

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

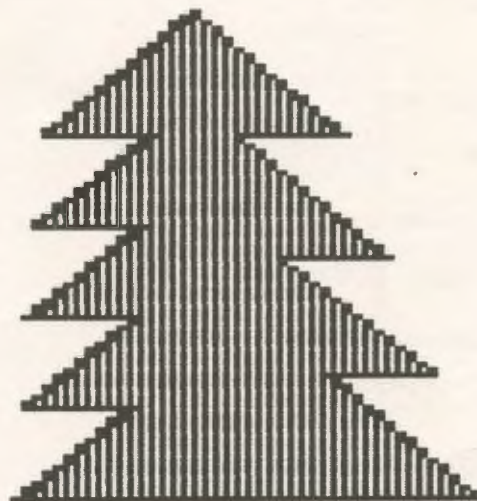


LOGO EXCHANGE

DECEMBER 1988

VOLUME 7 NUMBER 4

MERRY CHRISTMAS TO ALL



International Council for Computers in Education

✉ Long Distance Logo ✉

Educators - You don't have to go classes to earn graduate credit - let the classes come to you!

Introduction to Logo Using LogoWriter, a graduate level independent study course, allows you to learn at your own pace while corresponding with your instructor by mail.

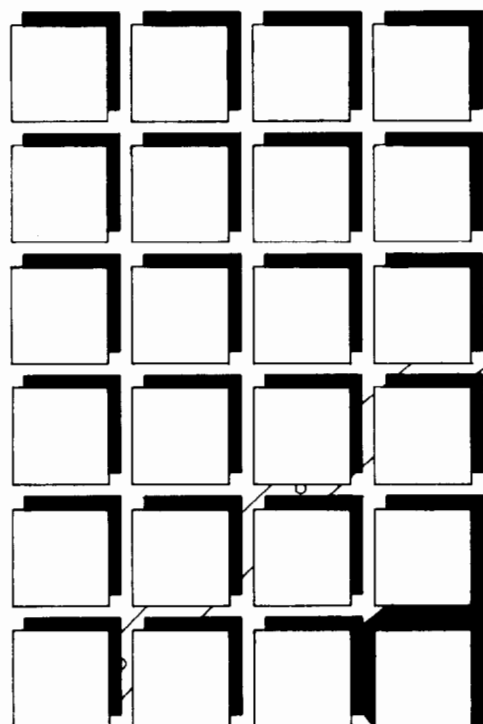
✎ **Work Individually or With a Group of Colleagues** - Take *Introduction to Logo Using LogoWriter* at home, or study with a group of colleagues at school. The course uses a combination of video tapes (ON LOGO) featuring MIT's Seymour Papert, printed materials, textbooks, and diskettes. You view the tapes, read and report on course materials, do projects, design LogoWriter lessons for your students and correspond with your instructor by mail.

📖 **Not Just Another Class** - Dr. Sharon Burrowes Yoder, editor of the *Logo Exchange* journal, designed *Introduction to Logo* to provide staff development and leadership training. The four quarter-hour course meets the standards of the College of Education at U. of Oregon, and carries graduate credit from the Oregon State System of Higher Education.

📼 **ON LOGO Video Tapes** - School Districts may acquire a license for the use of the ON LOGO package of 8 half-hour videotapes and 240 pages of supporting print for \$599.00. For a one-time fee of \$1295.00, the package may be obtained with both tape and print duplicating rights, enabling the district to build libraries at multiple sites.

💰 **Save \$\$** - If your district supports group training, you save money.

✎ **Tuition Information, Detailed Course Outlines, and Order Blanks** can be obtained from: LONG DISTANCE LEARNING, ICCE, U of O, 1787 Agate St., Eugene, OR 97403-9905. Phone 503/686-4414.



