e-mail at 76703.3005@compuserve.com or on CompuServe at 76703,3005.



December 5–8, 1996

Tampa, Florida USA

Monterrey, Nuevo León Mexico

International Society for Technology in Education (ISTE) will continue to partner with the *Florida Association for Computers in Education (FACE)* and will add the participation of a second site in Monterrey, Mexico via telecommunications for the Fifth International Conference on Telecommunications in Education.

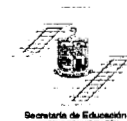
The world of telecommunications continues to grow dramatically, especially in the field of education. This exciting conference will give leaders in educational telecommunications/multimedia a forum in which to learn about rapidly advancing developments in telecommunications technologies. Be one of the thousands of educators, policymakers, and researchers who will join together in this unique global forum to exchange the latest in telecommunications, ideas, techniques, strategies, and policy concerns.

To receive an Advance Program, please contact:

International Society for Technology in Education
Tel•Ed '96
1787 Agate Street
Eugene, OR 97403-1923
Phone: 541/346-2411 Fax: 541/346-5890
Internet: Lori_Novak@ccmail.uoregon.edu

If you have Internet access and World Wide Web software, point to URL: <http://isteonline.uoregon.edu> for information on Tel•Ed.

For information on exhibiting at Tel•Ed '96 contact
Paul Katz, Exhibits Manager
Phone: 541/346-3537
Internet: pdkatz@oregon.uoregon.edu



The Last LX Column: A Retrospective Review

by Glen L. Bull and Gina L. Bull

Fifteen years ago a graduate physics major approached us about a new educational newsletter that he proposed to publish titled *The National Logo Exchange*. Tom Lough asked whether we would be willing to write a column for *Logo Exchange*. Volume 1, Issue 1 of *LX* appeared in September 1982. In the first issue Tom described his goals:

We are very excited by the potential represented by Logo. In the *Logo Exchange* we will have columns to help you develop Logo plans and activities. Important features will include articles written by classroom teachers and parents reporting on their Logo activities and discoveries. We also hope that you will be willing to share your experiences and lessons in Logo with your fellow readers.

The first issue contained a column, "Tips for Teachers," by Stephen Tipps; an article, "Logo Reinforces Geometric Skills," by Linda Nix, then a fifth-grade teacher in Austin, Texas; and our first column, titled "Design of a Language."

When Tom approached us we had no inkling that Logo would play such a central role in our lives. For a decade and a half we have been privileged to explore unending facets of Logo as it evolved. We lost count after the number of columns exceeded 100, but cumulatively we have written more about Logo than any other educational computing topic that we have explored. The unending series of explorations possible in Logo reflects the flexibility of the language. Like LEGO bricks, a few basic primitives can be reassembled in countless ways.

Our favorite titles

This flexibility is reflected in the range of titles about Logo that line our shelves. Some of our favorite titles over the years include *Teaching With Logo* by Dan and Molly Watt, *Beyond Turtle Graphics* by David Thornburg, *Visual*

Modeling With Logo by James Clayson, *Computer Science Logo Style* by Brian Harvey, *Special Technology for Special Children* by Paul Goldenberg, *Turtle Geometry* by Abelson and diSessa, and *Exploring Language With Logo* by Goldenberg and Feurzig. If we were stranded on a desert island with a (solar-powered) laptop computer, these books would be on the short list that we would want to take with us. Seymour Papert's book that introduced Logo to the world, *Mindstorms*, would need to be included as an essential reference as well. Topics covered in this selection range from mathematics to art, from computer science to special education, and from science to language arts.

Logo beginnings

Our first experience with Logo came with the TI-99 microcomputer. Our TI-99 had a tape drive instead of a hard disk, but the potential was intoxicating. We attended a workshop offered by Dan and Molly Watt where we were introduced to a prototype version of Logo for the Apple II. We were hooked.

Logo has proven to be three entities that share a single name. Logo is:

1. a programming language,
2. an educational philosophy, and
3. a culture that constitutes a shared community of ideas.

The programming language has continued to evolve over time. A shelf in our basement contains row after row of Logo manuals for dozens of different dialects and versions. Many of them are for computers that no longer exist: the Atari, the Texas Instruments TI 99-4, the Commodore PET, the Radio Shack TRS-80, and others. The earliest manual, published by Bolt, Beranek, and Newman, describes a version of Logo that predates turtle graphics, developed before the appearance of microcomputers with graphic displays.



The addition of the Logo turtle was an important invention that has come to symbolize the language. Seymour Papert describes another important characteristic of the Logo programming language in his book *Mindstorms: Computers, Children, and Powerful Ideas*. Logo incorporates the ability to easily define new commands. At that time, this capability distinguished Logo from any other educational programming language. Bill Atkinson, designer of the first popular hypermedia program, *HyperCard*, notes that his design was influenced by Logo—and it is notable that HyperTalk, the *HyperCard* programming language, also provides an easy means of defining new commands.

Logo's continuing evolution

The continuing evolution of the language is both a joy and a frustration. No one would want to use the first version of Logo today, printing turtleless (if that is a word) commands on a paper teletype. Revisions to incorporate new technologies are essential. The latest versions of Logo take full advantage of multimedia, a graphical user interface, mice, and windows. We have noted an interesting convergence between Logo and multimedia. The latest version of *MicroWorlds*, the flagship product of Logo Computer Systems, Inc. (LCSI), offers a range of multimedia capabilities. In comparison, Roger Wagner's *HyperStudio*, arguably the most popular hypermedia program in schools today, includes the programming language HyperLogo. One is a very good Logo with hypermedia capabilities, while the other is a very good multimedia program with Logo programming capabilities. The next step in the evolution will probably include links to the World Wide Web.

We would never advocate turning back the clock and, as a practical matter, it would not be possible even if it were desirable. However, the range of variation in newer versions of Logo has created an unavoidable Tower of Babel that has made it more difficult to share ideas and programs.

There is one area in which the programming language has probably not kept pace with its original promise—links to the external world. Packages with microcomputer-based laboratory

(MBL) interfaces, such as the Brøderbund *Science Toolkit*, demonstrated some time ago that inexpensive and affordable kits are feasible. Products such as *LEGO® TC logo* have demonstrated the power of this capability when linked to Logo. However, the versions of Logo provided with products such as *LEGO TC logo* and its successors have thus far only been subsets of a reasonably complete Logo language. Also, there has never been a relatively affordable version comparable to the *Science Toolkit*. Both factors have prevented this marriage of Logo with the real world from receiving the widespread attention and use that it deserves.

Evolution of the underlying educational philosophy that Logo represents has been as important as the language. The August 1982 issue of *Byte* magazine was devoted to the Logo programming language, but the cover of the November 1982 issue of the *Computing Teacher* (now *Learning and Leading With Technology*) focused on "Logo: A Learner-Driven Language." Logo helped define and frame the contrast between learner-centered approaches to educational computing and computer-assisted instruction.

Parting thoughts

Logo as a culture and a community of ideas has probably been the aspect that has had the greatest impact on our lives. Many of these ideas, such as juggling (see page 105 of *Mindstorms*), have little apparent connection to the programming language but have become embedded in the community of ideas. A learner-centered approach inevitably involves collaboration and sharing.

