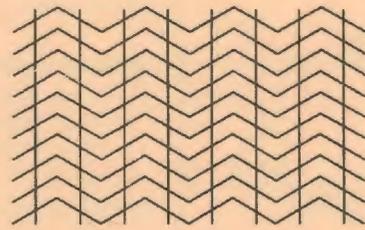


LOGO EXCHANGE

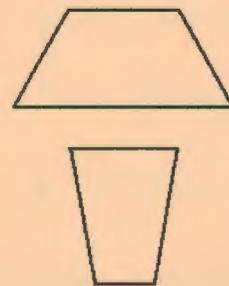
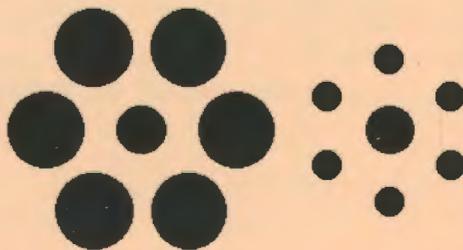
Volume 15 Number 2

Winter 1996-1997

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



Fun with Optical Illusions!



In this issue:

Simulations: Getting Started

Do You Remember When...?

Robotics: At the Supermarket

Fast Food Fun: Dollars and Cents

Learning in a Notebook-Saturated Environment

The Doctor's Dilemma

The Logo Revolution

Optical Illusions

A Tale of Three Teachers

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 Friesen, President
Steve Sesko, Vice-President
Hope Chafian, Secretary/Treasurer
Dorothy Fitch, Editor

Production

Kerry Lutz

Director of Advertising Services

Judy Stickney

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:

71760.366@compuserve.com

Deadlines

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

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

ISTE Board of Directors 1996-97

David Brittain, *President*
MGT of America (FL)
David Moursund, *Executive Officer*

Executive Board Members

Lynne Schrum, *President-Elect*
University of Georgia-Athens (GA)
Terrie Gray, *Secretary*
Chico Unified School District (CA)
Terry Killion, *Treasurer*
Jefferson County R1 School District (CO)
Cheryl Lemke, *At Large*
Illinois State Board of Education (IL)
Gwen Solomon, *At Large*
UIS Department of Education (DC)

Board Members

Chip Kimball
Lake Washington School District (WA)
Paul O'Driscoll
Salem-Keizer Public Schools (OR)
Kathryn Edwards Price
Apple Computer, Inc., Santa Monica (CA)
Carla Schutte
Moton Elementary School (FL)
Neal Strudler
University of Nevada-Las Vegas (NV)
Harriet Taylor
Louisiana State University (LA)
Peter Wholihan
Saints Peter and Paul Catholic School (VI)
Michael Turzanski
Cisco Systems (MA)

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*®.

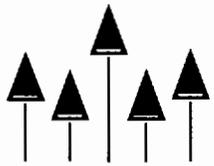
ISTE Members may join SIG/Logo for \$24.00. Dues include a subscription to *Logo Exchange*. Non ISTE Member subscription rate is \$34.00. Add \$10 for mailing outside the USA. Send membership dues to ISTE. Add \$4.00 for processing if payment does not accompany your dues. VISA, Mastercard, and Discover 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; 508/750-8400; Fax 508/750-4470. ISTE members may apply directly to the ISTE office for free reprint permission.

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



Logo Exchange

Volume 15 Number 2

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

Winter 1996-1997

Contents

From the Editor		
The Old and the New	<i>Dorothy M. Fitch</i>	2
Quarterly Quantum		
Make "It :Better	<i>Tom Lough</i>	3
The Beginner's Corner		
Simulations to Get You Started	<i>Jim Muller</i>	4
Do You Remember When...?	<i>MACUL Newsletter</i>	8
Robotics Fun: At the Supermarket	<i>LEGO DACTA CONNEXIONS</i>	12
Fast Food Fun: Displaying Dollars and Cents	<i>John Gough</i>	15
Whatever Happened to the Revolution?	<i>Geraldine Kozberg</i>	17
Optical Illusions	<i>William J. Spezeski</i>	21
Learning in a Notebook-Saturated Environment	<i>Jenny Betts</i>	26
A Simulation Challenge: The Doctor's Dilemma	<i>Jim Muller</i>	31
Logo: Search and Research		
Logo in the Classroom: A Tale of Three Teachers	<i>Julie Sarama and Douglas H. Clements</i>	34



From the Editor: The Old and the New

by Dorothy M. Fitch

Training an entire town to use technology—now there's an interesting concept. That's exactly what is happening in my town of Derry, New Hampshire (population 32,000), where the only supermarket in town is now the first in New England to offer shoppers the opportunity to "check themselves out."

We can now scan our own purchases at one of four computerized stations. Each customer scans bar-coded items, places produce on a scale to be priced, bags the items, and pays with cash or credit card. A touch screen and helpful voice guide us through the process. One clerk oversees the four stations, verifying IDs for alcohol and tobacco sales, identifying (via a monitoring screen) and pricing produce, and preparing credit card receipts for signature.

Alas, the system is not foolproof. As a security measure, scales under each of three bags are supposed to confirm that the objects scanned are the same ones put in the bags. However, this is unclear to the customer.

The store manager I talked to confirmed that this is the one area in which people are having trouble. They put pocketbooks and keys on the scale. They lean on it, and their kids climb on it. The system gets confused.

Shoppers have no trouble following the touch-screen instructions. The store expected older shoppers to have the most trouble with the system, but these folks seem to like it a lot. In fact, it is the 35–55 age group that doesn't like it. The store manager is surprised and has no explanation. (Bring on the sociologists!)

I was expecting (and hoping) to hear that results and observations from the test period were causing changes to be made to the computer software or hardware, but that is not the case. The system has been deemed fit for use and will not be changed. Instead, the store is changing the way in which it is working with the machines and helping customers.

Apparently the debugging process is for people only. An interesting idea. Time will tell if it is really the answer.

In this issue

Supermarket scanning is also the topic of the robotics article this issue. This article will help your students design and build a supermarket scanner. Perhaps they can build a foolproof scanning system!

Supermarket scanning is part of a recurring theme of this issue: looking backward and looking forward.

A computer timeline of sorts, offered by the Michigan Association for Computer-related Technology Users in Learning, takes us down memory lane.

Gerry Kozberg shares the text of her keynote address to the 1996 Logosium. In it, she looks back at the Logo revolution promised to us in the 1980s and ahead to the future of Logo.

Jenny Betts tells us of a forward-looking school in Australia where each student uses Logo extensively on his or her own laptop computer.

Julie Sarama and **Doug Clements** take a look at how teachers used Turtle Math in their school and what influenced their experiences.

In other fun and challenging articles, **John Gough** builds a fast food cash register (a nice tie-in to the supermarket theme), **Jim Muller** explores simulations and **Bill Spezeski** helps us explore optical illusions. I'm not sure which way those make us look!

Happy Logo adventures (and don't forget to send me feedback, articles, and student work)!▲

Dorothy M. Fitch

Logo Exchange, editor

3 Derby Road, Derry, NH 03038

E-mail: 71760.366@compuserve.com

Ph. 603/425-2010

Fax: 603/425-6487



Quarterly Quantum: Make "It :Better

by Tom Lough

By the time you read this, the Olympic fever of this past summer will have abated somewhat. But the impressions are still fresh in my mind as I write this column.

What a thrill it was to be in Atlanta during the Olympic Games, even if it was only for a couple of days! Thinking back over the experience, I came upon a question with implications for the classroom.

Of the hundreds of thousands who participate in the various Olympic sports, only a few get the opportunity to compete in the showcase quadrennial games. If so few "win," then why do so many continue to participate?

As teachers, we often ask ourselves the same type of question. Why do we continue to teach?

I suspect that athletes and teachers would share a couple of common responses.

Many athletes continue to participate in their sports because they simply love them. They enjoy the different aspects of their sports and take personal pleasure in participating. I hope that you feel that way about your teaching (and about your Logo teaching in particular).

Athletes are also highly motivated by improvement. Runners and swimmers keep detailed records of their "personal best" times for various distances, and celebrate when they beat them. This idea of incremental improvement is one we can bring into the Logo classroom.

There are no Logo Olympics yet (although who knows what our Australian colleagues are cooking up for the year 2000!). But so what? That doesn't stop us from trying to improve on our best.

I have seen teachers improve their Logo-based lessons each year, making changes based on what they learned from their students the year before. I have seen Logo teachers willing to take risks and branch out in completely new direc-

tions, developing and trying out different types and styles of projects. I have also seen Logo software companies enhancing current products and developing new types of applications for this powerful computer language.

This frame of mind, always trying to do it better, is what will bring the Logo perspective along as education continues its evolution into the Knowledge Age. Otherwise, Logo and everything it stands for runs the risk of becoming a dinosaur, relegated to the scrap heap of educational innovations that did not make it.

"How can we make our best even better?" This is a question athletes of every stripe ask themselves continuously. It is a question used by Logo teachers to challenge themselves.

The result? Not everyone can make the Olympic team, much less win a medal—especially in teaching. But members of the Logo community can take pride in knowing that we are participating in a manner that is making us better for it.

One final thought. As athletes progress through their careers, they themselves are usually the only beneficiaries of their improvements. In education, we have an added bonus: Our students benefit from our improvements as well.

So, what are we waiting for? As the lyrics of "Hey Jude" (Lennon-McCartney) invite us, "start to make it better." ▲

FD 100!

Tom Lough

Founding Editor

Box 394

Simsbury, CT 06070

E-mail: 70020.223@compuserve.com



The Beginner's Corner: Simulations to Get You Started

by Jim Muller

Back when a group of young people and I were attempting to learn Logo, we ran into a brick wall. We had all learned the basic commands and how to write procedures. But what came next was a rather random series of trial-and-error explorations as the young people tried to develop some more complex ideas.

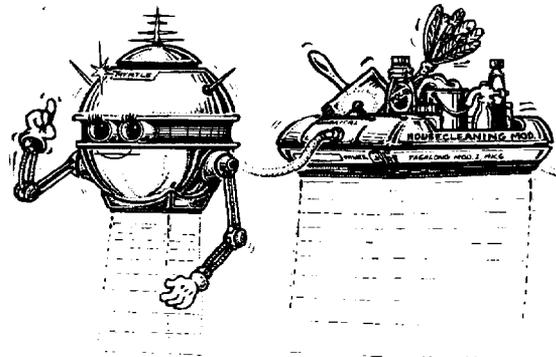
There is certainly nothing wrong with trial-and-error explorations. However, they can get out of hand when they turn into pure hit-and-miss experimentations. The problem was how to get those youngsters to “think like a computer.” What spark could help them add logic and structure to their experimentations?

This dilemma led to the development of Mrtle, the Multifunctional Robotic Task and Labor Expediter, Model X409A, who became the third in our series of cartoon characters. Just as we had done with our other characters, Logy and Morf, we naturally developed stories about how Mrtle came into being and her later misadventures.

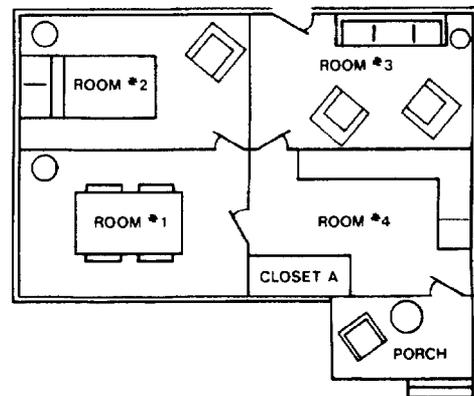
Mrtle's story

Mrtle was one of a new series of household helper robots complete with antigravity drive, natural language processing, synthetic speech, and speech recognition capabilities. However, it seems that Mrtle was a failure as a robot. She never learned to speak “computerese.” She also had this rude tendency to talk back. So she was packed up and was on her way back to the factory when the crate fell off the truck into a dry creekbed. Two brothers found the crate and, following the instructions inside, brought Mrtle “back to life.”

The boys had been told to clean the house while their mother was shopping. Now that they had this robot, all they had to do was program it to do the job while they watched TV.



This is where you can begin to get the children involved in the story. Have one child play the part of the robot while the others develop a program for the robot to clean the house. Here is the layout of a very simple house.



Simulating a robot is not quite as easy as it sounds.

The child selected to be the robot must simulate how a robot (or a computer) would carry out instructions. The robot cannot think. It cannot pick and choose which instructions to carry out, nor can it interpret instructions. It can only do exactly what it is programmed to do in the exact order set by the program—nothing more, nothing less.

The programmers do not have an easy job either. They must provide exact, step-by-step instructions. They must break the tasks into commands that tell the robot exactly what needs to be done. Although this sounds easy enough, it can lead to problems. For example, if the boys told the robot to get the vacuum cleaner out of closet A, the robot might just crash through the door or wall to get the vacuum cleaner. That just creates a bigger mess to clean up.

To get you started, here is a set of instructions for the robot. Are they correctly written? Are they complete? The lines are numbered to help you refer to specific steps in the process.

1. Scan the plan of the house.
2. Go to closet A.
3. Open the door.
4. Pick up the vacuum cleaner and dust cloth.
5. Go to room 1.
6. Find the wastebasket and pick it up.
7. Empty the wastebasket into the garbage can on the back porch.
8. Dust all of the furniture and shelves with the dust cloth.
9. Plug the vacuum cleaner into the electric wall socket.
10. Vacuum the floor.
11. Unplug the vacuum cleaner.
12. Go to room 2.
13. Find the wastebasket and pick it up.
14. Empty the wastebasket into the garbage can on the back porch.
15. Dust all of the furniture and shelves with the dust cloth.
16. Plug the vacuum cleaner into the electric wall socket.
17. Vacuum the floor.
18. Unplug the vacuum cleaner.
19. Go to room 3.
20. Find the wastebasket and pick it up.
21. Empty the wastebasket into the garbage can on the back porch.
22. Dust all of the furniture and shelves with the dust cloth.
23. Plug the vacuum cleaner into the electric wall socket.
24. Vacuum the floor.
25. Unplug the vacuum cleaner.

26. Go to room 4.
27. Empty the wastebasket into the garbage can on the back porch.
28. Dust all of the furniture and shelves with the dust cloth.
29. Plug the vacuum cleaner into the electric wall socket.
30. Vacuum the floor.
31. Unplug the vacuum cleaner.
32. Go to closet A.
33. Put the vacuum cleaner and the dust cloth in the closet.
34. Shut the closet door.
35. Shut yourself off. You deserve a rest.