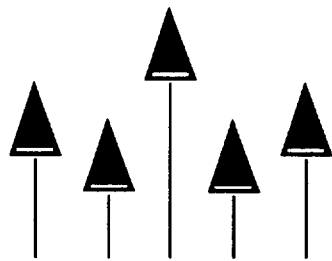
**A New
Book on Logo!**

Introduction to Programming in Logo Using LogoWriter

by
Sharon Burrowes Yoder

Designed for use either in teacher training or in an introductory secondary computer science class. Each section details new Logo primitives or programming concepts and ends with suggested open-ended activities for practice. Numerous appendices include key summaries for the Apple, IBM, and Commodore versions, a quick reference containing all LogoWriter primitives, and copies of the shapes pages. By Sharon Burrowes Yoder. \$14.95 plus \$2.50 U.S. shipping; \$3.25 Alaska, Hawaii, outside U.S. and P.O. Box shipping. All billed orders are charged \$2.50 for handling.

To order, call or write ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905. Ph: 503/686-4414.



LOGO EXCHANGE

Volume 7 Number 4

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

December 1988

Founding Editor

Tom Lough

Editor-In-Chief

Sharon Burrowes Yoder

International Editor

Dennis Harper

Senior Contributing Editor

Robb Muir

Field Editors

Anne McDougall

Richard Noss

Harry Pinxteren

Fatimata Seye Sylla

Jose Armando Valente

Hillel Weintraub

Contributing Editors

Eadie Adamson

Gina Bull

Glen Bull

Doug Clements

Sandy Dawson

Judi Harris

Gary Stager

Leslie Thyberg

Dan Watt

International Council for Computers in Education

Anita Best, Managing Editor

Vincent Elizabeth Barnett, Advertising

Dave Moursund, CEO

Keith Wetzel, SIG Coordinator

SIGLogo Board of Directors

Peter Rawitsch, President

Gary Stager, Vice-President

Ted Norton, Communications

Frank Matthews, Treasurer

Publisher

International Council for
Computers in Education

Logo Exchange is the journal of the International Council for Computers in Education Special Interest Group for Logo-using Educators (SIGLogo), published monthly September through May by ICCE, University of Oregon, 1787 Agate Street, Eugene, OR 97403-9905, USA.

POSTMASTER: Send address changes to Logo Exchange, UofO, 1787 Agate St., Eugene, OR 97403. Second-class postage paid at Eugene OR. USPS #000-554.

Contents

From the Editor — <i>Hidden Gems</i> Sharon Burrowes Yoder	2
Monthly Musings — <i>Transitions</i> Tom Lough	3
What is the International Computer Problem Solving Contest (ICPSC) Don Piele	4
Little Kids and Logo -- <i>Simple Gifts</i> Leslie Thyberg	5
Logo Ideas — <i>More Conversion!</i> Eadie Adamson	7
LogoLinux — <i>Subprocedure Anti-APPENDectomies</i> Judi Harris	10
Developing Creative Writing Projects with Text Files in Terrapin Logo Mel Levin	12
Logo Connections — Software Drivers: Electronic Glue Glen Bull and Gina Bull	15
MathWorlds -- Logo and Mathematics Education (LME4) at the International Congress of Mathematics Education: A report from Joel Hillel and Cecilia Hoyles Sandy Dawson, editor	18
Assessing Logo Learning in Classrooms -- III. Mathematics of Turtle Geometry -- Using 360 Degrees Dan Watt	20
Logo: Search and Research — Research on Teaching Problem Solving Douglas H. Clements	24
Global News Dennis Harper, editor	28
I Didn't Teach spelling Last Week Bill Craig	30

ICCE Membership (includes *The Computing Teacher*)

U.S.	Non-U.S.
28.50	31.50

SIGLogo Membership (includes *The Logo Exchange*)

	U.S.	Non-U.S.
ICCE Member Price	24.95	29.95
Non-ICCE Member Price	29.95	34.95

Send membership dues to ICCE. Add \$2.50 for processing if payment does not accompany your dues. VISA and Mastercard accepted. Add \$18.00 for airmail shipping.

© All papers and programs are copyrighted by ICCE unless otherwise specified. Permission for republication of programs or papers must first be gained from ICCE c/o Margaret McDonald Rasmussen.

Opinions expressed in this publication are those of the authors and do not necessarily reflect or represent the official policy of ICCE.