The title "Logo Exchange" was selected by Tom Lough to symbolize a vision of sharing driven by teachers, parents, and educators. This year *Logo Exchange* is returning to a renewed emphasis on an exchange of reports contributed by readers. This seemed an appropriate time to bring a close to our regular *LX* column. Our original commitment was for a single year of columns. We have exceeded that by a considerable margin.

Even though we won't be writing a regular *LX* column for the first time since its inception, we will still be continuing ongoing Logo explorations from time to time. Our son Stephen (named

for Steve Tipps, author of the original "Tipps for Teachers" *LX* column) received a train set for his thirteenth birthday this week. The layout that he is constructing has a number of switches that control and schedule the route of the train. This looks like a classic Logo problem to us—perhaps a Logo simulation of the routes will be in order. It even appears that it might be possible to control the switches themselves with an adaptation of *LEGO TC logo*. We are also experimenting with a comparison of fractal programs developed on graphing calculators and in Logo. There are some interesting parallels. And we are also experimenting with ... well, as we noted at the beginning of this our last *LX* column, the variations are endless. Perhaps in the future we will be able to provide an update of the more interesting outcomes. In the meantime, we will be looking forward to the descriptions of *your* explorations that we hope to read about in *LX*. ▲

Glen Bull is an associate professor in the Instructional Technology program of the Curry School of Education at the University of Virginia. **Gina Bull** is a computer systems engineer in the Information Technology and Communication organization at the University of Virginia. Internet Addresses: GBull@Virginia.edu, Gina@Virginia.edu

References

- Abelson, H., & diSessa, A. (1980). *Turtle geometry*. Cambridge, MA: The MIT Press.
- Clayson, J. (1988). *Visual modeling with Logo*. Cambridge, MA: The MIT Press.
- Goldenberg, E.P. (1979). *Special technology for special children*. Baltimore, MD: University Park Press.
- Goldenberg, E.P., & Feurzig, W. (1987). *Exploring language with Logo*. Cambridge, MA: The MIT Press.
- Harvey, B. (1985). *Computer science Logo style*. Cambridge, MA: The MIT Press.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.
- Thornburg, D. (1986). *Beyond turtle graphics*. Menlo Park, CA: Addison-Wesley Publishing Co.
- Watt, D., & Watt, M. (1986). *Teaching with Logo*. Menlo Park, CA: Addison-Wesley Publishing Co.

Logosium '96

St. Anthony Park Elementary School
St. Paul, Minnesota
Friday, June 14, 1996

The third annual Logosium will be a day of Logo workshops, discussions, and presentations hosted by the world renowned St. Paul Logo Project.

Logosium provides an opportunity for Logo users, old and new, to share ideas and experiences. This year, Logosium is scheduled for the day after NECC, so you will have a leisurely opportunity to follow up on presentations and discussions that began during the conference. This year's focus is on presentations by students and for students. Topics will include hypermedia, turtle graphics, LEGO® Logo, action research, simulations, game design, and much more. An area will be set aside for children to work and play with LEGO and Logo.

Friday, June 14, 8:30 am-4:30 pm
St. Anthony Park Elementary School
2180 Knapp Street, St. Paul
\$55 for adults, \$5 for children
(must be accompanied by an adult)

For registration and hotel information contact:
NECC '96
1787 Agate Street
Eugene, OR 97403-1923
Ph: 503/346-2834
Fax: 503/346-5890
e-mail: necc@oregon.uoregon.edu
www: <http://isteonline.uoregon.edu/>

Logosium is sponsored by the Logo Foundation and ISTE's SIGLogo.



To Glen and Gina Bull:

from *LX* editors, past and present...

From the editor:

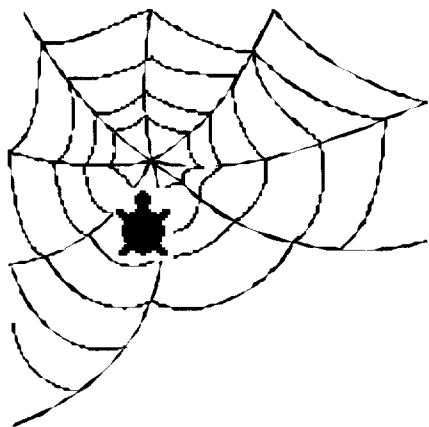
On behalf of *LX* and all its readers and contributors, I'd like to thank you for all your years of thought-provoking and insightful columns. We wish you both all the best in your future explorations, wherever they may lead you. You have an open invitation to report back to us on what you have been doing, what you have been thinking about, and what you have learned. We look forward to hearing from you!

Dorothy Fitch
LX Editor

From the former editor:

Glen and Gina Bull will be much missed by readers of *LX*. As the former editor of *LX*, I recall how much I enjoyed each new contribution they made to the Logo community. Not only were their articles on time, but they were accurate and well edited—characteristics that warm an editor's heart. Most of all, though, I value the breadth that Gina and Glen have given to Logo. They took their expertise from other areas of computing and curriculum and applied it to Logo. The result was often a bridge between Logo and other topics, such as hypermedia and robotics. Their bridges will be sadly missed.

Sharon Yoder
Former LX Editor



From the founding editor:

When I asked two University of Virginia professors named Glen Bull and Steve Tipps in December 1981 about a course they were offering for the spring term about a new computer language named Logo, I had no idea that it would change my life dramatically. About halfway through the course I got the idea that teachers were going to need the extra support that a newsletter could provide. So, I asked Glen and Steve to write a column each if I started a Logo newsletter. They looked at each other, grinned, and said, "Sure."

Ever since, Glen has been cranking out columns, making him one of the longest-running educational computing columnists in the world. His only "break in service" was in March 1983 when his son Stephen was born.

From that beginning, I have marveled at the depth and breadth of Glen's thinking and writing. The many visions he has shared over the years have given us an inkling of the power and potential of Logo. His uncommon concern for the classroom teacher and his enthusiastic embrace of new technology have been inspirational to me.

Although we will not enjoy a regular diet of Glen's columns in future *Logo Exchange* issues, I hope that you will go back into the archives and savor again the ones that he has contributed in years past. But I also hope that, in the coming months and years, Glen (and Gina, too) will occasionally send another column our way.

Glen, this heartfelt procedure is for you, from all of us.

TO GLEN
THANKS
FD 100
END

Tom Lough
LX Founding Editor

Logo: Search and Research

Parallel Programming in *StarLogo*

by Douglas H. Clements and Julie Sarama

Mitchel Resnick (1994) asks you to imagine a flock of birds swooping across the sky. They all follow the leader so perfectly...or do they? Most of us assume there is a "leader," but there is none. A flock is a "self-organizing system." Each bird follows the same simple local rules. The bird out in front just ended up there. As Resnick puts it, "The flock is organized without an organizer, coordinated without a coordinator" (p. 3).

Many systems—ants, highway traffic, market economies, and so on—are also determined by local interactions. Resnick's book is about such "decentralized" systems. It is also about exploring such systems in the massively parallel microworld he developed, called *StarLogo*. This is the new version of Logo we discussed in our last column.

Decentralized systems

Resnick investigates some large questions. Why do people resist thinking about decentralized systems? Why do we assume that all systems are centralized and controlled by leaders? How might tools be provided to help people think about decentralized systems?