As the story goes, the older brother was the first to notice that he didn't hear the vacuum cleaner running. Both boys went to see what was wrong. They found Mrtle vacuuming the back porch again as she had been instructed. It seems that the boys forgot a step between #7 and #8. They forgot to tell the robot to return to the room she left.

Finding errors such as this is the whole idea behind computer simulations. The goal is to find such errors before actual work begins. For example, if the first two little pigs had simulated the conditions their straw and twig houses would have to withstand, they might have elected to use brick in the first place.

Simulations in the real world

In the real world, systems engineering teams use design-automation and simulation tools to help define the requirements and specifications for a product that meets the needs of their customer—before any design or production work is done. They start with a customer requirement such as “design a car for commuters that is safe and energy efficient, and that will not pollute the environment.”

Next they create a system hierarchy that shows the relationship between the different systems to be assembled. Each subsystem is broken down and diagrammed until they reach the lowest level of the hierarchy that shows the car's simplest elements. The hierarchical diagram is like a giant organization chart except that each box represents a functional element of the system rather than a person.



Each part of the car has a cost, a weight, and operating characteristics. When the functional block diagram is complete, the design-automation software allows the development team to evaluate the total cost of the car, the total weight, and what the operating characteristics are when all the parts are working together properly.

Keep in mind that systems engineers are able to simulate a wide range of conditions *before* anyone actually begins to design the car. Manufacturers can no longer afford to build increasingly complex products and then test them to see if they work correctly. They need to be able to accurately simulate designs beforehand to minimize the time and expense of redesign.

Linear thinking vs. structured thinking

To put this on a simpler level, analyze the basic requirements for Mrtle to clean the house with n number of rooms. She must first prepare to do the job by gathering her materials. Next she does the required cleaning. Finally, she puts things away and shuts down.

When you apply Logo logic to this problem, you define a set of procedures to implement the cleaning process.

```
To Cleanhouse :n
  Prepare
  Clean :n
  Shutdown
End
```

```
To Prepare
  Go to closet A
  If door closed open door
  Pick up vacuum cleaner and dust cloth
  Close door
End
```

```
To Clean :n
  Go to room :n
  Put down vacuum cleaner and dust cloth
  Empty
  Dust
  Vacuum
  If :n + 1 = 5 shutdown
  Clean :n + 1
End
```

```
To Empty
  Go to wastebasket
  If empty dust
  Pick up wastebasket
  Go to back porch
  Empty wastebasket in garbage can
  Go to room :n
End
```

```
To Dust
  Pick up dust cloth
  Dust furniture and shelves
End
```

```
To Vacuum
  Pick up vacuum cleaner
  Put vacuum cleaner plug into empty electric
    wall socket
  Turn vacuum cleaner on
  Vacuum floor
  Turn vacuum cleaner off
  Unplug vacuum cleaner plug
End
```

```
To Shutdown
  Go to closet A
  If door closed open door
  Put vacuum cleaner and dust cloth in closet
  Close door
  Give yourself a rest
End
```

Each step in these procedures can be broken down further to simulate each movement of the robot. A simulation like this prevents such errors as going through walls or crashing into furniture.

This is all well and good, but these procedures total 47 lines, whereas the first program was only 35 lines. What makes this approach better? The answer is obvious to any Logo fan(atic).

- The procedural approach allows changes to be easily made to any step.
- Additional procedures can easily be added.
- Individual procedures can be copied to other applications as needed. You don't have to re-invent the wheel each time you want to follow the same steps somewhere else.
- Procedures are easier to debug and correct than long lists of separate steps.

A challenge to prove a point

To look at this from another perspective, suppose you purchased several such robots for your

CyberFrontier



lit the trail with Judi Harris' abled ferret as she unearths Web sites, original source documents, free and inexpensive software, and even more on the educational frontiers of the Internet.

Cyberspace is bigger and better and so is the *Way of the Ferret!* Fresh, new chapters in the second edition of *Way of the Ferret* will expand your...

...**information-gathering skills** using

- Gopher tools
- World Wide Web browsers
- Card file and transfer protocols

...**networking skills** in contacting other educators and people with similar interests via

- Electronic mail
- Listservs
- Newsgroups
- Directories

...**imagination** in designing dynamic curriculum-based activities for your students, choosing from 18 effective activity structures.

...**organizational abilities** in revolutionizing your students' educational experiences with telecollaborative and telereasearch activities.

Extending the common-sense approach of her "Mining the Internet" column in *Learning & Leading With Technology*, Judi Harris uses her experience with students and educators along with skillful, humorous analogies to explore the boundaries of the cyberfrontier.

Way of the Ferret—Finding and Using Educational Resources on the Internet, Second Edition—your gateway to new and exciting territory.



International Society for Technology in Education
Customer Service Office
Phone: 800/336-5191 (US and Canada)
Phone: 541/302-3777 (International)
Fax: 541/302-3778
World Wide Web: <http://isteonline.uoregon.edu>

housecleaning business. Using wireless technology, you have two-way communication with each robot. Which approach would you use to get the best performance from each robot: the linear approach or the procedure approach?

What are the steps you would need to add to control five robots simultaneously?

What additional information do you need from each homeowner?

Simulation is not all block diagrams and technical approaches to problems. It is what sparks the imagination. It is what turns fantasy into a unique form of reality.

Think about it. Are not many games a form of mathematical simulation? Flight simulators? Auto driving and racing simulators? Sports games?

What child hasn't simulated adult behaviors as a form of play? Played dress up? Served a tea party? Been a fireman? A warrior? There are those who look at childhood art as a child's simulation of what he or she sees, broken down into usable forms that the child can manipulate on his or her terms.

Why not have some fun and add your own endings to the "Misadventures of Mrtle"—how she and her friends cleaned every house in the neighborhood?

For another look at simulation, try the challenges offered in "The Doctor's Dilemma" in this issue. Even cellular activities can be fun! ▲

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. 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. Look for Jim's ideas and activities in the newly updated *The Turtle's Discovery Book*, an expanded two-volume set, published by Harvard Associates, Inc. You can reach Jim by e-mail at 76703.3005@compuserve.com or on CompuServe at 76703,3005.



Do You Remember When...?

from the 20th Anniversary Issue of the MACUL Newsletter, March 1996, by permission

[Editor's Note: In this Special Edition newsletter, MACUL (Michigan Association for Computer-related technology Users in Learning) celebrated 20 years of service to teachers and students. *LX* invites you to "Learn From the Past ... Look to the Future" with MACUL. How many of these milestones in computer education history do you remember?]

1975

- Steve Jobs, a 19-year-old technician at Atari, and Steven Wozniak, a young electronics engineer at Hewlett-Packard, bought a microprocessor for \$25 and built a computer in the living room of Job's parent's home in Palo Alto, California.
- In April, Bill Gates and Paul Allen form Microsoft as a software-writing partnership. Their first project was a version of BASIC for the Altair.
- David Ahl's *Creative Computing* magazine, which debuted in October 1974, is the first magazine devoted to microcomputers for hobbyists and, a few months later, educators.
- *Byte* computer magazine is launched by Wayne Green who in the next few years would also introduce other very popular computer magazines of their time—*Hot Coco*, *80-Micro*, *Run*, and *InCider*.

1976

- In the spring of 1976, Jobs and Wozniak showed their first machine, dubbed Apple I, at the new Home Brew Computer Club, the nation's first home-computer club. They also take the Apple I to a new Palo Alto computer store, which orders 50. Wozniak pawned his two H-P calculators and Jobs sold his VW van and with \$2,500 they start Apple Computer.

1977

- In April, the Apple II is shown publicly in commercial quantities at the First West Coast Computer Fair in San Francisco with first year sales totaling \$2.5 million!
- Tandy Corporation introduces TRS-80 computers in Radio Shack stores.

1978

- In the June *MACUL Journal* (Vol. I, Number 2), an article by University of Michigan Research Scientist Karl L. Zinn concluded: "I am convinced that the 'bicycle' of the educational computing world will find its place. The PET is only the first, primitive entry in this market, which will include a greater variety of models of increasing function and beauty every year. Perhaps some schools and homes will even be led to rethink the role of bicycles, in meeting their transportation needs most economically and healthfully. Surely the PET (or its successors) will play a major role in recreational education in the homes, even replacing educational activities previously carried on by the schools."
- In June, Intel introduces the 16-bit 8086 microprocessor.

1979

- In August 1979, David Moursund founded the International Council for Computers in Education (ICCE, now ISTE) as a nonprofit professional organization that has grown to be the largest professional computer (now technology) organization in the world.
- Radio Shack offers Level II 16K TRS-80 microcomputers for \$988.
- In June, Intel introduces the 8088 microprocessor, a hybrid 8-bit/16-bit version of the 8086. IBM will build its first PC (Personal Computer) around the 8088, which has the same raw computational power as an IBM 360/50 mainframe.
- Visicalc (visible calculator) is introduced for the Apple II and becomes the first user-friendly, large-selling spreadsheet software program.



- WordStar, the first big-selling word processing program is introduced by MicroPro.

1980

- Sinclair ZX80 (6.5 in. x 8.5 in. x 1.5 in.) is billed as the “world’s first truly portable computer.” The machine had 1K RAM and a wipe-clean, touch-sensitive keyboard with the standard QWERTY layout and was priced at \$199.95. The display of channel 2 of standard TV (user supplied) provided up to 24 lines of 32 characters each.
- New computer magazine for educators, *Classroom Computer Learning*, debuts. (Later the name is changed to *Technology and Learning*.) About half of the content of the early issues of *CCL* is devoted to computer programming, teaching about the computer, and programming codes teachers can type into a computer to create simple quizzes and games.
- The Apple III is introduced. Features include “80-character, upper/lower case display for easy text formatting, up to 128K of RAM for increased data handling, four dedicated cursor keys for single-keystroke cursor movement, built-in disk drive, integrated numeric pad, built-in clock/calendar, special built-in emulation capability that ‘converts’ an Apple III system to an Apple II in just seconds.”
- Sony Walkman cassette player introduced. More than 25 million units will be sold in the next 10 years.
- Ashton-Tate is founded in August and soon releases dBase II, the first big-selling PC database management software.

1981

- A computer cost-comparison sheet of the time compares four popular computer systems. Apple II Plus: 48K, Disk II with controller, 9" black-and-white TV monitor, TV interface—\$1,935; TRS-80 Model III: 16K with built-in black-and-white monitor, two 16K RAM kits, one internal mini-disk drive (note: no color TV output)—\$2,086; Atari 800: 16K with two 16K RAM modules, disk drive, black-and-white monitor cable (includes color TV output capability)—\$2,035; Commodore PET 2001: 32K (48K not available) with built-in TV monitor, dual disk drive (single disk not available,

so price indicated is half the double disk)—\$2,990.

- On August 12, IBM introduces its long-awaited microcomputer, the “IBM Personal Computer.” The IBM PC uses Microsoft’s BASIC 80 for its 40K ROM and up to 256K of RAM. List price exceeds \$3,100.
- Most schools that have computers are emphasizing programming in one of several languages: BASIC, which came with most computers; Logo; FORTRAN; and COBOL.
- Osborne Computer Corporation introduces the first portable computer, the Osborne I, which weighed 27 pounds and was priced at \$1,795 bundled with WordStar and SuperCalc.
- In December, Apple goes public, creating instant millionaires of many early employees who own stock in the five-year-old, billion-dollar company.

1982

- Commodore-64 at \$595 (plus a \$39.95 tape drive) becomes the first popular home computer.
- Popular software of the time included Bank Street Writer, Crossword Magic, Hodge Podge, Terrapin Logo, Krell Logo, SRA Courseware, and Apple Gradebook.
- A survey by Market Data Retrieval noted that “of all 15,442 school districts in the United States, only 42% had at least one microcomputer.”
- In February, Intel introduces the 286 microprocessor.
- Apple offers, for a limited time, a specially packaged Apple II Plus (48K of internal memory, disk with controller card, 12-in. monitor III with stand, and word processing program AppleWriter 2.0 for \$1,995. Regular price is \$2,604.
- *Time* magazine names the computer its “Man of the Year.”

1983

- First copies of Lotus 1-2-3 are shipped in January.
- In February, Apple introduces the short-lived Lisa for \$9,995.00.
- In March, IBM introduces the PC/XT and Microsoft introduces MS-DOS 2.0.



- Microsoft introduces its Word 1.0 word processing program.

1984

- IBM introduces the ill-fated PCjr.
- With much fanfare, Apple introduces the Macintosh with its small 9-in., black-and-white screen. Priced at \$2,495, the first Mac came bundled with MacPaint and MacWrite and popularized the use of the mouse, windows, and pull-down menus.
- Brøderbund's user-friendly The Print Shop enables every computer user to produce a tangible product. For several years, virtually every schoolroom displayed items produced using the popular graphics program.
- IBM begins marketing its "Writing to Read" multidimensional system that helps develop the writing and reading skills of kindergartners and first graders.

1985

- Commodore Amiga, which boasts tremendous graphics capabilities for the time, is introduced and is well received in artistic and video-production circles.
- Apple introduces the LaserWriter, the printer that, along with the Macintosh, helped launch desktop publishing.
- Intel introduces the 386 microprocessor.

1986

- One of the most popular software programs of the year is Brøderbund's Where in the World is Carmen Sandiego.
- In the fall, a national survey reported that 60% of all computers in K-12 schools carried the Apple nameplate.
- Market Data Retrieval reported that the nationwide student-to-computer ratio had fallen to 36.5 students for each computer, compared with 45.5 students per computer in 1985.
- Apple introduces the IIGs and the Mac II, a full-color modular Macintosh.

1987

- Apple introduces HyperCard.
- Microsoft brings out MS-DOS 3.3.
- IBM upgrades its school offerings from the PC to the PS/2 line, including models 25 and 30.

1988

- The Apple-compatible (IIE and IIC) Laser 128 computer is advertised for \$649.95, including a monochrome monitor, Panasonic KZP-1080i printer, and all necessary cables.
- Microsoft begins shipping MS-DOS 4.01.
- Computer users become aware of and concerned about computer viruses.