From the Editor

Hidden Gems

by Sharon Burrowes Yoder

Have you ever made a major move across a state, country, or the world? If so, then you have some sense of the small and large difficulties that you may encounter in such situations. Of course there are the not-so-little things like where to buy groceries and how to find a good car mechanic. But there are other things that mark you as a foreigner much more quickly than lack of knowledge about the local streets and stores.

Although I have now lived in Oregon more than 6 months, there are some fine points that still mark me as an easterner (yes, Ohioans are easterners out here.) As a newcomer, there were many things to learn...how to pronounce Oregon (Ory-gun), that I live in the Willamette valley (Will-AM'-et), and that OSU refers to Oregon State (not Ohio State). But there are subtle things that take more time. For example, there's a difference between pine and fir trees (this one still eludes me) and I need to remember that the sun sets over the ocean rather than rising there (I really do get confused.) But perhaps my most noticeable failure is my inability to recognize agates. Don't laugh! This is serious if you intend to be a real ocean-loving Oregonian.

Now, this midwesterner (easterner?) thought that agates were those lovely brown colored stones that were polished and used to make things like tie tacs and earrings. Wrong. It's much more complex than that. Further, a native has to be able to recognize them when strolling along the beach. Well, I tried. "This is an agate and this isn't" said Dave showing me two sandy rocks. "Oh?" I said, not understanding. He tried again and again "Oh?" I said each time a bit more confused. After many walks on the beach over a period of months, I was still confused. I was beginning to feel that I was the brunt of an enormous joke. Agates were just rocks that the collector (Dave in particular, Oregonians in general) thought were pretty. It was a game that the natives played to make outsiders think they were doing something esoteric instead of just enjoying the ocean.

Finally, in desperation, I looked in my trusty Columbia encyclopedia. Ah, agates are a "...variety of quartz banded in two or more different colors." No problem — I knew about quartz. Hmm, that's why agates polish so well. Perhaps I had it. But no, the next walk on the beach, and I was confused again. My bewildered state was compounded by a visit to some real agate collectors. Agates all over the place, big and small and of every imaginable color. What happened to my simple banded quartz? Our hosts handed me a booklet on agates and I quickly learned that agates come in many colors and types. Why didn't my encyclopedia give me all of this information? Where was a videodisk with hypertext so that I could have explored? Well, there

really was a lot to learn. I felt better...until the next walk on the beach. I failed again — those rocks on the beach just don't look like the polished beauties in that booklet that inhabit homes and shops out here. How in heavens name do these Oregonians know that *this* is an agate and *that* isn't? To me they are simply rocks on the beach — hidden gems to my untrained eyes.

I can't help but draw an analogy between those elusive agates and the students we find in our classrooms. Yes, I know, it's pretty obvious. Some of your students are certainly mere rocks. They just sit there. Others are gems and you polish them each day with your attention. As educators we often feel that it is clear which students will benefit from our TLC and which will not. We direct our energies towards these gems — sometimes consciously, sometimes unconsciously. We let the rocks take care of themselves.

But are we really as perceptive as we believe? I'm reminded of the student who sat quietly in the back of my Intro to Computers class for the entire semester without ever saying a word. He never asked a question. He never volunteered. He never caused a problem. In fact, it took me a long time to learn his name. He was a good student, turning in all of his work on time and quite well done. It wasn't until one of the final programming projects was submitted that I discovered that Mark was a hidden gem. He produced a magnificent program that illustrated "America the Beautiful" with text, music and multiple turtle animation. I still show that project when I do workshops and presentations (perhaps you have seen it) as an example of what a high school student can do in a single semester. How many other hidden gems did I miss in my assumption that I could tell which students would polish into candidates for more advanced programming? How many of the rocks in my classroom only needed a little more TLC to shine?

Perhaps this is a good time of year to take a second look at *your* classroom. Yes, I know things get a bit crazy as the holiday season approaches. Perhaps in the craziness you'll see a special sparkle in students that doesn't show other times of the year. Take note! Perhaps you have more gems than you thought. Perhaps you have a whole classroom of gems! How about a New Year's resolution to check out your entire class this month and then polish everyone in sight come January 1st?