To begin, Resnick shows that decentralization is a major trend. In politics, think of the USSR. In schools, think of school choice and the emphasis on local management. In technologies, think how the move from mainframes to microcomputers has become more decentralized in parallelism and in the Internet. In the sciences, think of new biological models and studies of artificial life. In psychology, think of movements from Freud's theories of the unconscious to development of theories of "societies of the mind" (Minsky, 1987). In research, think of the move from "objective," quantitative research to qualitative research with its multiple realities and decentralized knowledge.

How can we help people think about these new decentralized systems? If they're really all around us, won't we learn about them naturally? No. For thousands of years people observed flocks and birds and came to erroneous conclusions about the nature of the flocks. Observation and participation are inadequate. We need to engage in and design such systems.

Introducing *StarLogo*

For this reason, Resnick invented *StarLogo*. *StarLogo* has many of the features of his previous parallel-processing Logo, *MultiLogo*, but also goes beyond it in several ways. Whereas *MultiLogo* was concerned with a few processes acting in parallel, *StarLogo* is concerned about hundreds or thousands. Whereas *MultiLogo* was concerned with objects doing different things, *StarLogo* is concerned about objects doing similar things. Let's review the basics of *StarLogo*.

1. *StarLogo* has more turtles. Thousands of them. It is a massively parallel language, so all the turtles can perform their actions at the same time, in parallel. You need this: An ant colony with a couple of ants does not behave the same as one with hundreds of ants.
2. *StarLogo* turtles have better senses. They are "behavioral turtles." For example, they can "sniff" scents in their world. One primitive command makes them "follow a scent"; that is, turn in the direction where the scent is strongest. This lets them interact with each other and their world.
3. *The turtles' world in StarLogo is more complex and real.* The world is divided into small square sections called patches. The patches have many of the same capabilities as the turtles, although they cannot move. They can, however, hold information. For example, if a turtle releases a "chemical" as it moves over a patch, the patch can keep track of the



amount of chemical within its borders. Patches can also act. For example, they might grow food based on the level of that chemical.

Projects with *StarLogo*

Resnick worked with a group of a dozen high school students on many *StarLogo* projects. All these projects might be called “simulations.” However, they are not just computer models of a real-world situation, such as a nuclear reactor. They are explorations into decentralization.

For example, one of the projects was about slime mold. Slime-mold cells have a curious behavior. When food is scarce, they stop reproducing and move toward one another, forming a cluster with tens of thousands of cells. They act as a whole. The “they” act as a single “it.” It changes shape and begins crawling, seeking food. When it finds food, it breaks apart into many single cells. Until 1980, biologists believed that specialized cells coordinated this aggregation. But now they view it as a decentralized process. There is no leader, just local interactions among thousands of identical cells.

How do they get together? When each slime-mold cell moves, it emits a chemical. All cells are attracted to the same chemical. As they move, they follow the gradient of the chemical.

In *StarLogo*, each turtle (i.e., each “slime-mold cell”) was controlled by four demons: one made it move, a second added randomness, a third made it emit the chemical, and a fourth made it “sniff” for the chemical and turn in the direction where the “smell” was the strongest. Figure 1 provides illustrations of a *StarLogo* slime-mold program in progress.

What if you had the cells sniff more before moving? How will this improved sense of smell change the program? Will there be more clusters or fewer? Will they form more quickly or more slowly?

Answer the question first for yourself.

Most people think (as I did) that there would be fewer and bigger clusters. But, in fact, there are more and smaller clusters. Most people think that the cells are trying to get together, to combine. If they have a better sense of smell, they

will do a better job. What’s the flaw in our thinking? They are *not* “trying to get together.” Instead, they are following simple rules independently. They do follow the gradient more effectively. So, they form smaller clusters more quickly, which become (relatively) stable. Once again, we’ve imposed our ideas of intentionality and centralized thinking on a decentralized system.

There are fascinating projects involving ants, traffic jams, and even geometry! I could describe them, but nothing replaces your own experience with the program (see the postscript). They illustrate counterintuitive findings, such as how *adding* traffic lights on a straight road without intersections can help improve overall speed (throughout) and the many reasons why traffic jams form.

Back to the foreword!

Sounds great. But what’s this got to do with research?

In the foreword to the book, Seymour Papert describes its significance for research: “Resnick’s work provides a rare glimpse of what I am sure will become a new paradigm for research in education” (p. ix).

Papert argues that the critical bottleneck for research is not a shortage of machines or ideas, but the shortage of people like Resnick who are able to innovate fluently in an unprecedented combination of media and willing to take the risks inherent in doing so.

He also discussed two complementary fallacies: The “technocentric fallacy” is illustrated by questions like: “How do computers change the way children learn?” Papert says people change things, not computers, and computers can play different roles. But a too-easy rebuttal leads into a complementary fallacy: the “just-a-tool” fallacy. This fallacy does not distinguish between regular tools (reasonably described as “just tools”)—those that improve their users’ ability to do pre-existing jobs—and tools that are mind enhancers (i.e., that are more than “just tools” because of their role in the creation of a job nobody thought to do or nobody could have done before).

Papert says *Math Blaster* is certainly different from *Geometric Supposer*, but both are tools