1989

- Newly introduced Apple Macintosh portable with 40 MB hard disk retails for \$6,499, school price \$4,359.
- AppleWorks 3.0 is released by Claris Corporation for \$249.00 retail.
- Microsoft Works 2.0 is the most popular multipurpose program of the time.
- In April, Intel debuts its 32-bit 486 microprocessor.

1990

- Apple introduces the Macintosh LC (\$2,000 for 2 MB RAM, color monitor, and 40 MB hard drive), the first Macintosh specifically designed for schools.
- In May, Microsoft introduces Windows 3.0.
- A survey by Market Data Retrieval noted that "99% of school districts in the United States had at least one microcomputer."

1991

- Popular software of the year included Brøderbund's Kid Pix, Roger Wagner's HyperStudio 2.1, and National Geographic's Mammals: A Multimedia Encyclopedia.
- Several years later than expected, Apple introduces System 7 for the Macintosh.
- In June, Microsoft brings out MS-DOS 5.0.
- In July, long-time rivals Apple and IBM join forces to counter Microsoft's hold on operating systems software.

1992

- Early in the year Apple introduces QuickTime, which is followed by Microsoft's Video for Windows and an IBM version of digital video.

1993

- Apple discontinues the Apple IIGs and Tandy stops making Radio Shack computers.



- Davidson's Kid Works 2, Edmark's KidDesk, Microsoft's Encarta, and Edmark's Millie's Math House are popular software titles.

1994

- The computer marketplace is flooded with dozens of new CD-ROM titles at very reasonable prices when compared with software titles of earlier years.
- Early microcomputer industry leader Commodore, which at one time controlled more than 30% of the personal computer market, goes out of business.

1995

- Attendance at MACUL '95 was a record-shattering 3,767. Summary of conference attendance figures:

1977	189	1987	2,750
1978	298	1988	1,827
1979	305	1989	1,456
1980	623	1990	2,400
1981	560	1991	2,624
1982	1,200	1992	2,500
1983	3,250	1993	2,600
1984	2,174	1994	3,040
1985	1,995	1995	3,767
1986	2,108		

1996

- America Online President Steve Case stated in his February letter to AOL users that America Online reached yet another milestone by welcoming its five-millionth member. Case states, "To put this in perspective, AOL is now among the largest 'cities' in the world; more people live in the AOL community than in the metro areas of San Francisco, Madrid, or Sydney (and larger than the city populations of Los Angeles and Chicago). And more people now subscribe to AOL than to *The Wall Street Journal*, *USA Today*, and *The New York Times* combined." ▲

"The ... information [in this article] was gathered from a myriad of sources, including early periodicals, catalogs, and advertising brochures. An especially useful source was *Creative Computing* magazine, which provided a wealth of interesting details about the first decade of the microcomputer revolution that began in 1975, the same year MACUL was conceived."

—Burton H. Brooks, MACUL Newsletter Editor
E-mail: BHBrooks@aol.com



Robotics Fun: At the Supermarket

reprinted with permission from LEGO DACTA CONNEXION™, Vol. 3 No. 1, Jan. 1990.



In many stores, the prices of merchandise are marked on each item with a special code, which is called a bar code because it consists of a number of black lines or bars. The cash register can read these codes with a barcode reader and print the tab automatically.

In this activity you will build a LEGO bar-code reader and program the computer to be your cash register!

Key Ideas

1. Codes—what are they and why are they helpful?
2. Adding numbers with decimals (prices)
3. Using optosensor feedback to count
4. Programming concepts: adding items to a list, adding elements within a list

Materials needed:

1 LEGO® TC logo system (or other robotics system)

1 optosensor (light sensor)

1 light brick (light emitter)

black marker

stiff white paper at least 2 cm x 6 cm

Mark a few black lines on some stiff white paper strips.



Set up a LEGO light brick across from an optosensor. Plug the light into port 0. Turn the light on by typing **talkto 0 on**. Plug the sensor into port 6. As you pass the bar codes in front of the sensor, check whether the small green light at port 6 comes on the right number of times. (It should come on once for each bar on the paper strip.)

To make your optosensor read your bar code accurately, find out:

- How close together the light and optosensor should be.
- How close together the black lines can be.
- How thick the black lines can be.
- How quickly you can slide the bar code past the optosensor and still get an accurate reading.

Assume that items you sell with the  bar code cost \$.50; items with the  bar code cost \$1.00; items with the  bar code cost \$1.50; and items with the  bar code cost \$2.00.

Write a procedure to print a price for each bar code. Follow the sample below:

```
to cashregister
  listento 6
  resetc ; resets the counter
  wait 50 ; gives you time to pass the
           barcode in front of the optosensor
  if counter = 1 [print $.50] ; if the
           barcode has one line, print $.50
end
```

Change the procedure to print prices of any bar code you scan. Then flip your page and try the procedure **cashregister**.

You will want to add up a customer's tab automatically as each item is scanned. One way to do this is to keep each price in a list. Then you can add the numbers in the list.

```
to addup :list
if empty? :list [output 0]
output (first :list) + (addup butfirst
: list)
end
```

Try **addup** with a few numbers. In the Command Center, type:

```
make "tab [1 2 3]
show addup :tab
```

For the next step you must make the computer add your bar code readings to a list so that **addup** can add them together.

Append puts an object (price) on the end of a list (tab).

```
to append :object :list
make :list lput :object thing :list
end
```

Try **append** with a few numbers. In the Command Center, try:

```
append 4 "tab
show addup :tab
```

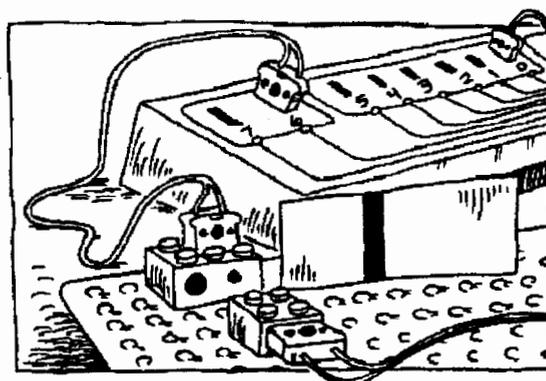
Now you can include **addup** and **append** to the **cashregister** procedure.

```
to cashregister
listento 6
resetc
wait 90 ; increased from 5 seconds
if counter = 0 [print sentence [Your total
bill is $] addup :tab stop]
; stop adding and print the total tab if,
after nine seconds, no barcode is scanned.
if counter = 1 [print "$.50 append .5 "tab]
if counter = 2 [print "$1.00 append 1 "tab]
if counter = 3 [print "$1.50 append 1.5
"tab]
if counter = 4 [print "$2.00 append 2 "tab]
wait 10
cashregister ; check for the next item. The
procedure is recursive; you can scan all
items a customer buys.
end
```

Stop! Before you try **cashregister** program, you must give "tab a starting value. For each customer we should begin with "tab as an empty list of prices.

```
to mybill
talkto 0
on ; turn on the light
make "tab [] ; make "tab an empty list
cashregister
end
```

Flip to the turtle side of the page. Then type **mybill**. Wait until you see an item's price printed on the screen before scanning the next item. You are ready to open shop!



Challenges

1. Include tones in your program to make the cash register ring so you know when to put the next bar code through.
2. Design and build a conveyor belt. Adjust the speed of the motor to ensure that bar codes are read correctly when items pass by the optosensor.

Teacher notes

Permanent black markers work well on stiff white paper. Transparency markers, even permanent ones, do not.

The optosensor is very sensitive and has to be reset to the "false" position before starting to scan. This is done by the LEGO light mounted across from the sensor.

The procedure **addup** uses embedded recursion, a complex, but important concept. If students are not ready to learn embedded recursion, they can use the **addup** and **append** procedures as if the procedures were primitives.



Use **output** when you want a procedure to “hand over” a value to another procedure.

The procedure **append** takes two inputs. The first (:object) is the price of an item calculated by **cashregister**. The value of :object is determined by the value of the counter. For example, if **counter = 1**, :object is .5. The second input to **append** is the name of a list. The value of the list to be used by **append** is input by **cashregister**.

Tab is the name of the list within the **cashregister** procedure. **Append** takes the value of :object and adds it to the end of the list. For example, if the value of the list "**tab**" (denoted by :tab) is [0 1 2], the instruction **append "tab 3** adds 3 to the list "**tab**, creating [0 1 2 3].

If you don't wait for the price of the scanned item to print on the screen, the program continues counting another item's bar code lines and adding them to the counter. As a result, the counter becomes inaccurate and the wrong price prints on the screen.

Challenge tips:

1. Here is one way to add a tone to the program:

```
to bell
tone 330 5
end
```

After creating the **bell** procedure, add **bell** after **wait 90** in **cashregister**.

2. The optosensor and light brick must be close together to read bar codes properly. Modify the conveyor belt model shown on LEGO card 9700-5 so that bricks with attached bar-code paper can pass between the optosensor and the light brick.

Then plug the light into port 0, the motor into port B, and the optosensor into port 6. Below is a suggestion for a new bar-code program using the conveyor belt.

```
to buy.groceries
light.motor.on
read.bar.code
bell
tabulate.price
wait 10
buy.groceries
end
```

```
to light.motor.on
talkto "b
setpower 2
talkto [b 0]
on
end
```

```
to read.bar.code
listento 6
resetc
wait 100
end
```

```
to bell
tone 330 5
end
```

```
to tabulate.price
if counter = 0 [print sentence [Your total
bill is $] addup :tab stopall]
if counter = 1 [print "$.50 append .5 "tab]
if counter = 2 [print "$1.00 append 1 "tab]
if counter = 3 [print "$1.50 append 1.5
"tab]
if counter = 4 [print "$2.00 append 2 "tab]
end
```

Then change **mybill** to:

```
to mybill
make "tab 0
buy.groceries
end
```

Remind students to wait for the bell before putting the next item on the conveyor. ▲



Fast Food Fun: Displaying Dollars and Cents

by John Gough

Here is a challenge: When a numerical quantity refers to money, how can we display it with two decimal places for cents?

The following procedures mimic the idea of a press-button restaurant "menu." By clicking with the mouse in the appropriate place, marked on the screen, the procedure notes where the user has clicked, keeps a running total of the items selected, and then displays the total amount of money this collection of menu choices costs.

These procedures are written using LogoWriter for the Macintosh. At the end of the article are conversion tips for other versions.

```
to menu
cg ct
pu setpos [-200 0] pd label [Apple]
pu setpos [-100 0] pd label [Burger]
pu setpos [ 0 0] pd label [Cola]
pu setpos [ 150 0] pd label [Total] ht
make "recall []
print [Move the mouse and click to choose,
      or pay the bill]
mouse.choice
end

to mouse.choice
if button? [get.choice first mousepos]
wait 1
mouse.choice
end

to mouse.choice :mouse
if and :mouse > -220 :mouse < -150 [make
  "recall sentence :recall 1.50]
if and :mouse > -120 :mouse < -50 [make
  "recall sentence :recall 3.95]
if and :mouse > -20 :mouse < 50 [make "re-
  call sentence :recall 1.75]
if and :mouse > 130 :mouse < 180 [total.it
  :recall 0 stopall] ]
end

to total.it :input :total
repeat count :input
[make "total :total + (first :input)
  make "input sentence :input first :input
  make "input butlast :input]
(pr [Total cost:] (word "$ :total ".))
end
```

Note that it is difficult, initially, to display money correctly in dollars and cents when there is a multiple of ten cents, or when there are no cents and only whole dollars. We need a way to make sure that, under all circumstances, we have two decimal places showing for the cents, whether this is \$x.00, or \$x.y0 or \$x.yz.

The following procedures will help. They split the money into whole dollars and whole cents, and then reassemble these separate parts into an appropriate word, which can be displayed using a print command.

```
to show.money :amount
if 0 = cents :amount [output (word "$ dol-
  lars :amount ".00)]
ifelse 0 = remainder (100 * cents :amount)
  10 [output (word "$ dollars :amount ".
    whole.cents :amount)] [output word "$
    :amount]
end

to cents :amount
output :amount - int :amount
end

to dollars :amount
output int :amount
end

to whole.cents :amount
output round 100 * cents :amount
end
```

Oddly, the **whole.cents** procedure misbehaves for some values of **:amount**, such as 4.7 or 5.2, when it does not include the command **round**. I have no idea why, but it appears only to be symptomatic of LogoWriter for the Macintosh. Try it for different amounts of money and see in what surprising ways it misbehaves.

To include this new set of procedures, replace the last line of the procedure **total.it** with the instruction:

```
(pr [Total cost is] show.money :total)
```



The following way of displaying a number as a money quantity is even simpler.

```
to money :amount
ifelse 0 = cents :amount [output (word "$
  dollars :amount ".00)] [output (word "$
  dollars :amount ". whole.cents :amount)]
end
```

Enhancement Ideas

Here are some additional ideas for making the Logo screen look as much like a real fast-food menu machine as possible.

- Add a tone when a word is clicked to simulate the sound of a cash register or supermarket scanner.
- Add other items to the menu.
- Put a box around each word so it resembles a button.
- Add procedures so that the program can make change for the customer.
- Add an "Oops" button to delete that last item clicked.
- Display the items that have been selected so far.
- Add pictures of the food for a fancier menu.

Conversion Tips

Because MicroWorlds has no primitive for detecting if the mouse button is down, you will have to invent your own nonmouse interface!

For Logo PLUS for the Macintosh, add **splitscreen** to the beginning of the **menu** procedure and change:

ifelse	to	if
int	to	integer
mousepos	to	mouse
stopall	to	toplevel

For PC Logo, change:

cg	to	draw
ifelse	to	if
label	to	turtletext
mousepos	to	mouse
setpos	to	setxy
stopall	to	toplevel

PC Logo treats numbers differently from words, so you will need to use the following version of **show.money**:

```
to show.money :amount
if 0 = cents :amount [output (word "$ dol-
  lars :amount ". "0 "0)]
if 3 = count cents :amount [output (word "$
  dollars :amount ". whole.cents :amount)]
output word "$ :amount
end
```

John Gough lectures in mathematics and computer education at Deakin University and is particularly interested in connections between mathematics and language. He can be reached at Deakin University SDS, 221 Burwood Highway, Burwood, Victoria 3125, Australia, or by e-mail at jugh@deakin.edu.au.



Whatever Happened to the Revolution?

by Geraldine Kozberg, text of her June 14 keynote address at Logosium '96, St. Paul, MN

In the halcyon days of the early 1980s, the technology was new and euphoria high. Logo would transform our schools. From all over the world, the Logo community converged on MIT. Logo '84. Logo '85. Logo '86. We were ready for Star Wars.