And now, if I could just figure out which are fir trees and which are pine trees, and which of those silly rocks are agates. I guess I'll just have to go for lots of walks on the beach, won't I?

Sharon Burrowes Yoder SIGLogo/ICCE
1787 Agate Street, Eugene, Oregon 97403
CIS: 73007,1645 BitNet: YODER@OREGON

Monthly Musings

Transitions

by Tom Lough

Soon, something will happen to each of us which will change us all. We will go through a transition, the transition from 1988 to 1989. As the old year comes to a close, we will take down the calendar and its long awaited final December page, to replace it with a 12-page one for the new year. Transitions such as this provide opportunities for pausing and reflecting. I would like to take advantage of that opportunity to muse about transitions themselves.

I believe that there are important implications for teaching associated with the idea of transitions. Let me offer a few thoughts about them.

A transition involves a change from one stage or state into another. Transitions for young children can be momentous occasions. Some years ago, I was fortunate enough to observe the exact moment when my son experienced a transition which changed his life irrevocably: the occasion when he learned to crawl. Kyser, like most children, has always been the curious sort. As an infant, he was perpetually interested in what was "over there." When he learned that he could change his location by pushing up with his hands and then flopping forward, nothing near the floor in the house was safe.

One Sunday afternoon, he and I were down in the basement. While playing on the floor, he became interested in a colorful top on the other side of the room. So he flopped, flopped, flopped over to it. I happened to be observing him, and noticed that he seemed to do something extra on his last flop. Wanting to see him make that extra move once more, I placed the top on the opposite side of the room. Sure enough, he went for it. Flop, flop, flop, there! I would have sworn that he sort of crawled that last step just before getting to the top.

Being extremely curious by now, I placed the top on the opposite side of the room once more. No flopping now; he crawled swiftly over to the top and grabbed it gleefully! He had made the transition from flopping to crawling; his life (and mine!) was changed forever! He never flopped again.

Kyser's transition can be described as a permanent change affected by experience. This description contains the key elements of most generally accepted definitions of learning. But transitions have other properties which are educationally interesting.

Transitions can be one-way or two-way. By one-way, I mean that once the transition has occurred, there is no

turning back. Kyser's transition from flopping to crawling, for example, was one-way. I remember the one-way transition I experienced when Steve Tipps and Glen Bull introduced me to Logo in early 1982. I had been looking for a way to use the computer in physics teaching, but had not been encouraged by anything I had surveyed at that point. When I observed that the turtle used a self-referencing differential geometry, all my antennae went up. The clincher (I can remember it so well) was the occasion when Steve showed how to put SQUARE and TRIANGLE into a new procedure called HOUSE. Suddenly, something clicked. I could see for miles. I was never the same again, and will forever be grateful to Glen and Steve for this experience. (It was just after this experience, by the way, that the idea came to me to start a newsletter for teachers, so that information related to teaching with Logo could be exchanged and shared. If only I could figure out a good name for the publication ...)

A two-way transition permits a return to the previous state. An easy example to point out is the state of the Logo turtle's pen. You can always go from the PENDOWN state to the PENUP state and back again. Perhaps a less obvious example might be a student's transition from the concrete reasoning stage of development to the stage of formal operations. Research suggests that, under stressful conditions, students who have reached formal operations may regress to a temporary concrete reasoning state to solve particularly difficult problems. As teachers, do we experience something similar when faced with an unfamiliar or difficult situation?

Transitions can be transparent or opaque. By that, I mean that some transitions enable us to remember aspects of our former state. Like being able to see through transparent material, we can "see back" to our former state. For example, a person who has undergone the transition of marriage can remember what life was like (to some degree) before taking the nuptial vows. With other transitions, we seem to be unable to remember our former state; it is like attempting to look through an opaque object. In a classic experiment while working with Piaget, Seymour Papert demonstrated that, when children make the transition to a state in which they conserve, they are evidently unable to remember that they once said, "This container has more water than that one." When shown pre-conservation films taken earlier depicting them saying that very statement, children accused Papert of faking the footage with trick photography! The element of awareness is interesting here. I wonder what other opaque transitions we and our students undergo without being aware of them? What subtle impact might this have on our teaching and learning?