Figure 1.
Simulation of Progression of Slime Mold using *StarLogo*

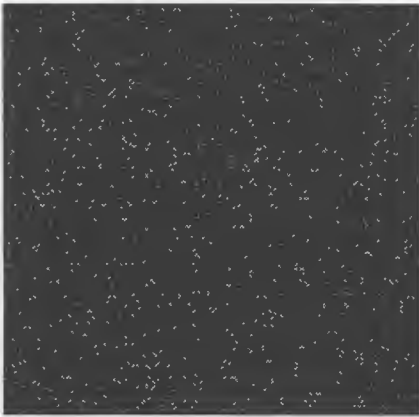


Figure 1a

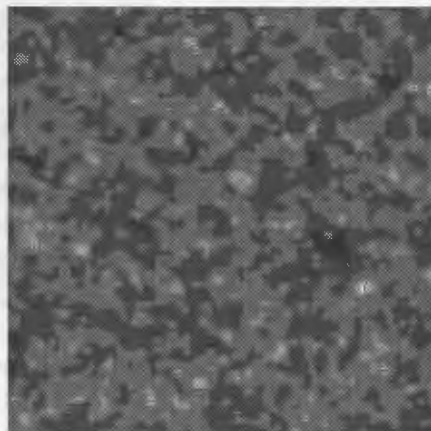


Figure 1b

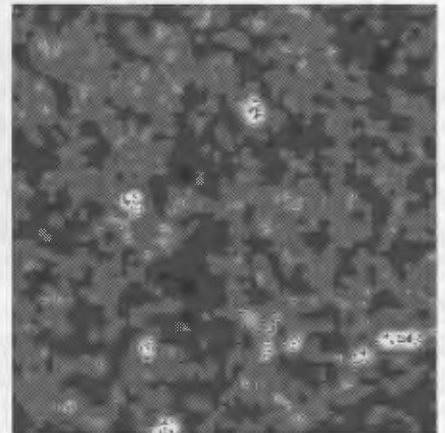


Figure 1c

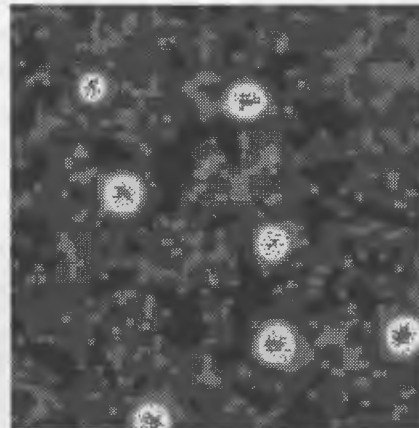


Figure 1d

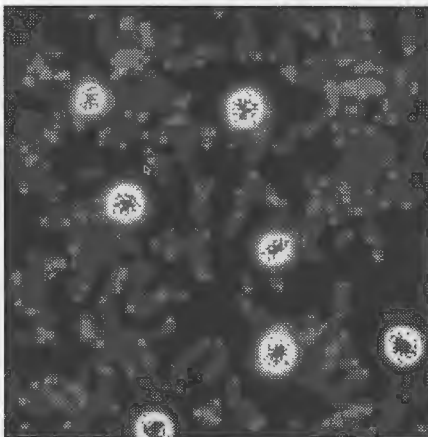


Figure 1e

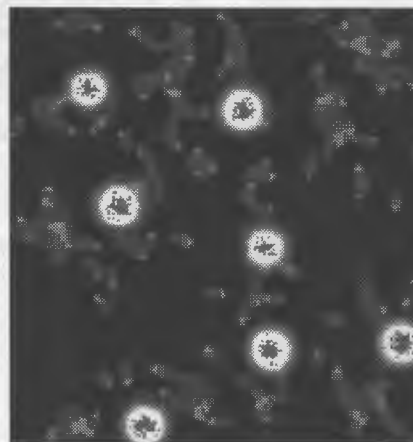


Figure 1f

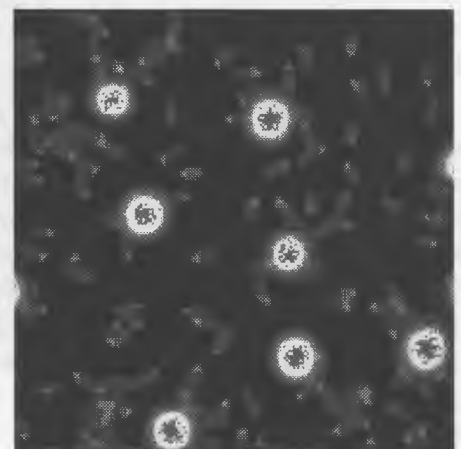


Figure 1g

for well-established educational purposes. *StarLogo*, in contrast, liberates students from the confines of the centralized mindset...and provides a tool to do it. It is a mind-enhancing tool.

Resnick's research with *StarLogo*

One basic question Resnick asks is: How did people become engaged with decentralized phenomena and the new ways of thinking entailed? Most people were fascinated by the view of self-organizing, decentralized systems as they were displayed on the computer screen. It almost seemed like "cheating." Frank (a high school student) said, "In this version of Logo, you can get more than what you tell it to do" (p. 131). Interesting, thinking about Lady Lovelace's original formulation that a computer could only be programmed to follow the commands of the programmer. You can know what each individual object will do, and yet have no idea what the overall system will do.

Resnick concludes the book with more reflections and vignettes on people struggling to understand decentralized systems. In so many cases, the design of *StarLogo* programs was based on an unquestioned assumption of a "leader," an "outside force," or a "seed for change." For instance, people would quickly assume that a radar trap might cause traffic jams or an accident. They predicted that without such outside forces, "there was nothing," so traffic would proceed smoothly. However, traffic slowdowns and jams emerge *even with no seed and no leader*.

Across of variety of areas, the simplest and often most accurate programs did *not* use leads or seeds. Instead, they used myriad interactions among objects or beings following a few simple rules. Centralized thinking seems to be a strong bias in our thinking.

People who thrived in the *StarLogo* environment tended to be those who relished, instead of resisted, unpredictability. One student's partner suggested that they give up on a decentralized approach to a termite program. Instead, he wanted to program the termites to go to predetermined spots. The student said, "Oh, that's boring!... You would almost know what to expect.... [The decentralized way, the termites]

made the piles by themselves. It was like they [the piles] were artificially put in" (p. 132).

Resnick found that five heuristic strategies emerged for thinking about decentralized worlds.

1. *Positive feedback isn't always negative.* Negative feedback, like a thermostat, often keeps things nicely in balance. Positive feedback, such as a screeching speaker too near a microphone, is often considered negative. However, in decentralized systems, positive feedback has many positive results. In the slime-mold program, for example, aggregates form because more chemical attracts more cells, which makes more chemical, and so on.
2. *Randomness can help create order.* People, naturally, think of randomness as the opposite of order. In self-organizing systems, though, randomness can help create the "seed" of order. The seeds of clusters emerge in the slime-mold project when a few cells happen to wander near one another. Then positive feedback takes over. Randomness also helps combine small clusters. When a cell breaks free, it might combine with a nearby cluster. Small clusters often disappear or grow bigger as a result. This is very important to ensure that the system grows to have optimally large clusters. In the ant project, randomness plays another important role. Ants will randomly "lose" their way off a trail to a food source. But in so doing, they often find a superior source.
3. *A flock isn't a big bird.* The notion of levels is very important to self-organizing systems. Interactions among objects at one level give rise to new types of objects at another level. Interactions among slime-mold cells give rise to clusters. Interactions among birds give rise to flocks. And the behaviors of the clusters and flocks is very different from that of cells and birds alone.
4. *A traffic jam isn't just a collection of cars.* In a similar vein, you can't think of a traffic jam as just cars. The cars' positions and speeds in traffic are changing dynamically.
5. *The hills are alive.* The impact and role of the environment should not be minimized. The

environment is not just a "setting" that is "acted upon." It is interacted with. It can be a communication medium. It can act, such as by making a chemical evaporate when it is no longer "needed."

These heuristics are certainly not prescriptive. But they may help people think about systems in new ways.

Tools like *StarLogo* and books like Resnick's are necessary. They help us break the centralized mindset that traps us into narrow ways of thinking.

Postscript

When I asked Mitch if there were any words he wanted to add, he wrote: "I can't believe you condensed my book into a couple of pages of your stagnant prose." No, that's not what he wrote! He hasn't even *read* our stagnant prose yet! But he did share the following:

There are also several articles about *StarLogo* accessible through my Web site (<http://www.media.mit.edu/~mres/>). The original version of *StarLogo* was implemented on the Connection Machine. We now have a Macintosh version of *StarLogo*. The Mac version is available free of charge. Anyone wanting a copy of the Mac version should write to StarLogo-request@media.mit.edu.

You can't beat the price. The book is also a bargain and a must-read for the intellectually curious, Logophiles, and everyone else. ▲

References

- Minsky, M. (1987). *The society of mind*. New York: Simon & Schuster.
- Resnick, M. (1994). *Turtles, termites, and traffic jams*. Cambridge, MA: The MIT Press.

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

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, mathematical, 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 is 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 LCSl.

Julie Sarama, PhD, is a research codirector of the NSF project at the State University of New York at Buffalo. She has taught secondary mathematics and computer science, gifted math at the middle school level, and mathematics methods courses. She is coauthor of *Turtle Math* and is currently designing and programming new versions of Logo.