At the final session of Logo '86, however, Brian Harvey and Bill Higginson stunned the audience with the question, "Whatever happened to the Logo revolution?" *Logo Exchange* asked me to respond to Brian and Bill, an open letter of sorts. Here are parts of that letter:

August 10, 1986

Dear Brian and Bill,

Your presentation at Logo '86 was stunning. It was a piece of theater, a New Yorker post-script, a witty, sophisticated, calculated statement on education in general, and Logo in particular.

You speak of revolution. You ask, "Whatever happened to the revolution?" It depends. It depends on definitions and perspectives.

Nikolai Lenin: From a Lenin-like view of revolution, we know that we have the weapons. We know how to teach: We have the tools and the resources. We do not have what Lenin called "an intolerable level of discontent." People bitch a lot, but not enough to upset the status quo.

Martin Luther King: From the teaching and life of Martin Luther King, we know that revolutions are not created by the masses, rather by the small committed minority. We have a small committed minority within the Logo community. We do not have a critical mass.

Stephen Jay Gould: From the theoretical mind of Stephen Jay Gould, we understand cultural change in terms of "transformation." Change is preceded by a slow accumulation of stresses that the culture resists until a breaking point is

reached, moving the culture to the next higher level of transformation.

Learning in a computer culture in 1986 is learning in a changing culture. I would say we are a culture in transition, not revolution. We experience resistance, but the more things change ... the more things change!

Logo teachers see change. What do they see? They "see" students thinking. The following learning behaviors—there are eight—were culled from teacher observations and anecdotal reports.

Students talk to themselves and others.

The act of languaging one's thoughts is a form of verbal mediation, a way of mediating and clarifying understanding.

Students move.

Heads, hands, shoulders move with the turtle. Syntonic learning appears to be firmly related to how children process their worlds.

Students draw upon mental images.

Memory is grounded in words *and* images. Visual imagery facilitates the storage and retrieval of information and instruction.

Students regulate their work.

Casual explorations with the turtle give way to conscious control. Self-correction and -evaluation are related to the sense of empowerment and control that we see in mature learners.

Students look for and create patterns.

Pattern recognition is a basic strategy of organization and comprehension at the metacognitive state of learning awareness.

Students use Logo procedures as building blocks.

The process of breaking problems into meaningful pieces of deconstruction has long been associated with formal problem solving.

Students compose and create.

Artists describe the process of composition as impressionistic and generative. Ideas seemingly



emerge, one idea triggering or guiding the next one, the subconscious driving the conscious.

Students present.

Young people enter the adult world of presenters with confidence and poise. The work they present is their own.

These behaviors tell us something. Collectively, they give us an extraordinary new way of looking at learning. It may not be a revolution, but a new culture is emerging.

We call it a Logo culture. We are too close to it to fully understand it. The impact on learners and learning environments involves complex relationships. We have yet to tease out the nuances and subtleties.

This is what I see:

Stronger learners ... stronger in the sense of learners who use their intuition, who claim ownership of their learning, and are thus free to risk in order to learn more. I see students using learning modalities that are not tapped by the present curriculum.

Real work ... a work intensity that is satisfying and worthwhile; a work ethic that encourages colearning. Gender roles and age differences are blurred.

Expanded vision ... in the shape of new visual forms that expand our vision and guide us toward a clearer understanding of our changing world. Throughout history, artists have operated at the edge of social and cultural change. The fluid rhythms and transformations of Logo images are visual expressions of a new aesthetic.

Thank you, Brian and Bill, for pushing us to new levels of consciousness and concern.

I leave you with the words of the poet Robert Frost: "I bid you a one-man revolution."

Sincerely,

Geraldine Kozberg
Saint Paul, Minnesota

The above letter is dated August 10, 1986.

Ten years have passed.

My enthusiasm and support for Logo have not diminished. I still see stronger learners, real work, and new visions, but the realities of our schools are harsh and compelling. There has been no revolution. The promise of systematic change is yet to come as the power of Logo waits its turn.

It is time for another letter. This one is addressed to you, the larger Logo community.

June 14, 1996

Dear Friends,

Whatever *did* happen to the revolution? The conditions for revolution were not present: there was no tolerable level of discontent, no critical mass, no breaking point in the larger culture; some cracks, perhaps, but no breaking point.

Revolution? In my school district we talk about test scores, violence, poverty, immigration. Technology? We talk about computers and the Internet. We buy computers and we inventory them.

The Logo community has been unable or unwilling to confront the larger social issues that are tearing at public education. In 1981, I wrote: "Logo is one part of a larger change effort designed to serve as an intervention in learning and learning environments."

For the most part, this has not happened. The problem is not the technology, certainly not Logo. The problem is one of equity. Logo is for all kids, but the kids who need Logo the most have no access to it. They are relegated to educational games and instruction in the basic skills.

Logo remains the province of a small, somewhat special group of schools, teachers, and students:

- Teachers who are sophisticated learners themselves
- Students identified as gifted and talented
- Middle class and upper middle class populations
- Private and suburban schools

Logo is for all kids. That does not mean all kids are the same, or that the Logo experience is the same for all kids. Rather it talks to the power of Logo as a language of learning, an instructional process appropriate for a wide range of learning behaviors.

Which brings me to my second point, learning and learning environments. We have not done a good job of helping teachers understand the learning process—not Logo, but learning itself. We have not paid serious attention to the social determinants of achievement, focusing instead on cognition and the technology. For some students, social and behavioral skills may be a prerequisite to classroom participation and learning.

Papert taught that the context for human development is always a culture, never an isolated technology. School is a social institution and Logo understands the principles of learning within the culture of the school. Learning takes place in a social context. Learning is grounded in our social and cultural histories. Why, then, doesn't Logo learning transfer? Why doesn't the integrated learning of Logo pervade the culture of the school?

In Czarist Russia, Jewish children were not allowed to attend the public schools. When Marc Chagall was thirteen, his mother bribed a local school official and Chagall entered the secular, academic world. In his autobiography, Chagall remembers a wondrous time in his geometry class of "lines, angles, triangles, squares.... They carried me far away to enchanting horizons. And during those hours of drawing, I lacked only a throne."

Piet Mondrian, the Dutch artist. When we examine the structural evolution of Mondrian's work, we begin to understand the deconstruction of the process into its most elemental forms. Reconstruction takes on different forms.

What could be more relevant for today's young people than the visual learning of Chagall, Mondrian, Logo?

The findings of Logo studies in Saint Paul suggest that improved achievement is more likely to occur at the lower end of the scale. Kids at the

top already have their set of "gears" to work with. Kids at the bottom are still looking. The odds are that when they find their "gears," the learning process will be strengthened. It may be that Logo facilitates the development of natural learning systems, and especially for those youngsters who have not had the freedom to make learning connections in their own unique ways.

On Monday, we begin our 15th year of continuous Logo teacher education in Saint Paul. I hope the next 15 years will concentrate on issues of equity and the dynamics of learning. Some will argue that I am asking too much of Logo. Not so. I would ask why have we lost the educational context to technocentric commercial interests?

Logo takes on social and political perspectives as schools search for solutions that will act upon the forces that create and maintain inequity in public education.

At the end of the year, we plan to open five Logo centers in Cambodia, a small country half-way around the world, ravaged by war and social upheaval. Phnom Penh ... Battambang ... Siem Reap ... Takeo ... Ratanakeri. Strange and exotic sounds that translate into people and places.

Phnom Penh, a city of one million. The Logo center will be housed in a high school serving 6,500 students in three shifts.

Siem Reap, a rice village, site of the ancient ruins of Angkor Wat, on the edge of the jungle hiding Pol Pot and the Khmer Rouge guerrillas.

Ratanakeri, a remote mountainous province bordering Laos and Vietnam, accessible only by air. Electricity was installed last year.

Our challenge will be to apply the lessons of the first 15 years: helping teachers understand the learning process, learning environments, and why Logo is for all kids.



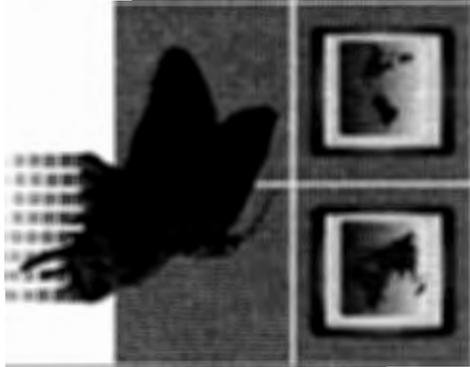
I end this letter with a story:

In old Cambodia, the people tell of three monks who were looking at a banner flying from the top of their temple. The first monk said, "The banner is moving." The second monk said, "No. The wind is moving." The third monk said, "It is the mind that is moving."

I heard this story is 1988 in a makeshift temple in a refugee camp on the Thai-Cambodian border. The image of "the moving mind" helped me transcend the suffering and deprivation of the camp, and moved me to focus on the wonder of the human mind and spirit. The mental image of the mind in motion has stayed with me. It is a fitting metaphor that helps me understand Logo and learning in a computer culture.

Love,
Gerry

▲
Geraldine Kozberg has been involved in issues of equity in public education since the early days of the civil rights movement. She has served as a teacher and administrator in the Saint Paul Schools since 1962. Her experiences in teacher training and program design lead her to concentrate on the forces that mediate substantive change in education for disenfranchised populations. A central and unifying force is the integration of Logo as a reform technology.



TEL•ED '97
Taking Flight In the Digital Age 

Austin, Texas, USA  Mexico City, Mexico
November 13-16, 1997

THE MIGRATION BEGINS WITH YOU.

ISTE'S SIXTH INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS AND MULTIMEDIA IN EDUCATION is migrating to Austin, Texas and Mexico City, Mexico next November — and as an education stake holder of the '90s who understands the dramatic, positive changes interactive technologies have made in the classroom, you should be too.

Exponential growth on the Internet. Transformations in the use of the World Wide Web. These developments have made a revolutionary impact in education over the past decade — literally opening up the classroom walls to new collaborations and innovative ways of communication.

ISTE's Tel•Ed has been on the forefront of this educational metamorphosis since 1989, providing practical solutions to better integrate multimedia and telecommunications into our diverse learning environments. ISTE's multisite Tel•Ed conference in 1997 will offer hands-on workshops, classroom-tested sessions, professional networking, a cutting-edge trade show and much more, inspiring you to take your students of today into the learning opportunities of tomorrow.

So don't let the Tel•Ed migration pass overhead. Spread your wings and "Take Flight in the Digital Age."



For more Tel•Ed information or to receive a free *Advantage Program* (to be mailed summer '97), please contact ISTE:

Phone: 541/346-2472 • Fax: 541/346-5890
WWW: <http://isteonline.uoregon.edu>
Internet: laurie_thoraley@ccmail.uoregon.edu

Presented By:
International Society for Technology in Education (ISTE)
Texas Computer Educators Association (TCEA)
Instituto Latinoamericano de la Comunicación Educativa (ILCE)



Optical Illusions

by William J. Spezeski

Optical illusions are images that fool the eye. They are used to demonstrate anomalies in our visual perception. The most common optical illusions demonstrate the eye's weakness in perceiving length, size, color, and lines when it is distracted. The distractions are other parts of the image that are extraneous to the task at hand.

Recreating common optical illusions with Logo is not only interesting for providing a problem-solving challenge, but also for proving that the image is, in fact, just an illusion. Lengths, directions, and sizes created by Logo instructions are irrefutable.

These procedures are written using PC Logo for Windows. See the end of the article for conversion tips for other versions.

Comparing lengths

The first illusion (Figure 1) is a pair of lines of equal length drawn by procedure **lines**. The arrowheads on the ends of the lines provide the distraction.

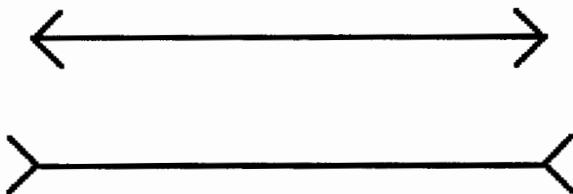


Figure 1

The instructions that draw the two lines appear underlined in the procedures for emphasis. In addition, the last instruction (commented out) draws extra lines that aid the eye in deciphering the correct information.

```
to lines
cs ht
pu setxy [-200 50] pd
rt 45 fd 30 bk 30 rt 90 fd 30 bk 30
lt 45 fd 400 ; draw line 1
lt 135 fd 30 bk 30 lt 90 fd 30
pu setxy [-200 -50] pd seth 0
lt 45 fd 30 bk 30 lt 90 fd 30 bk 30
lt 135 fd 400 ; draw line 2
lt 45 fd 30 bk 30 rt 90 fd 30 bk 30
; seth 0 fd 100 lt 90 fd 400 lt 90 fd 100
end
```

The second illusion (Figure 2) consists of a pair of lines of equal length as well, drawn by procedure **trapezoids**. Each line in question is the top part of a trapezoid. The distraction here is the different shape of each trapezoid.

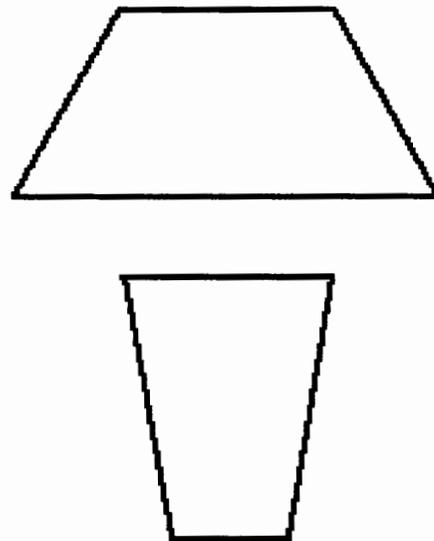


Figure 2

The instructions that draw the two lines appear below underlined in the procedures for emphasis. In addition, the last instruction (commented out) draws extra lines that aid the eye in deciphering the correct information.



```

to trapezoids
cs ht
pu setxy [-40 100] pd
seth 90 fd 80 ; top of trapezoid 1
rt 60 fd 80
rt 120 fd 160
rt 120 fd 80
seth 180 pu fd 100 pd
seth 90 fd 80 ; top of trapezoid 2
rt 100 fd 100
rt 80 fd 45
rt 80 fd 100
; seth 0 fd 100 rt 90 fd 80 rt 90 fd 100
end

```

Comparing relative sizes

In the next illusion (Figure 3), two circles of equal size (radius) are drawn by procedure **circles**. They are surrounded by distractions—larger circles in one case (left) and smaller circles in the other (right).