Monthly Musings -- continued

All of us have seen transitions with different speeds. Kyser's flopping to crawling transition was an example of a fast one. The "Aha! Now I've got it!" from a student is an indication of another. On the other hand, a student's complete transition from the concrete reasoning stage to the formal operations stage may take years. A student's transition from ignorance to understanding of a particular concept might take longer than a teacher expected. Are we sufficiently patient with students who are negotiating slow transitions? Are we patient with ourselves?

Sometimes, people seem to possess a sort of psychological inertia, a desire to keep things just as they are, particularly if things are generally pleasant or satisfying. In those cases, transitions can be times of stress, because of the accompanying changes. Was I sufficiently sensitive to my ninth grade science students, as they studied with me and at the same time endured the stressful transition associated with adolescence? And to the seniors in my high school physics classes, as they prepared for the transition to higher education and adulthood? I hope so. This semester, when introducing Logo to my community college physics students, I do my best to understand their concerns as they go through the transition of learning Logo and physics, as well as any other transitions they are negotiating in their personal or professional lives.

Transitions can be exciting! I am presently going through a transition which I would like to tell you about, briefly. Having served in the U.S. Army, taught high school, attained graduate degrees, and taught in a community college, I have been through some interesting and challenging transitions already. But the transition facing me this month is the most exciting of my life thus far! Effective immediately, I will be taking a position as the Curriculum Coordinator of the Educational Division of LEGO Systems, Inc. My responsibilities will include developing instructional materials for LEGO product use in science, mathematics, and other curriculum areas for grades K - 12, helping to train teachers and teacher trainers to use these materials, and supervising educational research on the results of these materials and products.

I do not intend for this transition to be opaque. The needs of real teachers in real classrooms are of great concern to me. I will continue to meet these needs in as many ways as I can, and look forward to opportunities to do so.

May each of you have a spectacular 1989, as you make the transition into the New Year. See you in January!

As ever, FD 100!

Tom Lough, LX Founding Editor
Box 394, Simsbury, CT 06070

ICPSC

What is the International Computer Problem Solving Contest (ICPSC)?

To some schools it is a chance to challenge their top computer and mathematics students in an annual computer problem solving event that compares their solutions to the best in the world. To others it is a challenging set of problems to be used as enrichment material for computer programming classes. However you choose to use the ICPSC, it can be a valuable resource for any programming teacher.

What is unique about the ICPSC?

The ICPSC combines the art of problem solving with the skill of computer programming. Our contest challenges teams (from one to three students each) to create short, original solutions to a set of five problems within a two-hour period. The problems fall into five categories: (1) computation, (2) simulation, (3) graphic patterns, (4) words, and (5) mind benders.

Can an individual teacher in a school become a contest director for his/her school?

Yes! We want to challenge and enrich your students so you don't have to make it a major district-wide event if that would stop you from doing it. Of course the more students in your local area who can get involved the better. But, if you only have one student in your whole school who wants to compete, that is all you need.

Can Logo be used?

Our original contest problems were not appropriate for Logo students. So we added a new Logo contest and last year expanded this contest to include all three age groups.

What are all the age groups?

Elementary Division (Grades 4-6), Junior Division (Grades 7-9) and Senior Division (Grades 10-12) are now offered in either the Open contest (all languages) or the Logo contest.

When is the contest held?

The 9th Annual ICPSC is scheduled for Saturday, April 29, 1989. Friday, April 28 and Monday, May 1, are alternate dates that can be used if Saturday is impossible. This year's set of contest problems will be mailed to the contest director on April 1, 1989. Sets of previous problems and solutions for students to practice on are always available.

How can I get more information about the ICPSC?