Douglas H. Clements and Julie Sarama
State University of New York at Buffalo
Department of Learning and Instruction
593 Baldy Hall
Buffalo, NY 14260
E-mail: Clements@acsu.buffalo.edu
Sarama@acsu.buffalo.edu



Global Logo Happenings

Edited by Dennis Harper

Logo Exchange Continental Editors

Africa

Fatimata Seye Sylla
UNESCO/BREDA
BP 3311 Dakar
Senegal, West Africa

Asia

Marie Tada
St. Mary's Int. Sch.
6-19 Seta 1-Chome
Setagaya-Ku
Tokyo 158, Japan

Australia

Anne McDougall
Monash Univ.
6 Riverside Dr.
East Kew
Victoria 3120
Australia

Europe

Harry Pinxteren
Logo Cent. Nederland
P.O. Box 1408
BK Nijmegen 6501
Netherlands

Latin America

José Valente
NIED
UNICAMP
13082 Campinas
Sao Paulo, Brazil

Constructing Change in Schools Through Teacher Training in Logo

by José Armando Valente

Introduction

Logo is assuming the reputation of being a great methodology, but one that is very hard to implement in schools. One reason for this is that its implementation requires too much change in current administrative and pedagogical paradigms. Thus, the process of Logo inclusion should start with the training of the teachers. In most cases, courses to train teachers to use Logo consist of a series of disciplines that are taught outside the context in which the teacher works. The teacher, besides acquiring the information given, has to contextualize it to his or her particular working situation.

The training course described in this article uses a contextualized-constructionist approach, which offers an alternative to standard Logo training. A Logo learning environment is created in the working place and the activities developed by the teacher help him or her construct the knowledge about how to use Logo in this context. These courses, developed by professionals from NIED (Center of Informatics Applied to Education) at UNICAMP (State University of Campinas, Brazil), have been used in several schools.

Throughout the world, typical courses that prepare teachers to use Logo in their teaching

practice are done at the graduate level, continuing education courses, or even Logo workshops. They may require the teachers' physical presence or they can be done at distance, and to some extent they can help teachers learn about Logo. But they are decontextualized from the reality of school. In general, they consist of learning the Logo language; reading and discussing topics related to the Logo methodology; and reviewing a particular subject, such as math or science, and integrating it with Logo. All the courses we have developed in Brazil, such as FORMAR (Valente, 1993a) or the courses provided by the Centers of Informatics in Education (CIEd) used this model (Em Aberto, Ano XII, # 57, 1993). Much of the research on Logo teaching training also investigated this decontextualized approach (Alto, 1993; Mattos, 1992; Menezes, 1993; Silva Neto, 1992) and the international literature describing Logo teacher training courses in other countries, especially in the U.S., report similar decontextualized models (Estes & Thomas, 1993; Harper, 1989; Papert, 1986).

The problem with this decontextualized approach is that the courses or the activities developed are not related to the actual needs of the teachers or the specific situations in which the teachers are immersed. This means that the

teachers have to go back to their particular working places and be able to adapt the knowledge acquired to their working realities. For example, these courses do not create a situation in which the teachers learn how to use Logo with one student or with a class and how to deal with the administrative and pedagogical changes necessary to implement Logo. It is similar to the way in which long division is taught in the schools today. The children have no idea how to use it or if the answer they cranked out is reasonable or not. The application of what the teachers have learned in terms of Logo methodology to their particular teaching situations has to be done by the teachers; they are alone in this. And this task is not trivial.

What are the problems in delivering decontextualized Logo courses? First, to be able to apply knowledge to a particular situation requires another type of knowledge (Ackermann, 1990). It does not necessarily follow that if we know something, we know how to apply it. The application of knowledge has to be exercised so that we learn how to use that knowledge in different situations. Because the use of Logo in a teaching situation is not exercised in standard courses, teachers are not prepared to do this task. Second, the use of Logo in the teacher's teaching situation is done without support from their Logo trainers. Professors are not there with the teachers to help them go through the process of using Logo with their students. Third, the implementation of any pedagogical changes requires the involvement of others in their school (colleagues, students, and supervisors) who have to collaborate and be willing to embark on this new adventure. Without some changes in the teaching paradigm, Logo will not be able to effectively fit into current school systems.

The solution to the problem of decontextualization of knowledge is to develop courses that use a pedagogical approach tailored to teachers' needs and their working contexts. The educational changes that Logo requires cannot be seen as an electrical switch that the trained teacher activates when decontextualized knowledge has been acquired. These changes have to be constructed in the working place, as part of the teacher's training courses and implemented within

the course structure. This is what the contextualized-constructionist approach to Logo teacher training is proposing.

Contextualized-constructionist teacher training

The process of constructing knowledge is not new. Piaget has shown that everything we know was constructed through our own actions. However, as was mentioned in a previous *LX* article, the computer plays an important role in the process of knowledge construction. It allows the student to engage in the cycle of description–execution–reflection–debugging–description and becomes a very effective learning tool (Valente, 1993b).

When students program using Logo, their procedures are a description of the problem solution through the Logo computer language. The computer executes these procedures and presents a result on the screen. The students can look at the pictures that are constructed and reflect upon the procedures. This reflective activity can lead to either no action, when the students' original ideas correspond to the results presented, or debugging, when the results are different from what the students intended. Debugging means that the students have to look for new information to fix the previous solutions. That is when the students have the opportunity to acquire new knowledge or to restructure old knowledge. However, the debugging activity is facilitated by the existence of the computer program. The program is the students' description of their ideas in terms of a formal, precise, and simple language.

The objective of the training course using the contextualized-constructionist approach is to provide teachers with the ability to use computers in their classrooms and to help the students using computers to construct their knowledge. We want the teachers to know about constructionism, experience it, and know how to use it in their teaching practices.

However, these changes are not trivial for teachers to implement. Thus, the contextualized-constructionist approach to teacher training must provide teachers with the conditions to act, reflect, and debug their knowledge in all phases of



the implementation of Logo in the school: to learn the Logo language and methodology, to learn how to use Logo with a student, to learn to use Logo in a classroom, and to integrate Logo into their teaching practices. This means that the teachers and course professors must experience the same situations so that each can understand the other and provide solutions that are effective for the school reality.

The course starts only after the school has installed the computer laboratory. The reasons for this are twofold. First, the teacher training course should take place at the school because teachers need this access. Second, this is an indication of the level of commitment by the school administration.

The course consists of the following 5 modules.

1. *Introduction to Logo.* This module can take 30–40 hours. The teachers learn to program using the Logo language, reflect upon their own learning process, and read and discuss the basic literature about Logo methodology. After each programming session the course professor summarizes what was seen during that session through what is called a reflective activity. The objective of this activity is to formalize what the teachers used intuitively and to discuss the teachers' impressions as learners, the professor-teacher interactions, and theoretical topics. This reflective process is done based upon the experiences of the teachers and is very much contextualized.

2. *Use of Logo with a student.* The objective of this module is to provide the teachers with the experience of using Logo with a student. Each teacher selects one or two students and the teacher's role is to help the students learn to program the computer using Logo. Thus the students become the computer users and the teacher becomes the facilitator of this process. This activity takes approximately 20 hours and is supervised by the course professor. The professor observes the teacher's interaction with the students and these observations are discussed individually or with the group of teachers. The course professor also solicits the teacher's observations and reflections as to the student's learning process a) in terms of comparison with

his or her experience in the first module and b) in terms of the student's learning style. The theoretical material discussed in Module 1 now takes on new meaning.