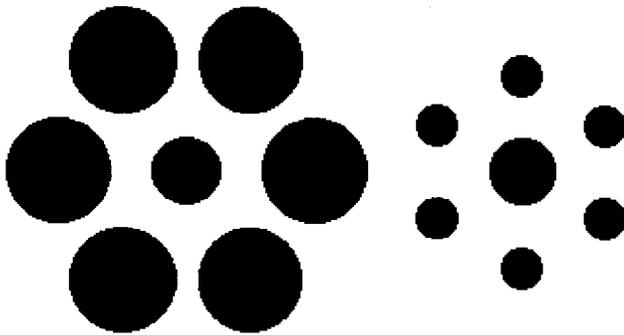


Figure 3

The instructions that draw the two circles of equal size appear underlined in the Logo procedures for emphasis. In addition, the last instruction (commented out) draws extra lines that aid the eye to see that, in fact, the circles in question have the same radius.

```

to circles
cs ht
pu setxy [-125 0]
rt 30
repeat 6 [pu fd 100 pd (stampoval 40 40
"true) pu bk 100 rt 60]
pd (stampoval 25 25 "true)
pu setxy [150 0]
rt 30
repeat 6 [pu fd 75 pd (stampoval 15 15
"true) pu bk 75 rt 60]
pd (stampoval 25 25 "true)
; seth 0 fd 25 lt 90 fd 275 lt 90 fd 50 lt
90 fd 275
end

```

Viewing (parallel) lines

The procedure **parallel.lines** (Figure 4) draws eight vertical lines using the **repeat** instruction. The procedure **wave**, on the other hand, provides the distraction, questioning their parallelism by drawing wavy lines.

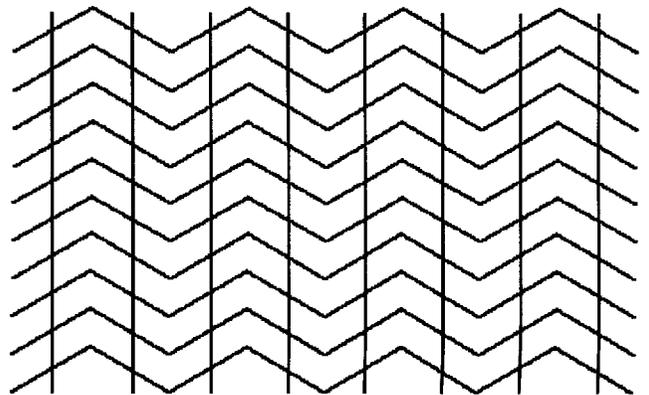


Figure 4

```

to parallel.lines
cs ht
pu setxy [-208 100] pd
seth 60
repeat 10 [wave seth 270 pu fd 416 lt 90 fd
25 seth 60 pd]
pu setxy [-183 -125]
repeat 8 [seth 0 pd fd 250 bk 250 rt 90 pu
fd 52]
end

to wave
repeat 4 [ fd 60 rt 60 fd 60 lt 60]
end

```

Comparing colors

The next two illusions involve color contrast. Procedure `contrast1` (Figure 5) draws two small squares of the same color, each surrounded by two larger squares of different colors.

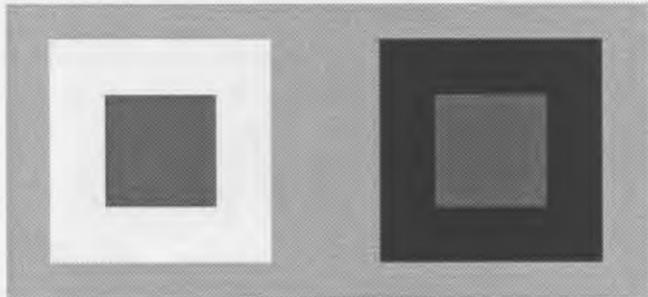


Figure 5

The large white square makes the smaller square inside appear slightly darker, and the large black square makes the smaller square inside appear slightly lighter. However, the bold instructions prove that the small squares are exactly the same color.

```
to contrast1
cs ht setbg 7 ; light gray
pu setxy [-250 -100] pd
setpc 15 ; white
(stamprect 200 200 "true)
pu setxy [50 -100] pd
setpc 0 ; black
(stamprect 200 200 "true)
pu setxy [-200 -50] pd
setpc 8 ; dark gray
(stamprect 100 100 "true)
pu setxy [100 -50] pd
(stamprect 100 100 "true)
end
```

Procedure `contrast2` (Figure 6) draws two small blue-green squares, each surrounded by a larger square, one blue and one green. (You will have to view this illusion on your computer to appreciate the color contrast.)

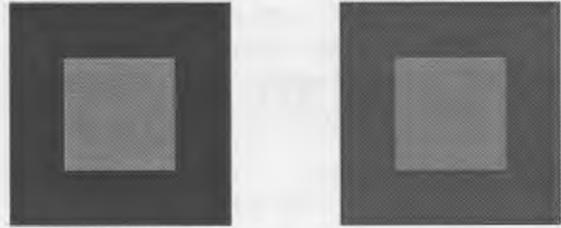


Figure 6

The large blue square on the left makes the smaller square inside appear slightly greener, and the large green square on the right makes the smaller square inside appear slightly bluer. However, the underlined instructions prove that the small squares are exactly the same color.

```
to contrast2
cs ht setbg 15 ; white
pu setxy [-250 -100] pd
setpc 1 ; blue
(stamprect 200 200 "true)
pu setxy [50 -100] pd
setpc 2 ; green
(stamprect 200 200 "true)
pu setxy [-200 -50] pd
setpc 3 ; cyan
(stamprect 100 100 "true)
pu setxy [100 -50] pd
(stamprect 100 100 "true)
end
```

Optical illusions are fascinating. Logo provides not only the means to create the illusion but also the means to evaluate the deception objectively.

Conversion tips

To use these procedures in your version of Logo, you may need to change `setxy` to `setpos` and change `setpc` to `pc`.

The commands `stampoval` and `stamprect` stamp an oval or rectangle on the graphics screen. When the optional input `"true` is provided, the shape is filled with the current pen color.

If you are using Logo PLUS for the Macintosh, change (`stampoval number number "true`) to `stampcirclef number`. The `f` suffix causes the stamped circle to be filled.

If your version of Logo does not include these stamping commands, but does include a **fill** command, you can write **stampcircle** and **stamprect** procedures as follows.

```
to stampcircle :num
  pu lt 90 fd :num rt 90 pd
  repeat 360 [fd (2 * :num * 3.14159) / 360 rt
    1]
  pu rt 90 fd :num lt 90 pd
end
```

```
to stamprect :height :width
  repeat 2 [fd :height rt 90 fd :width rt 90]
  pu rt 45 fd 10 pd
  fill
  pu bk 10 lt 45 pd
end
```

William J. Spezeski is an associate professor in the Computer Science Department at North Adams State College in North Adams, Massachusetts. The department offers an elective course in problem solving that is built around Logo. The course features a variety of problems for computer solution emphasizing top-down design and modular programming. The text for the course is his book, *Logo: Models and Methods for Problem Solving*, recently published by Harvard Associates, Inc.

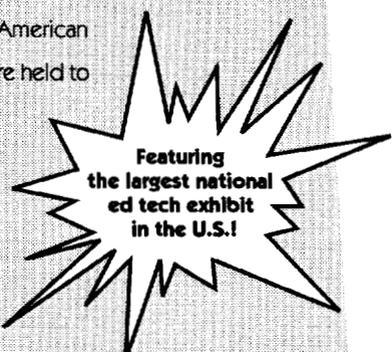
He can be reached at:
 Computer Science Department
 North Adams State College
 North Adams, MA 01247
 Ph. 413/662-5591
 Fax: 413/662-5010
 E-mail: wspeszsk@nasc.mass.edu

N • E • C • C

POTLATCH

@NECC97.seattle.wa.edu

In Northwest Native American culture, potlatches are held to commemorate important cultural events. Tribes from all over come together to reenact legends, raise Totem poles, and present gifts to each other.



We thought the concept sounded a lot like NECC, and so we selected it for the theme of our first Northwest conference—to spotlight the three-day exchange of knowledge, the sharing of new technologies, and the networking among colleagues that we bring you each year in the name of educational technology.

Don't miss your opportunity to accept NECC '97's "Potlatch" gift of knowledge! For the latest information on attending or exhibiting, check us out on the Web or call us toll-free today!

#AD001

	June 30–July 2, 1997 Seattle, Washington
	Hosted by Western Washington University
	Sponsored by the National Educational Computing Association (NECA)

<http://www.wce.wvu.edu/necc97>



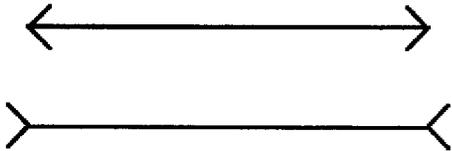
1.800.280.6218



Exploring Optical Illusions in Logo

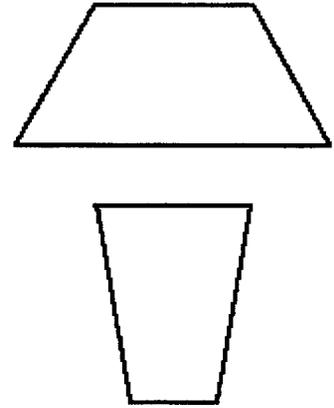
Illusion #1

Which horizontal line is longer?
Can you use Logo to prove it?



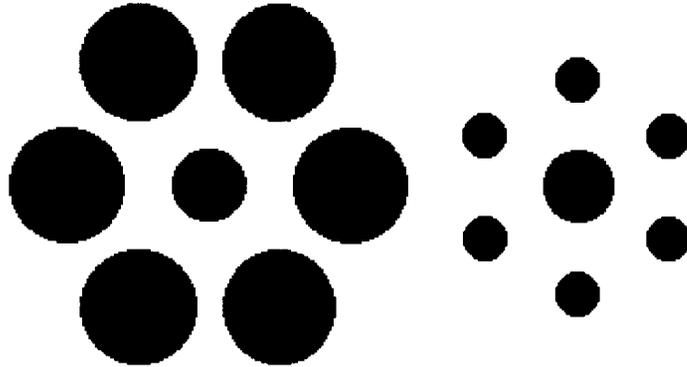
Illusion #2

Which trapezoid has a longer top?
Can you use Logo to prove it?



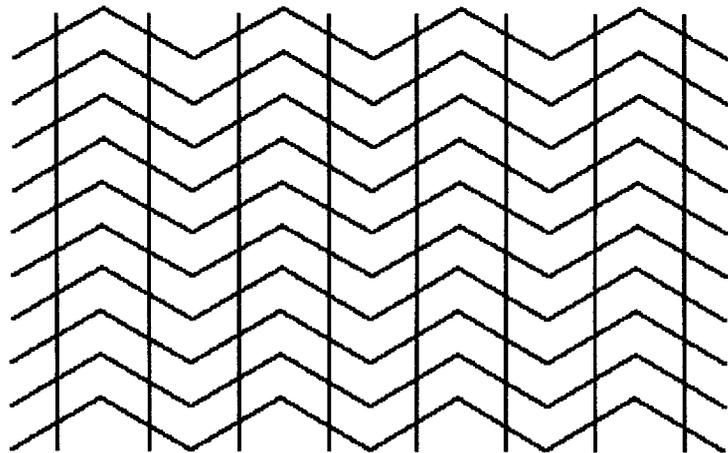
Illusion #3

Which central circle is larger?
Can you use Logo to prove it?



Illusion #4

Are these vertical lines parallel?
Can you use Logo to prove it?



Illusion #5

Now it's your turn! Can you design an optical illusion in Logo?

Learning in a Notebook-Saturated Logo Environment

by Jenny Betts

Introduction

“Learning takes place when the learner takes charge.”

— Papert (1980)

The main focus of this article is to review the teaching strategies I use when working with students who learn in a notebook-saturated, primary classroom. Each student possesses a notebook computer, which becomes a very personal tool that each individual can control in his or her own way.

Students use Logo throughout most of the school day and continue to work on the Logo projects at home each night because they can take the computer home at the end of the school day.

The classroom setting

Picture this: As you make your way through the classroom, you spy two children seated at a desktop computer downloading information from a computer located in another room on campus (networking). A small group huddles around a large monitor, deciphering code (LogoWriter computer programming) and trying to unravel the puzzle of why their graphics and text are being displayed out of sequence. Others are in pairs, typing frantically on their own computers, which are positioned on their laps, while they discuss how to write a code that will have the computer ask a question and then respond appropriately, depending upon the answer given by the user.

For a moment you are distracted as a few children brush by you heading towards the library, each with a laptop computer tucked under an arm (portability). Two more students are gathering information from a CD-ROM, which is located elsewhere on the campus (networking). They can access this CD-ROM bank by attaching the notebook computer to the access point within the classroom.

Some children gather around the printers that are placed in various sections of the room. They are producing explanatory and instructional notes for the Logo programs they have been writing, or they are printing out information booklets that have been produced with the aid of a word processor. You see students using a scanner. You can see other students colouring, cutting, painting, or reading, depending on the needs of their projects.

This equipment is at the fingertips of all students and is used throughout the school day, achieving Papert’s vision (1980) of generating a “new kind of learning environment” that allows “free contact between children and computers.” I have no “computer lessons” or “computer times.” The use of this technology has become a tool with which we work throughout the school day. Gone are the days of catching 1 hr here, and snatching 40 min there. Even the regular hour-per-week computer lessons have disappeared.

When working in this environment, I noticed that as students were becoming more proficient with the Logo programming language and more skilful when using other technological bits and pieces, they began to change the ways in which they were carrying out tasks. I then began to wonder how I could empower both myself and the students to function effectively so that we could take advantage of this situation. Allowing students to take more control over the direction in which their learning was heading—that is, being more learner directed—seemed to be the solution. Spending a substantial amount of time during the day on the learner-directed projects became common, and gave students more time to explore and develop their ideas.