The Computing Teacher annually publishes articles about the contest in the October, November, December issues. Also the *Logo Exchange* is now publishing stories on the Logo contest. You may receive a free copy of *Compute It!*, the official newsletter of the ICPSC which is published in November each year, by writing to:

Dr. Donald T. Piele, ICPSC
P.O. Box 085664, Racine, WI 53408
Ph 414-634-0868

Little Kids and Logo

Simple Gifts by Leslie Thyberg

Deck the walls with a children's keyboard
Fa-la-la-la-la-la-la-la-la
Tis' the season for learning FORWARD
Fa-la-la-la-la-la-la-la-la
Don' we now our Turtle apparel
Fa-la-la-la-la-la-la-la-la
As we sing of learning pitfalls and perils
Fa-la-la-la-la-la-la-la-la

Logo Lesson

A Logo learning environment is no different than any truly stimulating classroom. Simply put, it is a place that allows for children to explore, discover, and build their own intellectual structures with materials drawn from their surroundings. So too, a Logo lesson runs much the same as any other lesson plan might.

1. A new concept is introduced.
2. When appropriate, review and reteaching are done to tie the concept in with what is already known.
3. Examples are practiced together as a group.
4. Guided practice occurs.
5. Free exploration takes place.

All lessons should proceed from the familiar to the unfamiliar; the known to the unknown; the concrete to the abstract. The individual strengths, needs and differences of each child should always be kept in mind. For example, perhaps you are working on some keyboard familiarity with your students. The following is a sample Logo lesson using this format:

The new concept: Orientation to the keyboard

Specific objectives:

1. The students will be able to locate the following keys on the keyboard: F, D, B, K, L, T, R, all numbers, Return/Enter, Esc (or Delete), and the Space bar.
2. The students will be able to differentiate between the letter O and the number 0, and between the letter L and the number 1.

Group practice:

To orient the children to the keyboard, have a discussion about its similarity to a typewriter. Use an actual typewriter for demonstration and comparison. Introduce the keyboard using a poster-size replica or an overhead projection. Show all the letters and the numbers. If you're more secure as a ditto user, pass out dittoed copies of the keyboard to all the students. Then, as you have various children come to the poster or overhead display and press the "keys" to carry out specified commands, all the students can role play this in

their seats with their individual keyboard replicas. Then, gather the group around a computer. Select one volunteer at a time to sit at the computer and have the group dictate which key to use, until many of the children have experimented with a variety of keys.

Guided practice:

Remember the game Twister? You can use an actual Twister mat, a checkered plastic table cloth, or you can label squares on the linoleum floor to get your students keyed in and kinesthetically involved. Too chaotic or crowded for your teaching style? I employed the zealous energies of some older students to make me some small Bingo-like game boards for a game we called "ERTLE." On the boards, we placed abbreviations for the most frequently used primitives. Of course, just like Bingo, ERTLE has HOME as its center square. It was a highly motivational, very inexpensive way of giving the children lots of exposure and review so that when their altogether too limited computer time rolled around they could utilize it to the fullest.

Prefer individual or paired peer practice? Patricia Mocello, a second grade teacher in the Pittsburgh Public School system, cut out tagboard ladybugs. She called the game "BUG and DEBUG." She made 18 cards with Ladybugs on one side and computer bugs and debugs (e.g., missing spaces, misspellings, etc.) on the other side. The objective was to have the student match the bug and the debug. Much like concentration, the students mix up cards and spread them out face down. They are to turn over two cards and try to match the bug with its debug. If they match, they take the pair. If they don't match, turn them back over.

Several exemplary teachers from past workshops that I have conducted put together some wonderful learning centers for their students. One made a huge, felt monitor screen with the grid lines filled in. She cut out colorful arrows and turtles for her young students to freely play with and explore. There were also specific task cards with open ended exploratory activities for the students to engage in as well. Another creative teacher made a velcro-keyboard with a similar set of activity cards. Being budget minded and totally inept with a sewing machine, I chose to use the burger containers (e.g., for a Big Mac) and invert them on the wall as a combination bulletin board display, visual aid, and learning center. The carton bottoms were covered with construction paper squares and labelled according to the kind of machines in the classroom.