3. *Advanced Logo.* After their experience with one or two students, the teachers have the opportunity to enhance their knowledge about the Logo language and methodology. In general, the experience with the students guides the teachers to select a particular topic (either the Logo language or Logo methodology) to explore in depth. This module can take 30–40 hours and the structure is similar to the Introduction to Logo module.

4. *Use of Logo with a class.* The objective of this activity is to provide the teachers with the experience of using Logo with a class. Each teacher works with one class and the students use Logo to solve problems or to develop projects as part of the subjects taught by the teacher. This activity takes 20 hours and is supervised by the course professor. The professor observes the teachers' performance and these observations are discussed with each teacher or with a group of teachers. Also, the professor may show activities and ways the teachers can integrate Logo into their respective teaching subjects. The discussion of theory and Logo methodology is similar to Module 2. This reflection is still based on concrete activities, but is contextualized to a broader realm of experiences.

5. *Elaboration of a pedagogical project.* Teachers develop individual projects describing how they are going to use Logo as part of their classroom activities. Each project is discussed with the course professor, presented to and discussed by all the other teachers, and proposed to the school administration. The collection of projects to be implemented by each teacher becomes the school's computers in education project.

Training that takes place at the school in which the teachers work presents several advantages for the teachers and the course professor. First, the knowledge acquired is contextualized. The teachers' familiarity with computers happens through the use of the school computers, operation system, and local network. The experience of learning and using Logo also happens at the school setting using the school

population as a means to exercise and construct the teacher's knowledge. Second, the teachers do not have to leave the school and stop their teaching. The course activities can be organized around teaching schedules. Third, the course professors can be more effective. They experience and come to understand the school's idiosyncrasies so that the pedagogical or administrative solutions proposed are based on the school's reality. This makes the proposed solution much easier to implement.

The teacher training course based upon the contextualized-constructionist approach allows the introduction of Logo into school activities to be the product of a construction process. This process implies the construction of pedagogical and administrative changes so that Logo can become part of regular school activities.

Results from this training experience

This training approach was used in one special school for physically handicapped children and is still in progress in several other regular schools.

In the special school, we worked with 14 teachers from kindergarten to 4th grade. These teachers were divided into two groups. One group of 6 teachers participated in the course activities in the morning and teaching activities in the afternoon. The other group of 8 teachers developed course activities in the afternoon and taught in the morning. The Introduction to Logo and the Advanced Logo modules courses happened intensively during vacation breaks planned by the school. Modules 2 and 4 took place during the academic year, at which time the course professors from NIED would go to the school once a week. Today, 13 teachers use computers in their classrooms. Each classroom has two computers that are shared among 7–10 students. The students use Logo and a word processor to develop their classroom activities with the help of the trained teachers.

In another regular school we started working with 45 people (35 teachers and 10 administrators), who took the Introduction to Logo module (25 hours). After this experience, 21 teachers from kindergarten to ninth grade started working with one or two students. This experience

was very brief because the students who were not involved in this activity demanded their rights to use the computers. After the 25-hour Advanced Logo module, the teachers started to use Logo with their classes. Our involvement with this school has taken one year and its evaluation has shown that the teachers need more experience in programming, in interacting with one student, and in integrating Logo into their classroom activities. However, all the teachers participating in the training course were able to use Logo with their students and some teachers were able to document some of the student's experience so they could describe it at a Logo conference. The rest of the training course and the debugging of the deficiencies observed will be implemented in the next academic year.

Conclusions

The Logo training course based on the contextualized-constructionist approach was an alternative to the Logo training course we had developed in the past. These traditional Logo training courses produced teachers with knowledge about the Logo language and techniques but with very little experience using this knowledge in a real classroom. The end result of this experience was that Logo was too hard to do and very difficult to implement as part of classroom activities. In these cases, either the teachers abandoned the use of Logo and shifted to the use of instructional software or the school administration contracted a Logo expert who stays at the computer laboratory working with the students on computer literacy, independently of what is happening in the classroom.

The use of the contextualized-constructionist approach as an alternative has produced better and more rewarding results. We are seeing Logo used by teachers as part of the classroom activities. However, this training course is much longer, more costly, and takes much more effort from the course professors. The involvement of both teachers and course professors has been much greater and more intense. Part of the cost and the professors' involvement can be minimized through the use of networking facilities such as e-mail. This facility will be implemented next year and we will see how effective it can be in



minimizing the physical presence of the course professor in the school.

Another important result is that even though the contextualized-constructionist training course has produced teachers capable of using Logo in their classroom, it does not guarantee the involvement of all the school's teachers. Not all teachers who start the course finish it. This was very striking because we thought we were helping teachers construct their knowledge in the use of Logo.

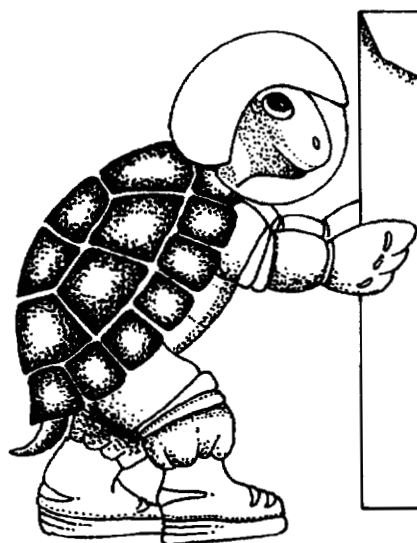
In our traditional Logo training courses, the majority of the teachers who started the course would finish it. However, they would go back to their respective schools and not be able to implement the knowledge acquired. One reason these teachers gave for this occurrence was that it was very difficult to convince their colleagues or administrators to implement Logo. This may be true. But through the contextualized-constructionist teacher training course we are discovering that there is more to this story. Most of the teachers who abandoned the contextualized-constructionist course did so because they lacked the knowledge about the structure of the subject they teach. Their activities with a student or with the class showed that they had great difficulty understanding what they teach, so they were incapable of changing the original curriculum to integrate Logo. If they cannot implement these changes they have a very difficult time understanding how Logo can be helpful as a learning tool and are unwilling to fight to change the existing pedagogical system. The easiest path for them is to leave the training course and continue using their traditional methodology. But it would be better if they would go back to undergraduate school to learn their subject! ▲

José Armando Valente

Núcleo de Informática Aplicada à Educação– NIED
Universidade Estadual de Campinas– UNICAMP
Cidade Universitária
Prédio V de Reitoria – 2º Piso 13083-970 Campinas,
SP – Brasil
Phone: ++55 (192) 39 7350
Fax: ++55 (192) 39 4717
E-mail: jvalente@turing.unicamp.br