Three Pedagogical Issues

- Learning is self-directed. Students are given the opportunity to control the direction in

which their own learning takes place by choosing topics that are of interest to them.

- Papert's idea of "learning how to learn" is frequently applied to the learning situation.
- Emphasis is on the student developing a plan of attack. A great deal of time is spent assisting students with planning their time to work more efficiently, developing strategies to gather and collate information from a large variety of sources, and implementing relevant strategies to aid in organising information, no matter what the content.

Students are responsible for their major topics, any subtopics that are relevant to their topics (Papert, 1993, coined this term as a "connection"), how they organise their information, how they work in a group or individually, how they collect and collate information, and how their information is presented. Therefore, instead of focusing on a particular content area, the focus is placed on developing those skills. Many discussions take place with regard to "how to work smart and effectively."

Of course, it would be unprofessional and downright improper to leave a student alone to learn all that is needed to be learned; therefore, teacher intervention is a critical component of this learning environment.

Where the students start

In this environment, students depend on one another, working together to solve problems, plan outcomes, and achieve desired results. Students become aware of their strengths and weaknesses very quickly and learn to work cooperatively and effectively if they are to reach the goals they have set for themselves. They develop the ability to organise themselves confidently, tackle complex tasks more confidently, and yet, they are not ashamed or embarrassed to seek help from other members of the class who possess skills they lack. Papert (1980) stated that "part of the fun is sharing, ... modifying, and experimenting with each other's work." This is a crucial element within our classrooms and an activity in which we engage frequently.

Perhaps all this is sounding very familiar to some teachers and, possibly, many teachers are

already applying these strategies in their classrooms. However, I believe the major difference for me is that I use this strategy and apply it to all parts of the curriculum and throughout the school day.

Most times, students work on only one major task during the day; sometimes the task may even take the entire week. If this is the case, then the entire week will be devoted to that task. That is, maths is not perpetually tackled each and every day. At times, an entire day or week will be devoted to exploring only maths concepts. Or, an entire day or week may be devoted to exploring language components.

To make it easier for the students to organise their time, it was necessary to put in place some sort of structure for them to follow. We worked through four phases.

1. Planning
2. Action
3. Assimilation
4. Outcome

Phase 1: Planning—What will be done

The initial planning is a huge task, and records are kept when any changes to the student's original plans are made. The topic chosen needs to be open ended so that a thorough investigation, or deep study, can be undertaken. The emphasis is on developing a plan of attack or developing a proposal. Both the major topic and subtopics are listed on the proposal. Other ideas that may be included are presentation ideas for each section, the resources needed, and the job allocation of each group member. Much time is spent helping students develop this plan and make modifications when necessary. The proposal is meant to be very flexible and is never adhered to strictly. It is simply a way for students to organise their ideas formally. It is altered each time a new idea is carried out or an old idea rejected.

The freedom of allowing students to explore topics of their choice does not mean they are left unattended to discover all things that are meant to be learned, which is similar to incidental learning. Children are guided carefully during all phases. I will intervene during the action phase



by suggesting a subtopic that they could include; discussing another method of presentation; showing them a few lines of Logo code that may help; and directing students to books, other software, or even other students for help.

Phase 2: Action—Do it

It is in this phase that we are most active. All group members begin to carry out their designated tasks, either individually or in small groups. All actions undertaken are recorded on the proposal, which in effect becomes a journal.

Actions undertaken include researching, writing Logo code, attending skills lessons (e.g., learning about timelines, interpreting maps, note taking, writing letters, learning how to scan, etc.), scanning appropriate pictures to provide relevant graphics, gathering information, viewing videos, and sorting through magazines.

Intervention by the teacher is necessary during this phase, especially when it is evident that a particular skill is needed by the student so that the task can be carried out efficiently and effectively. At these times the teacher may:

- demonstrate a skill; for example, finding key words in articles or learning how to operate a particular piece of software effectively.
- encourage students to attack a task in different way. I may encourage students to use the idea of a timeline to display information. This timeline may be programmed using Logo or created using pen and paper.
- provide students with resources. I may need to obtain the video, "The best way to see Europe" if the students are including a European flavour in their projects. This video may contain some helpful ideas from a tourist's point of view. Students may include this idea in their "deep study."
- guide the children toward exploring other sub-topics.

You may be saying to yourself that you already teach these skills; the difference lies in the fact that not all students are expected to attend the same skills lessons as their peers. Each student attends the lessons that are relevant to his or her current project.

Phase 3: Assimilation—Discuss progress

All members of a group, or a group of students using the same ideas, meet to discuss the progress being made. Students are asked to reflect upon actions taken and modify plans where necessary. Any modifications are noted on the original proposal. It is during this phase that I carry out the majority of the evaluation. Comments made and modifications of ideas are noted on the proposal. This continual evaluation is very important because it indicates how the students get the job done. Evaluation of the end product plays a minor role in the overall evaluation.

The action and assimilation phases are revisited many times and changes to the original plan are recorded on the proposal either by me or the students. We record such things as lessons attended, suggestions offered by me or another student, problems encountered and solved, and diagrams that were used to illustrate ideas and rough notes. This proposal is a vital piece of documentation for me because it assists during the evaluation.

Phase 4: Outcome—Final Presentation

The proposal, records kept, rough notes made, diagrams constructed, rough plans, lists of resource materials, the final product, and any other pertinent materials are a part of this outcome.

The final production of a project, or as much of it as possible, is displayed and shared with all members of the class.

Where the teacher starts

Not all students will cover the same content, and not all students will be exposed to the same skills. The skills and content that each student experiences depend upon the current project. Time is not strictly scheduled, therefore, one group may be doing art activities while another may be conducting a science experiment. One student may be working in the language area while another student is exploring a maths idea.

Types of Lessons

To identify the types of activities, lessons, skills, or concepts covered, it seemed necessary to use categories.

A. What is the nature of the activity or lesson?

Compulsory **Optional** **Invitational**

B. Who is teaching the concept or skill?

A Teacher **A Peer**

C. What type of activity or lesson?

Individual **Small Group** Whole Class

D. Who initiated the idea, concept, or skill that was needed?

A Teacher **A Student**

Time is spent mostly in the bold-faced areas. Whole-class lessons are rare. Small-group lessons conducted by either me or a student are routine. We spend most of the day working this way.

Teacher-directed activities (compulsory activities) are planned according to the students' proposals. All children are given the opportunity to participate in any of the planned lessons that are categorised as optional activities, regardless of whether it is used for their current task or not. If a child wants to participate "just because," then they are welcome to do so. There are also times when particular students are encouraged to participate (invitational activities).

Devoting so much time to these activities helps achieve the in-depth investigation of a topic. For example, an optional activity on timelines may be undertaken by many students. I do not feel compelled to cover a certain amount of content in a particular period of time simply because I'd rather explore a topic thoroughly than stop because it is 10:00. If it takes the whole day to complete a lesson, then the whole day will be devoted. If a student chooses to spend the next three days working on this idea, then so be it.

Discussion

Before giving students complete freedom to choose their own topics, I found there was a need to model the process. To do this, it was necessary for me to take on the role of group coordinator as the class developed a project. Together we would work our way through the four stages.

For example, one of my classes decided to use a "space" theme, and wanted to build a Logo project. We developed a story about how we were a team from NASA who was about to colonise a new planet. It was decided that our Logo project would be similar to a database where all the planet's information would be stored. Each member of the class took on a particular role such as scientist, pilot, and so on. These roles were recorded on the proposal, alongside the name of our ship, planet, and any other details that all members of our crew needed to know.

As the story unfolded, the role of each crew member became clearer. Our scientists created a map of the planet, our architects designed buildings, security worked very closely with our government officials to develop a set of laws for all new colonists to abide by, our biologists explored the marine life, food technicians studied the plant life, and so on.

Throughout the entire development of the project, I modelled the way in which the proposal should be used, how tasks were to be recorded, and how group members could assist one another. Every day we would meet together as a group (assimilation) to discuss the progress of each individual or small group. Again, the progress was recorded on the proposal.

My focus throughout this time was to model the four stages each group (or individual) is to follow when they undertake their own major tasks and how individual group members need to work effectively within a group. I also used this time to show students how to:

- research a topic thoroughly and diversely.
- use the CD-ROM effectively to search for relevant information.
- download information from the network onto their own computers.
- attach their notebook computers to the large TV monitors to use during small-group work.
- use the network.
- use a large variety of resources.
- import graphics into a report.
- summarise and take notes.
- develop a methodical approach to planning their work as well as making sure that they



adopt the notion that it is all right to change original plans.

I also used this time to:

- offer advice on presentation techniques such as which colours would make an advertisement more effective.
- permit the use of other resources such as craft materials to build 3D models.
- allow students to use wall and floor space in the classroom to the best advantage; for example, setting up a corner of the room or using the pin board to display their findings.
- develop a reference section within the classroom to store wall maps, atlases, road maps, dictionaries, and so on. This is an appropriate time to discuss other types of books and resources that may be needed. We constantly add to the resource area.

Conclusion

I think we would all agree that skills and processes are important, simply because the current curricula are written so that those skills and processes are developed. However, the curriculum is still coercing teachers into spending the majority of their time providing all students with the same opportunities by teaching all students the same content. I do not concern myself with trying to cover all the content matter embedded in the current curriculums; however, I do try to cover all the skills that are required.

Although it may still be necessary at this stage for Year 10–12 to cover certain content areas in secondary schools (to gain entrance into tertiary institutions and meet employer expectations), for the lower year levels, especially primary school students, it may be more beneficial to focus on developing learning strategies and good working habits. Young students should be able to explore, experiment and develop various learning strategies that suit them, then in later years, apply those strategies to mandatory topics studied in senior school. ▲

Bibliography

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books.

Papert, S. (1993). *The children's machine—Re-thinking school in the age of the computer*. New York: Basic Books.

Jenny Betts is the Coordinator for Computing (K–5) at the Elementary School of John Paul College in Australia.

E-mail: jbetts@powerup.com.au

A Simulation Challenge: The Doctors Dilemma

by Jim Muller

People are always asking what you can do with Logo besides draw pictures. One area in which Logo shines is in the exploration of simulations, breaking a complex issue into more easily understandable chunks of information and then mimicking their real-world actions.

- Logo helped a group of junior high students simulate the Battle of Baltimore to show why the British, with their superior firepower, could not defeat the Americans entrenched in Fort McHenry during the War of 1812.
- We simulated the Gun Fight at the OK Corral and found that if Billy Clanton had not missed his first shot at Wyatt Earp, then Tombstone, Arizona, might be a very different place today.
- “Simulations to Get You Started” in this issue describes how simulating a robot can help young Logo fan(atic)s focus on structured thinking.

If you want a simulation to explore on your own, read the following e-mail message I received from a doctor in the United Kingdom.

I am interested in writing a program that simulates the behavior of the growing tip of a brain cell called a growth cone. The growth cone responds in characteristic ways to various objects it encounters. For example, if it encounters another nerve fibre it often grows along it. However, nerve fibres come in different flavors and often a growth cone chooses between a number of nerve fibres when it comes to an intersection. Another characteristic is that it can detect gradients of different molecules and can grow toward an increasing concentration of such a molecule.

I am interested in making the turtle mimic the behavior of the growth cone: make it grow along a line that it encounters, perhaps choosing a line of one color

over another. Or make it grow toward an increasing concentration of dots of a given color or away from dots of another color.

I am a complete novice at both programming in general, and Logo in particular, but Papert’s book has inspired me to learn and give this a shot anyway! Do you know of anyone who may have similar interests or of programs that deal with how the turtle responds to objects it encounters? Thanks for your patience with this long message!

This problem is reminiscent of the mouse-in-a-maze problem. If the mouse continually turns right every time it comes to a wall, it will eventually find its way out. Because this worked for a maze, a first step toward finding the appropriate nerve fiber seemed to be to have the turtle turn right and move forward until it found a nerve fiber.

The first line of the second paragraph of the message adds a new twist to the maze exercise, however: Make the turtle grow along the nerve fiber that it encounters, perhaps choosing a line of one color over another.

The first task was to create three random groups of red, blue, and green cells.

Each group was randomly placed on the screen. The next task was to have the turtle find the red cells and attach a series of brown cells, while ignoring the blue and green cells. These tasks are addressed by the **cells** procedure below.

Turning right all the time is fine for a robot mouse, turtle, and probably even a nerve fiber. Real animals and cells do not follow such rigid rules. They tend to move, evaluate their position, and then move again. To make the search a bit more realistic, the variable `⍺` refers to a random turn, and the variable `⍟` refers to a right turn.