References

- Ackermann, E. (1990). *From decontextualized to situated knowledge: Revising Piaget's water-level experiment* (Epistemology and Learning Group Memo #A 5). Cambridge, MA: Massachusetts Institute of Technology.
- Alto, A. (1993). *O computador na escola: O facilitador no ambiente Logo*. Tese de Mestrado, Departamento de Supervisão e Currículo da PUC, São Paulo, Brazil.
- Estes, N., & Thomas, M. (1993). Rethinking the roles of technology in education. In *Proceedings of the 10th International Conference on Technology and Education*. Cambridge, MA: Massachusetts Institute of Technology.
- Harper, D. (1989). *Logo: Theory and practice*. Pacific Grove, CA: Brooks/Cole Publishing Co.
- Mattos, M.I.L. (1992). *O computador na escola pública—Análise do processo de forma de professores de segundo grau no uso desta tecnologia*. Tese de Doutorado, Instituto de Psicologia da USP, São Paulo, Brazil.
- Menezes, S.P. (1993). *Logo e a forma de professores: O uso interdisciplinar do computador na educação*. Tese de Mestrado, Escola de Comunicar Arte, USP, São Paulo, Brazil.
- Papert, S. (1986). *Constructionism: A new opportunity for elementary science education*. Proposal to the National Science Foundation. Cambridge, MA: Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group.
- Silva Neto, J.C. (1992). *Capacitado de recursos humanos em informática educativa: Uma proposta*. Tese de Mestrado, Faculdade de Educação, UFRJ, Rio de Janeiro, Brazil.
- Valente, J.A. (1993a). Formação de profissionais na área de informática em educação. In J.A. Valente (Ed.), *Computadores e conhecimento: repensando a educação*. Campinas, Brazil: Gráfica da UNICAMP.
- Valente, J.A. (1993b). Computers in education: shifting the pedagogical paradigm from instructionism to constructionism. *Logo Exchange*, 12(2), 39–42.



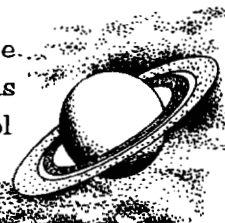
Push the envelope with MicroWorlds



Launch your students toward stimulating new classroom challenges and adventures. They can soar to a galaxy of new skills using MicroWorlds.

MicroWorlds is an exciting application that offers in a single program many of the features available in HyperCard, Kid Pix, and LogoWriter.

You're the pilot. Get ready now to grab the controls, blast off, and explore these exciting new worlds with your class! To help you prepare your crew for this fantastic voyage, Sharon Yoder and Dave Moursund back at mission control have developed a 222-page flight manual titled *Introduction to MicroWorlds—A Logo-Based Hypermedia Environment*.



As a computer application, MicroWorlds contains a wide range of features that make it easy and fun to work with color graphics, sound, text, and animation. As a programming environment, MicroWorlds includes a powerful and modern version of the Logo programming language.

It's time to broaden your universe. The countdown is underway. Have a nice trip!



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 Fax-on-Demand: 800/891-0212
 America Online: ISTE CompuServe: 70014,2117
 Internet: iste@oregon.uoregon.edu
 World Wide Web: <http://isteonline.uoregon.edu>

HYPERCARD® IS A REGISTERED TRADEMARK OF CLARIS CORPORATION, SANTA CLARA, CA, U.S.A. KID PIX® IS A REGISTERED TRADEMARK OF BRØDERBUND SOFTWARE, NOVATO, CA, U.S.A. LOGO®, LOGOWRITER® AND MICROWORLDS® ARE REGISTERED TRADEMARKS OF LOGO COMPUTER SYSTEMS, INC., HIGHGATE SPRINGS, VT, U.S.A.



ISTE Books & Courseware Order Form

To order ISTE products advertised in this publication, find the product title in the following list and enter it on the form below.

To receive a free Resource Guide with a complete listing of ISTE products and services, please call our toll-free number, 800/336-5191.

Product (• Indicates an ISTE-published title.)	Member Price	Nonmember Price
<i>Children's Machine—Rethinking School in the Age of the Computer, The</i>	13.00	13.00
• <i>Exploring Recursion With LogoWriter</i>		
<i>Student Book (Specify Apple II series, Mac, or MS-DOS data disk)</i>	10.75	11.95
<i>Teacher Manual (Specify Apple II series, Mac, or MS-DOS data disk)</i>	10.75	11.95
• <i>Grabbing With LogoWriter</i>	9.85	10.95
• <i>Introduction to MicroWorlds—A Logo-Based Hypermedia Environment</i>	25.15	27.95
• <i>Introduction to Programming in Logo Using Logo PLUS</i>	18.85	20.95
• <i>Introduction to Programming in Logo Using LogoWriter, 3rd Ed.</i>	24.25	26.95
• <i>Introduction to Programming Using Terrapin Logo for the Macintosh</i>	19.80	22.00
• <i>Logo Learning—Strategies for Assessing Content and Process</i>	12.55	13.95
• <i>Logo PLUS for Educators—A Problem Solving Approach</i>	11.70	13.00
• <i>LogoWriter for Educators—A Problem Solving Approach</i>	11.70	13.00
• <i>Math Activities Using LogoWriter (Specify Apple II series, Mac, or MS-DOS data disk)</i>		
<i>Patterns and Designs</i>	18.85	20.95
<i>Investigations</i>	14.35	15.95
<i>Numbers & Operations</i>	16.15	17.95
<i>High School Math</i>	18.85	20.95
<i>More Investigations</i>	14.35	15.95
<i>More Patterns and Designs</i>	16.15	17.95
<i>Probability and Statistics</i>	12.55	13.95
• <i>MicroWorlds—Hypermedia Project Development and Logo Scripting</i>	31.45	34.95
• <i>More Turtle Power—Text, Graphics, and Sound With LogoWriter (Specify Apple II series, Mac, or MS-DOS data disk)</i>	18.85	20.95
• <i>Turtle Power—Beginning Graphics With LogoWriter, Rev. Ed. (Specify Apple II series, Mac, or MS-DOS data disk)</i>	20.65	22.95
• <i>Turtle Time Designs—Working With Patterns in LogoWriter (Specify Apple II series, Mac, or MS-DOS data disk)</i>	11.65	12.95
• <i>Wacky Words for Traveling Tales—Logo Applications in Language Arts</i>	15.25	16.95

Receive an additional 18% discount when ordering 10 or more of the same title of ISTE-published products.

Name		Membership #
School/Business		
Address		
City	State	Zip/Postal Code
Country	Phone	

Shipping & Handling

\$0-\$15.99 (subtotal)	add \$4.50
\$16-\$45.99 (subtotal)	\$6.00
\$46-\$75.99 (subtotal)	\$7.00
\$76-\$100.99 (subtotal)	\$8.00
\$101 or more	8% of subtotal

GST Registration Number 128828431

Code LX4

ORDER

Quantity	Title	Member Unit Price	Nonmember Unit Price	Total Price

PAYMENT OPTIONS

☐ **Payment enclosed.** Make checks payable to ISTE—
International orders must be prepaid with U.S. funds or credit card.
☐ VISA ☐ MasterCard ☐ Discover Card Expiration Date _____

☐ **Purchase Order enclosed.** Please add \$4.00 for order processing—
P.O. not including \$4.00 fee will be returned.

☐ **C.O.D.** for U.S. book orders only. You will pay UPS the total upon delivery by check or cash—
ISTE will add \$4.50 order processing.

☐ **Airmail.** International orders for Books & Courseware are sent surface mail—
ISTE will bill you the additional shipping charge for airmail.

☐ Send me ISTE membership and subscription information.

☐ Send me a free ISTE catalog.

Deduct 18% on ISTE-published titles if ordering quantities of 10 or more of the same title

SUBTOTAL

SUBTOTAL

*Shipping and Handling (see box above)

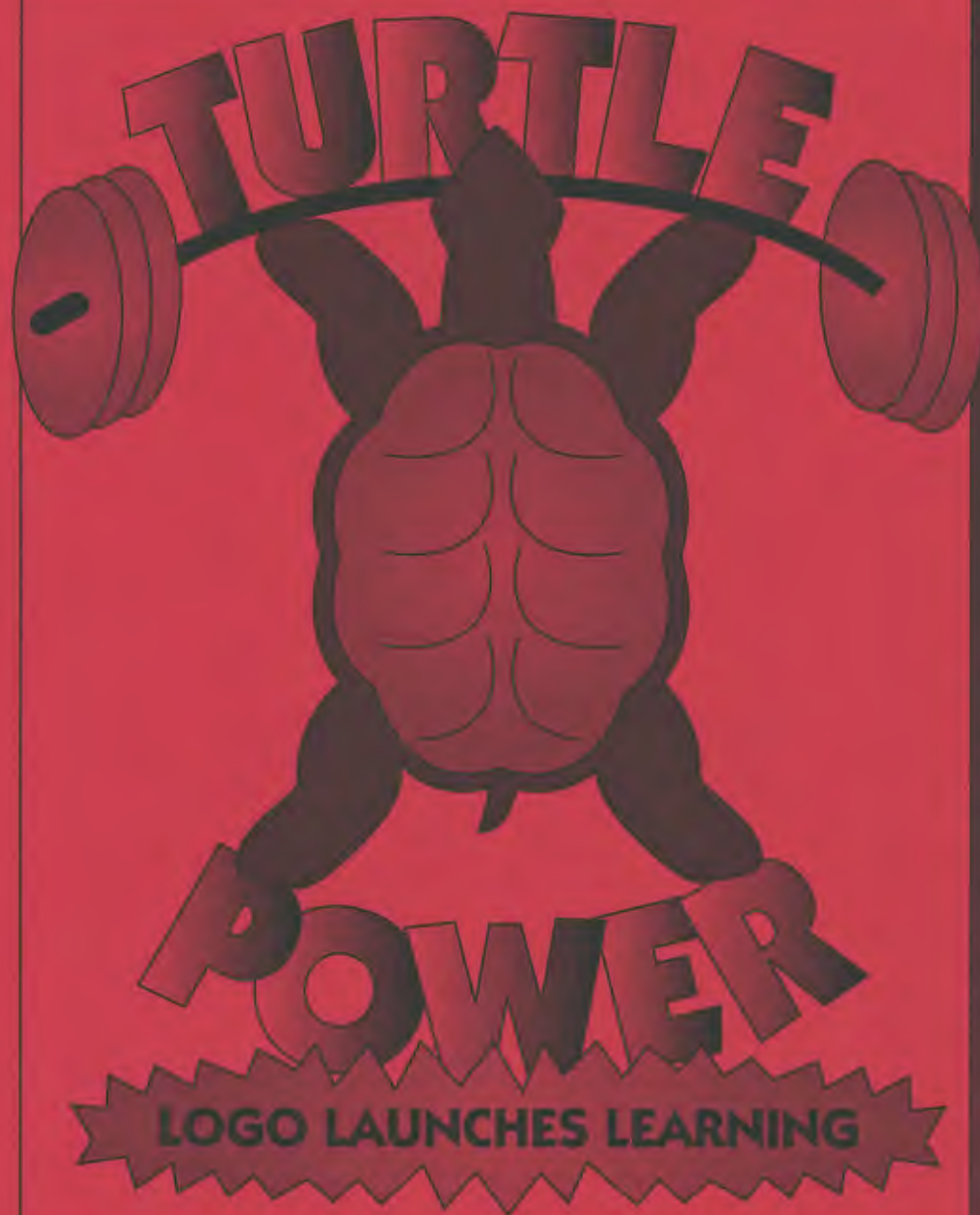
*Add Additional 7% of SUBTOTAL if shipped to PO Box, AK, HI, or outside U.S.

Add 7% of SUBTOTAL for GST if shipped to Canada

If billed with purchase order, add \$4.00; If COD, add \$4.50

* If actual shipping cost exceeds this amount, we will bill you for the difference.

TOTAL



Many educators around the globe have been using turtle power in their classrooms with very positive results. As a beginning programming language, Logo is your logical choice.

ISTE's Logo in the Classroom series gives teachers fun, easy, and innovative methods for using Logo programming to improve the thinking and problem-solving skills of precollege students.

Logo concepts and tools can be used to teach a variety of subjects. The series offers interactive ways to use Logo for teaching mathematics, language arts, art, modeling and simulations, and computer science.



International Society for Technology in Education
Customer Service Office
480 Chamblin Street, Eugene, OR 97401-2626 USA
Phone: 800/336-5191 (US and Canada)
Phone: 541/302-3777 (International)
Fax: 541/302-3778 Fax-on-Demand: 800/891-0311
America Online: ISTE CompuServe: 70014,2117
Internet: iste@oregon.oregon.edu
World Wide Web: <http://isteonline.oregon.edu>

LOGO, LOGOWRITER, AND MICROWORLD ARE REGISTERED TRADEMARKS OF LOGO COMPUTER SYSTEMS, INCORPORATED, 17501 BURNING WOOD DRIVE, BURNING WOOD, OHIO 44125-1099. ILLUSTRATION AND DESIGN BY TONY BROWN. ALL RIGHTS RESERVED. ISTE, 1995.

The ISTE Logo in the Classroom Books include the following:

101 Ideas for Logo

**The Children's Machine—
Rethinking School in the
Age of the Computer**

Exploring Recursion With LogoWriter

Graphing With LogoWriter

**Introduction to MicroWorlds—
A Logo-Based Hypermedia
Environment**

**Introduction to Programming in Logo
Using LogoWriter, Third Edition**

**Introduction to Programming in Logo
Using Logo PLUS**

**Introduction to Programming
Using Terrapin Logo for the
Macintosh**

**LogoWriter for Educators—
A Problem Solving Approach**

**Logo PLUS for Educators—
A Problem Solving Approach**

**Logo Learning—
Strategies for Assessing Content and
Process**

**Math Activities Using LogoWriter
Series:**

- High School Math
- Investigations
- Maze Investigations
- Numbers and Operations
- Patterns and Designs
- More Patterns and Designs
- Probability and Statistics

**Turtle Power—Beginning Graphics
With LogoWriter, Revised Edition**

**More Turtle Power—Text, Graphics,
and Sound With LogoWriter**

**Turtle Time Designs—
Working With Patterns in LogoWriter**

**Weekly Words for Traveling Turtles—
Logo Applications in Language Arts**

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 yourself prepare for the future of education.

ISTE members benefit from the wide variety of publications, specialized courseware, and professional organizations available to them.

They also enjoy exciting conferences, global peer networking, and mind-expanding in 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 Fax-on-Demand: 800/891-0212

America Online: ISTE CompuServe: 70014,2117

Internet: cust_svc@cmail.uoregon.edu

World Wide Web: <http://isteonline.uoregon.edu>

WE'LL PUT YOU IN TOUCH WITH THE WORLD.