To get started, try these combinations:

```
cells 10 30
cells 30 60
cells 15 75
```

If you want to see how the mathematical simulation works, trace the procedures to follow the turtle step by step.

```
to cells :r :s
cs setup
seek
end
```

```
to setup
;Creates a random list of the color numbers
12, 10, and 9. Change the numbers as
required to represent the colors red,
blue, and green.
make "x []
make "x shuffle [12 10 9]
make "color1 first :x
make "color2 first butfirst :x
make "color3 last :x
cellgroupa
cellgroupb
cellgroupc
st setpc 6
end
```

```
to shuffle :deck
; Randomizes the input list without repeat-
ing any of the elements.
if empty? :deck [output :x]
if (random 2) = 1 [make "x fput (first
:deck) :x] [make "x lput (first :deck)
:x]
output shuffle butfirst :deck
end
```

```
to cellgroupa
;Randomly places the first group of cells on
the screen and defines the group's coordi-
nates as X1 and Y1.
make "x1 random 250 - random 125
make "y1 random 160 - random 60
pu ht setxy list :x1 :y1 pd
setpc :color1
line1
make "a pos
pu setxy list 100 -50 pd
if :color1 = 12 [make "f list :x1 :y1] ;de-
fines :f if :color1 is red.
end
```

```
to cellgroupb
;Randomly places the first group of cells on
the screen and defines the group's coordi-
nates as X2 and Y2.
make "x2 :x1 + 30
if or :x2 > 150 :x2 < -150 [make "x2 50]
make "y2 :y1 + 30
if or :y2 > 100 :y2 < 70 [make "y2 0]
pu ht setxy list :x2 :y2 pd
setpc :color2
line2
make "b pos
pu setxy list 100 -50 pd
if :color2 = 12 [make "f list :x2 :y2] ;de-
fines :f if :color2 is red.
end
```

```
to cellgroupc
;Randomly places the first group of cells on
the screen and defines the group's coordi-
nates as X3 and Y3.
make "x3 :x2 + 30
if or :x3 > 150 :x3 < -150 [make "x3 100]
make "y3 :y2 + 30
if or :y3 > 100 :y3 < 70 [make "y3 50]
make "c pos
pu ht setxy list :x3 :y3 pd
setpc :color3
line3
pu setxy list 100 -50 pd
if :color3 = 12 [make "f list :x3 :y3] ;de-
fines :f if :color3 is red.
end
```

```
to seek
;Determines if the red cell group has been
found. If so, cells are grown around the
red group.
find :r :s :f
if distancepos :f < 12 [seek] [grow]
end
```

```
to find :r :s :f
(local "ans "ans1)
pu fd 8 rt :r - 2 * random :r
make "ans le :f
make "ans1 re :f
if and :ans :ans1 [fd 8 stop]
if le :f [lt :s stop]
if re :f [rt :s]
end
```

```
to grow
;Grows cells around the red cell group
if :f = list :x1 :y1 [seth :h1]
if :f = list :x2 :y2 [seth :h2]
if :f = list :x3 :y3 [seth :h3]
pd seth heading + 90
repeat 10 [repeat 12 [fd 1 rt 30] pu rt
random 180 fd 7 pd]
end
```



```

to line1
;Creates the first group of cells
make "h1 0 - random 15
seth :h1 + 90
repeat 10 [repeat 12 [fd 1 rt 30] pu rt
  random 180 fd 7 pd]
end

to line2
;Creates the second group of cells
make "h2 0 - random 30
seth :h2 + 90
repeat 10 [repeat 12 [fd 1 rt 30] pu rt
  random 180 fd 7 pd]
end

to line3
;Creates the third group of cells
make "h3 0 - random 45
seth :h3 + 90
repeat 10 [repeat 12 [fd 1 rt 30] pu rt
  random 180 fd 7 pd]
end

to le :f
local "ans
make "ans br :f
if or (:ans < 90) (:ans > 340) [output
  "true]
output "false
end

to re :f
local "ans
make "ans br :f
if or :ans < 20 :ans > 270 [output "true]
output "false
end

to br :f
local "ans
make "ans 360 + heading - int towards :f
if :ans < 360 [output :ans]
output remainder :ans 360
end

```

Read the doctor's letter carefully. You'll find a number of challenges to explore.

1. Make the growth cone grow toward an increasing concentration of dots of a given color or away from dots of another color.
2. If it encounters another nerve fibre it often grows along it. However, nerve fibres come in different flavors and often a growth cone chooses between a number of nerve fibres when it comes to an intersection.

3. Another characteristic is that it can detect gradients of different molecules and can grow toward an increasing concentration of such a molecule.

Who said learning couldn't be fun?

Conversion Tips

These procedures have been written in PC Logo to accommodate as many dialects of Logo as possible. Those who use **then** and **else** in **if** instructions will need to use the syntax:

```
if expression then action1 else action2
```

The **setxy** command has been written as **setxy list num1 num2**. You may need to either remove the command **list** or replace **setxy** with **setpos**, depending on your version of Logo. You may also need to replace **and** with **allof** and **or** with **anyof**.

The commands **pos**, **remainder**, **distancepos**, and **sq** are offered below as procedures for those who do not have these commands available as primitives.

```
to pos
output list xcor ycor
end
```

```
to remainder :aa :bb
make "cc :aa / :bb
make "dd (:cc - int :cc) * :bb
output :dd
end
```

```
to distancepos :f
output sqrt sum sq ((first :f) - (first
  pos)) sq ((last :f) - (last pos))
end
```

```
to sq :a
output :a * :a
end
```

Read about **Jim Muller** at the end of his other article in this issue, "Simulations to Get You Started." Editor's note: For an electronic copy of this program (PC Logo version), send an e-mail message to: logoinfo@harvassoc.com. In the subject line, put: **request cells.lgo**

This automated e-mail reply system will send you a text file containing the program. For an index of all available programs and memos, put **request info** in the subject line.



Logo: Search and Research

Logo in the Classroom: A Tale of Three Teachers

by Julie Sarama and Douglas H. Clements

“As the births of living creatures, at first, are ill-shapen: so are all Innovations.”

—Francis Bacon

In our last column, we considered the general question: How can educators implement computers well? Although general research suggestions are helpful, we need to look at specific classrooms. This was our goal when we set out to study how a group of teachers implemented our own Turtle Math™ (Clements, 1994) version of Logo. The individual variations were so profound, we began to question the very notion of “systemic reform.” Let us share some of their story with you.

Story Starter

When we created Turtle Math, we worked closely with teachers. While we were finishing the last of the activities, a friend of ours asked us about innovative mathematics curricula. Ardie was a whole-language specialist and consultant and wanted to help her teachers try a “whole math” approach. We told her about our vision of Logo and Turtle Math. She liked the idea.

Turtle Math as enrichment

To get the ball rolling, Ardie interested two teachers in working with some students after school. We helped the teachers get started with Turtle Math. They did not request any further help and reported that the program was a success. Teachers and students enjoyed and learned from their work. Parents asked for more.

Turtle Math as curriculum innovation

In fact, the experience was so positive that Ardie and the principal asked us about the possibility of implementing it throughout the school. They recruited several teachers. We agreed enthusiastically, for two reasons. First, the teachers could help with the final field testing. Second,

we believed that this could be a fruitful area for the exploration of how teachers implement an innovation and how their philosophies are affected by the implementation. So, we decided to study the process (Sarama, 1996).

The interested teachers met with us several times. They appeared enthusiastic, and we planned how to get started. Mr. Barrett was provided to the teachers on an as-needed basis. He was available for support during and after school.

Rocky road

Despite initial enthusiasm, teachers’ use of Turtle Math began dropping off. Ardie had been transferred to another position. Although Mr. Barrett was still available for help, we believed that a third party might form a different relationship with the teachers and, thus, uncover valuable information about the teachers’ perspectives about Turtle Math. So, we asked a colleague, Julie Jacobs Henry, to conduct a series of semistructured interviews with teachers who had been involved with the project.

What Happened?

The theme of this story is unrealized potential and missed opportunities for facilitation. A major cause was divergent beliefs of different people. So, to tell this part of the story, we have to look through the eyes of different groups of people. How did they view factors that affect implementation—factors such as those we discussed in our previous article?

Computer access

The principal and the teachers initially thought computer access was adequate. They were all proud of the new lab, which provided a display area for the new technology in which parents and administrators could easily see the school’s progress. Issues of equity were suppos-

edly solved because all the computers and the printers were in one room and the teachers had equal access to the room. When the teachers became more involved with using the computers, they realized that one or two blocks of time per week represented inadequate access. We communicated this concern to the principal. The principal told us that there were free periods (not assigned) that were available for teachers to sign up if they wished.

Support for teaching reform

We as designers believed that the school would continue to provide someone (Ardie or someone like her) to help the teachers integrate the software and alter their approaches to teaching mathematics, as had been done with language arts. The principal believed that innovation had already been supported in the whole-language context and therefore, continuing the same level of support was unnecessary. She viewed the teachers as professionals able to facilitate each other's growth without outside help.

Our research conclusion was that teachers need continuing support in both integrating technology and reflecting on their mathematics teaching. After the research team met with the fifth-grade teachers specifically for the purposes of planning to integrate Turtle Math activities into specific lessons, the fifth-grade teachers experienced the most success. It should be noted that the teachers did not ask for this specific meeting, but it accomplished more than the numerous training sessions.

Technical support

Mr. Barrett was available to the teachers after school as often as they wished, but attendance at these after-school sessions dwindled. The teachers preferred to have the software to "play around with" on their own time. However, the teachers did not take advantage of the written materials that accompanied the software. This resulted in their teaching the activities in ways not consistent with the goals of Turtle Math. So, although technical support was available, the level at which it was used was inadequate for successful integration.

External validation

When Ardie was transferred, the teachers did not receive the continued support from the administration that they expected. This was perhaps the most unfortunate turn of events. We still hoped that validation from each other, observations of students' growth, and we as university professionals would be sufficient for the teachers. Mr. Barrett, being a part of the university team, believed that his positive feedback would also validate their work. The principal believed that the teachers, as professionals, offered each other validation and that no other feedback was necessary. However, it appeared that all of these viewpoints were inaccurate.

Near the end of the year, the district math coordinator visited the fifth-grade rooms to teach a geometry lesson. The students at this school did well, and the mathematics coordinator complimented their performance:

Yesterday (the mathematics coordinator) came into my classroom and started speaking to the children about geometry ... and she was amazed how much they knew ... not just knew, but understood more of the concepts (Ms. Moore).

For these teachers, the math coordinator's comments validated the usefulness of the program. This validation by someone at the district level (an authority not connected with Turtle Math) accomplished more than positive comments by the project staff. When this was suggested to the principal, she did not agree that this was important. She felt that the teachers could offer each other such validation.

Time

During the course of this study, the teachers frequently mentioned that they wished that they had more time to explore the software. Mr. Barrett believed that he could ease the time pressure by helping schedule the teachers in the computer lab, as well as generally being available when the teachers were teaching. The principal believed that time for inservice training and after-school sessions would suffice. However, these conditions were inadequate.



Multiple reform efforts

Multiple simultaneous reform efforts overwhelmed the teachers. During one meeting, they named *ten* different reforms they were implementing *that year*. The principal, conversely, believed that she was serving her faculty's needs by offering them a variety of opportunities for reform.

Some Morals of the Story (before intermission)

From the research of others, we had known that many factors affect implementation of any educational innovation. As we concluded in our previous column, we, like most implementers, underestimated the complexity and difficulty of dealing with all these factors. The most pervasive and powerful influence, however, was the different perspectives on these factors that were held by the people involved. For now, we draw implications from those parts of the story we've already told.

Computer access

For example, consider the seemingly straightforward issue of "computer access." Everyone agreed it was important. But teachers and administrators disagreed about what "adequate access" meant. So, even with adequate material resources, this essential problem remained unsolved.

Support for teaching reform

Similarly, different views weakened the effectiveness of available support for teaching reform. The principal's notion that support was already provided was not valid from others' perspectives. Also, the support provided was not optimal.

First, there was a subtle but important difference between Mr. Barrett's ideas of modeling the innovation and those of the teachers. Staff believed that they were modeling an innovation that included a philosophical stance, a pedagogical approach, activities, and software tools. Teachers saw a model only of the latter two. This difference needs to be addressed.

Second, teachers would have benefited most from meeting a support person frequently, for a short period of time (as Ardie had) to plan activities and provide time for reflection. In such meetings, special care should be taken in establishing connections between the innovation and the teachers' regular mathematics program. Teachers need these connections as a springboard. However, if too much is made of such explicit connections, teachers may limit their vision of the innovation to a surface-level extension of traditional mathematics curricula.

Technical support

We believed that teachers needed to understand the goals of the software and activities. Teachers preferred to "play around." We conclude that a synthesis of these two approaches is necessary. Teachers should have the time and technical means to play with the software on their own; however, without appropriate guidance (e.g., written materials), "playing" may be unproductive.

External validation

The teachers were not sure of themselves until a district supervisor observed their classes and confirmed that the innovation was worth their investment of time and energy. Our conclusion is that teachers attempting an innovation need administrative feedback to complement peer validation and the internal validation of observing students' growth. Each plays a distinct and necessary role in sustaining the teachers' use of the innovation. Although the administrators' model appears consistent with a constructivist philosophy, the reality of this school environment provided conflicting messages about locus of control and independent decision making for teachers.

Time

More time always helps. We found, however, that the quantity of time spent with the software is less important than the way the teachers' time is spent. We believe that the best use of personnel time would be to have the teachers meet regularly with an administrative assistant, such as Ardie. Giving the teachers time for reflection is important, but the teachers in this

study did not reflect on their experiences and needs until they were interviewed. Time spent conversing with such an assistant could facilitate the desired reflection. Time with the software also needs to be scheduled so that the teachers can not only have time to explore the software, but work with someone on integrating the software activities into their mathematics curriculum.

Multiple reform efforts

Making matters worse, the teachers were inundated with an overwhelming number of innovative programs or practices. Offering multiple reform efforts without ensuring a consistent, coherent theoretical base and integrating their implementation is a course doomed for failure. Although the reform efforts themselves are nonreductionist, the method of implementation—multiple, separate one-day workshops—was reductionistic. The teachers then do not see the forest for the trees.

Although a program intended to increase higher order thinking skills, a hands-on science curriculum, and Turtle Math have consistent and overlapping goals, the teachers had difficulty seeing this and, thus, implementation was difficult. Although the teachers could see the connections between some of the innovations, there was no effort on anyone's part to integrate them or choose among them.

Therefore, those planning an innovation may wish to ascertain what other reform strategies are being implemented and to work hard and early to establish a consensus that the new innovation (a) is seamlessly consistent with other reform efforts, (b) will accomplish the same objectives as these other efforts and can therefore replace them, or (c) has a higher priority than these other efforts and therefore the staff will put them aside for a time.

Words of advice

Overall, a basic implication for administrators is that they should remain fully involved with the project (e.g., attend project meetings). They might also plan to help teachers integrate the demands of multiple reform movements and provide the time for teachers to experiment, dis-

cuss, and, in general, construct their own meanings of constructivist pedagogies.

There are also implications specific to computer-based innovations. Computer laboratories may hinder the full use and integration of innovations. In contrast, even a small number of computers in the room seems to put teachers in control of the innovation, giving them responsibility for integrating it into their curricula.

Those designing innovative computer materials, and implementation programs for them, should be aware of the tendency for teachers to ignore documentation. Instead, teachers often run the program, apparently expecting that the full extent and worth of the software and the activity will be immediately obvious. In general, designers of innovations must continually strive to see things from the teachers' point of view.

In every case, different perspectives made implementation complex and challenging. In fact, different beliefs about mathematics, teaching, and technology were at the heart of our tale. But that is a story unto itself, and we shall tell it in our next column. ▲

References

- Clements, D. H., & Meredith, J. S. (1994). *Turtle math* [Computer program]. Montreal, Quebec: Logo Computer Systems, Inc. (LCSI).
- Sarama, J., Clements, D., & Henry, J. J. (1996). *Network of influences in an implementation of a mathematics curriculum innovation*.

Time to prepare this material was partially provided by a National Science Foundation Research Grant, "An Investigation of the Development of Elementary Children's Geometric Thinking in Computer and Noncomputer Environments," NSF MDR-8954664. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Douglas H. Clements, professor at the State University of New York at Buffalo, has studied the use of Logo environments in developing children's creative, mathematics, metacognitive, problem-solving, and social abilities. Through a National Science Foundation (NSF) grant, he developed a K-6 elementary



geometry curriculum, *Logo Geometry* (published by Silver Burdett, & Ginn, 1991). With colleagues, he is working on the previously mentioned NSF research grant and 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 Logo for learning elementary mathematics. One, Turtle Math, is presently available from LCSi.

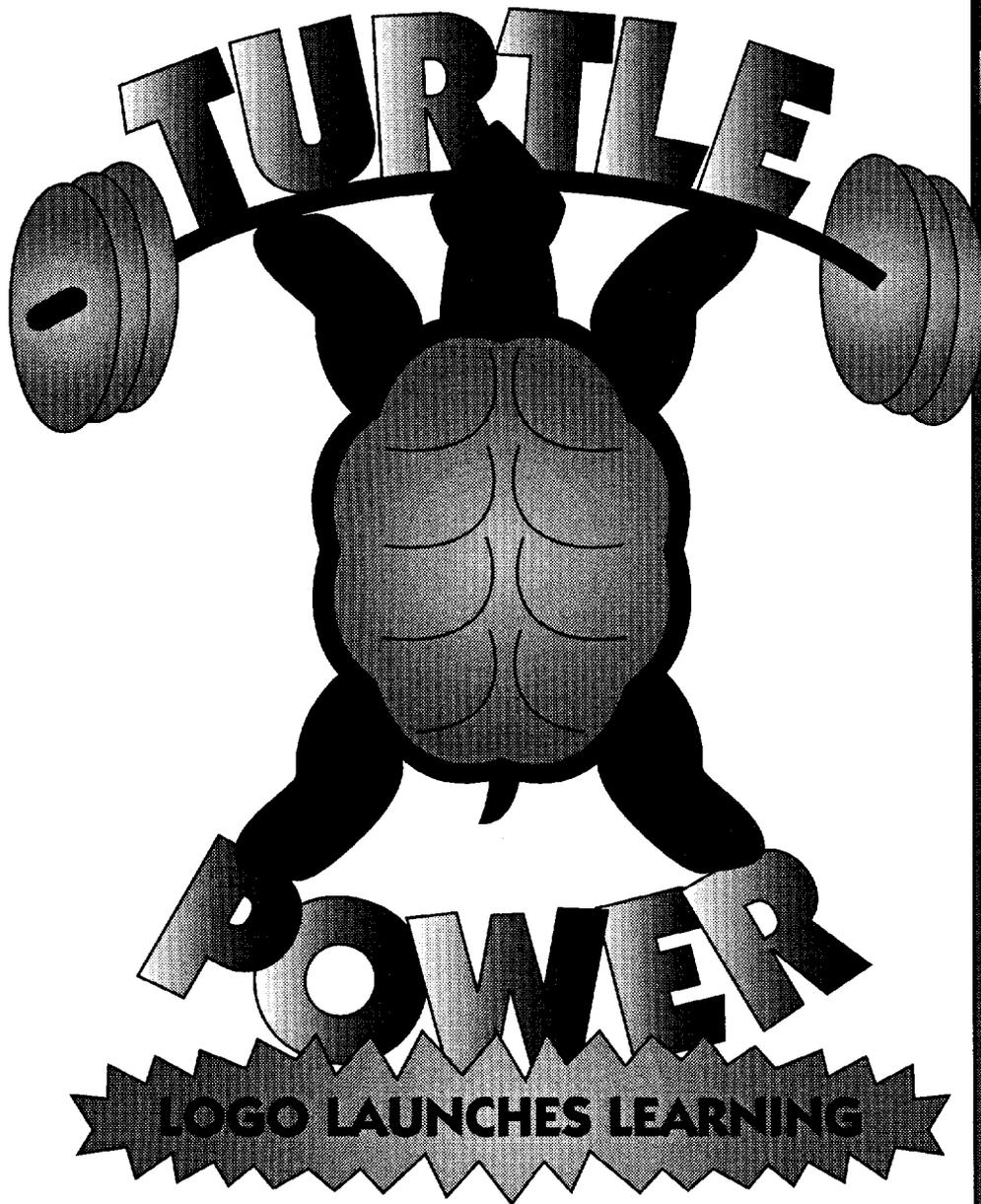
Julie Sarama, PhD, is an assistant professor at Wayne State University. Previously she has taught secondary mathematics and computer science, gifted math at the middle school level, and mathematics methods courses. She is coauthor of several *Investigations* units and of Turtle Math and is currently designing and programming new versions of Logo and other computer microworlds.

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

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

Statement of Ownership, Management, and Circulation (Required by 39 U.S.C. 3685). 1. Publication Title: Logo Exchange. 2. Publication No.: 0888-6970. 3. Filing Date: October 18, 1996. 4. Issue Frequency: Quarterly. 5. No. of Issues Published Annually: 4. 6. Annual Subscription Price: \$34.00. 7. Complete Mailing Address of Known Office of Publication: ISTE, 1787 Agate St., Eugene, Lane, OR 97403-1923. 8. Complete Mailing Address of the Headquarters or General Business Offices of the Publisher (Not Printer): ISTE, 1787 Agate St., Eugene, OR 97403-1923. 9. Full Names and Complete Mailing Addresses of the Publisher, Editor, and Managing Editor: Publisher—International Society for Technology in Education, 1787 Agate St., Eugene, OR 97403-1923; Editor—Dorothy M. Fitch, 3 Derby Road, Derry, NH 03038; Managing Editor—David Moursund; 1787 Agate St., Eugene, OR 97403-1923. 10. Owner: International Society for Technology in Education, 1787 Agate St., Eugene, OR 97403-1923. 11. Known Bondholders, Mortgages, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bondholders, Mortgages, and Other Securities: None. 12. The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes: Has Not Changed During Preceding 12 Months. 13. Publication Name: Logo Exchange. 14. Issue Date for Circulation Data Below: Fall 1996 (Volume 15, No. 1). 15. Extent and Nature of Circulation. Average No. Copies Each Issue During Preceding 12 Months. 15a. Total No. Copies (Net Press Run): 1,038 15b. Paid and/or Requested Circulation. (1) Sales Through Dealers and Carriers, Street Vendors and Counter Sales: none. (2) Paid or Requested Mail Subscriptions: 674. 15c. Total Paid and/or Requested Circulation (Sum of 15b(1) and 15b(2)): 674. 15d. Free Distribution by Mail (Samples, Complimentary, and Other Free): 73. 15e. Free Distribution Outside the mail (Carriers or Other Means): 44. 15f. Total Free Distribution (Sum of 15d and 15e), 117. 15g. Total Distribution (Sum of 15c and 15f): 791. 15h. Copies Not Distributed. (1) Office Use, Leftovers, Spoiled: 247. (2) Returns from News Agents: none. 15i. Total (Sum of 15g, 15h(1), and 15h(2)): 1038. Percent Paid and/or Requested Circulation: 82%. 15. Extent and Nature of Circulation. Actual No. Copies of Single Issue Published Nearest to Filing Date. 15a. Total No. Copies (Net Press Run): 1150. 15b. Paid and/or Requested Circulation. (1) Sales Through Dealers and Carriers, Street Vendors and Counter Sales: none. (2) Paid or Requested Mail Subscriptions: 640. 15c. Total Paid and/or Requested Circulation (Sum of 15b(1) and 15b(2)): 640. 15d. Free Distribution by Mail (Samples, Complimentary, and Other Free): 32. 15e. Free Distribution Outside the mail (Carriers or Other Means): 25. 15f. Total Free Distribution (Sum of 15d and 15e): 57. 15g. Total Distribution (Sum of 15c and 15f): 697. 15h. Copies Not Distributed. (1) Office Use, Leftovers, Spoiled: 453. (2) Returns from News Agents: none. 15i. Total (Sum of 15g, 15h(1), and 15h(2)): 1150. Percent Paid and/or Requested Circulation: 92%. 16. This Statement of Ownership will be printed in Volume 15, No. 2 issue. 17. I certify that the statements made by me above are correct and complete. Patti Van Ordstrand, Distribution/Circulation Manager, October 18, 1996.





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
 Phone: 800/336-5191 (US and Canada)
 Phone: 541/302-3777 (International)
 Fax: 541/302-3778
 World Wide Web: <http://isteonline.uoregon.edu>

LOGO®, LOGOWRITER®, AND MICROWORLDS® ARE REGISTERED TRADEMARKS OF LOGO COMPUTER SYSTEMS INC., HIGHGATE SPRINGS, VT, USA. TERRAPIN AND LOGO PLUS® ARE TRADEMARKS OF TERRAPIN SOFTWARE, INC., CAMBRIDGE, MA, USA.

ISTE PUBLICATIONS

The ISTE Logo in the Classroom Books include the following:

**Introduction to MicroWorlds—
 A Logo-Based Hypermedia
 Environment, Second Edition**

**Introduction to MicroWorlds 2.0—
 A Logo-Based Hypermedia
 Environment**

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

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

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

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

**Logo Musings—Ten Mathematical
 Encounters Using LogoWriter**

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

**Math Activities Using LogoWriter
 Series:**

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

**MicroWorlds—Hypermedia Project
 Development and Logo Scripting**

**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**

**Wacky Words for Traveling Tales—
 Logo Applications in Language Arts**

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>Introduction to MicroWorlds 2.0—A Logo-Based Hypermedia Environment</i>	25.15	27.95
• <i>Introduction to MicroWorlds—A Logo-Based Hypermedia Environment, 2nd Ed.</i>	25.15	27.95
• <i>Introduction to Programming in Logo Using Logo PLUS</i>	13.45	14.95
• <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>	13.45	14.95
• <i>LogoWriter for Educators—A Problem Solving Approach</i>	11.70	13.00
• <i>Logo Learning—Strategies for Assessing Content and Process</i>	6.25	6.95
• <i>Logo Musings—Ten Mathematical Encounters Using LogoWriter</i>	19.75	21.95
• <i>Math Activities Using LogoWriter (Specify Apple II series, Mac, or MS-DOS data disk)</i>		
<i>Patterns and Designs</i>	11.65	12.95
<i>Investigations</i>	8.95	9.95
<i>Numbers & Operations</i>	8.95	9.95
<i>High School Math</i>	11.65	12.95
<i>More Investigations</i>	8.95	9.95
<i>More Patterns and Designs</i>	8.95	9.95
<i>Probability and Statistics</i>	8.95	9.95
• <i>MicroWorlds—Hypermedia Project Development and Logo Scripting</i>	31.45	34.95
• <i>Turtle Power—Beginning Graphics With LogoWriter, Rev. Ed. (Specify Apple II series, Mac, or MS-DOS data disk)</i>	12.55	13.95
• <i>More Turtle Power—Text, Graphics, and Sound With LogoWriter (Specify Apple II series, Mac, or MS-DOS data disk)</i>	11.65	12.95
• <i>Turtle Time Designs—Working With Patterns in LogoWriter (Specify Apple II series, Mac, or MS-DOS data disk)</i>	8.95	9.95
• <i>Wacky Words for Traveling Tales—Logo Applications in Language Arts</i>	8.05	8.95
• <i>Way of the Ferret—Finding and Using Educational Resources on the Internet, 2nd Ed.</i>	26.95	29.95



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 LX2

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 \$4.75 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.

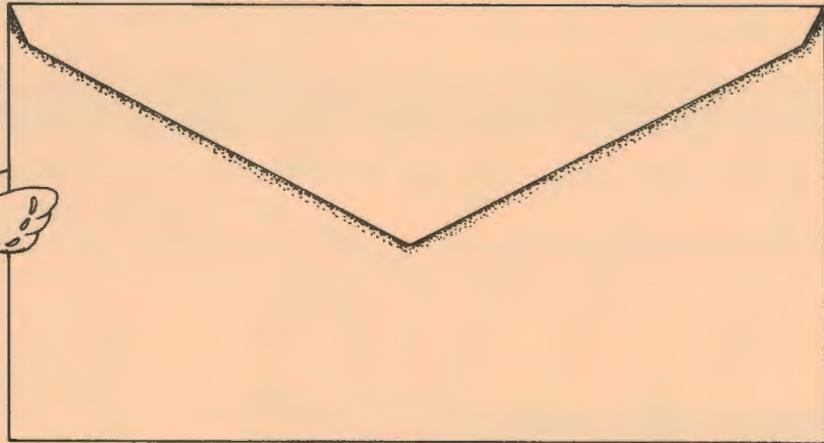
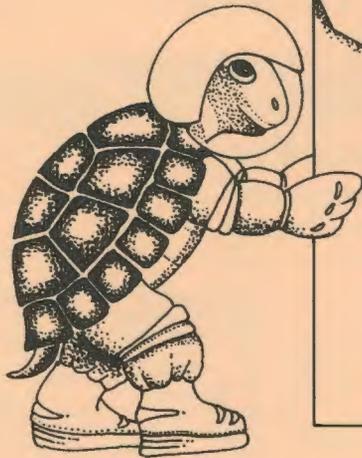
Send me a free ISTE catalog.

SUBTOTAL	-
Deduct 18% on ISTE-published titles if ordering quantities of 10 or more of the same title	-
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 \$4.75	+
TOTAL	=

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

ISTE PUBLICATIONS

A GREAT Book from ISTE



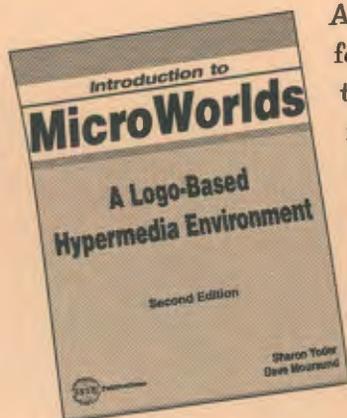
Push the envelope with MicroWorlds



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
Phone: 800/336-5191 (US and Canada)
Phone: 541/302-3777 (International)
Fax: 541/302-3778
World Wide Web: <http://isteonline.uoregon.edu>

ISTE BRINGS THE WORLD OF TECHNOLOGY CLOSER TO YOU.

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

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

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

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



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

WE'LL PUT YOU IN TOUCH WITH THE WORLD.