Other Off Computer Activities

Most classrooms do not have the luxury of one computer for every student. More often than not, the situation is one computer and thirty students or an infrequent visit to the

Little Kids and Logo -- continued

computer lab. To maximize hands on time, the use of off computer or precomputer activities is strongly suggested. These activities help give the students direction and purpose.

Myrtle says:

No Logo classroom is complete without a turtle mascot. Furry Folk ® make wonderful puppets. Their turtle does a great HIDE TURTLE retreat into its shell. Trying to help your students differentiate between left and right? Use a turtle mascot (puppet, stuffed toy, or engineer's cap on your head with turtle appendages attached!) Introduce Myrtle the mascot and tell the children that the turtle will be giving them directions in the Myrtle Says Game. Play the game just like Simon Says. Give each child a set of LT and RT stickers or tie colored yarn around their right wrists. Show the children on the board or monitor screen the abbreviations for LEFT and RIGHT and FORWARD and BACK. Play Myrtle Says giving oral directions or by using flashcards. Let several children take turns giving the directions.

Playing Turtle

The advantage of your taking the role of the turtle is that you can model error messages for your students. Take the game a step further and include your keyboard center. Have students take turns typing the commands on the keyboard. Have them verbally relate what they are typing "Forward Space bar Fifty", "Right turn Ninety Space bar," and so forth. This way, if they neglect to leave a space, you can model with the response I DON'T KNOW HOW TO... It helps them learn to debug their own problems, and to view errors as helpful and playful prompts from the turtle.

Treasure Hunts

Since repetition and experience are two trademarks of early learning, another way to help your students learn to give and/or receive commands is to send them on a Turtle Treasure Hunt. Much like Button Button, Who's Got the Button, you can choose a child to leave the room while an object is being hidden. Then select volunteers from the group to give commands to the child when the child comes back into the room. Let them give commands until the object is found. After each round, discuss the efficiency and correctness of commands given and carried out.

Take the treasure hunt a step further. Have a 5 minute hunting spree for all the square (round, rectangular, triangular, etc.) shapes that they can find in the classroom. (For homework they can go on a shape hunt. It's amazing how many bathroom floors suddenly become objects of interest when Logo is being learned!) Have a shape of the day (or of the week — whatever time frame your students require.) In the art center have lots of construction paper cutouts of that

shape. If its squares, have them create a picture using different sizes and colors of squares. The ultimate challenge comes when they have their computer time. Can they make that same picture by drawing it with the turtle? Maybe you've covered all the shapes by now. Have the students create geometric stick figures. In the writing center have them write stories (or take dictation from them) about their creations. Can they make a Logo stick person using the shapes they used in the art center? This is a great springboard into subprocedures. But I'm getting ahead of myself. My On Computer Gifts and Utilities are coming your way in a future column!

Dr. Leslie F. Thyberg
5637 Rippey Street
Pittsburgh, PA 15206
or
c/o Chatham College
Woodland Road
Pittsburgh, PA 15208
AppleLink ALS 038

The ECCO Logo Project: *Materials for Classroom Teachers and Teacher Trainers*



*Edited by
Theodore C. Burrowes &
Sharon K. Burrowes*

This book presents Logo activities for use in grades four through eight. Use it for teacher training or take it directly into the classroom to enhance the teaching of language arts, social studies, science and math.

The ECCO Logo Project Includes: Student worksheets, teacher-training materials, Logo II version of all materials, and Apple Logo version of teacher materials.

The Educational Computer Consortium of Ohio (ECCO) developed these materials over the course of an academic year, in conjunction with its extensive series of Logo workshops.

To Order, write or call:

ICCE, University of Oregon, 1787 Agate St, Eugene, OR
97403-9905, or phone 503/686-4414.

An International Council for Computers in Education Publication

Logo Ideas

More Conversions!

by Eadie Adamson

I was so delighted to read Judi Harris' September LX column on "Secular Conversions" just as I was procrastinating over finishing this column (yes, we **all** really do write that far ahead!). This month's "Logo Idea" is about yet another conversion or, better yet, a *link* between versions of Logo and the pictures that they produce.

Increasingly you will find that newer programs create files that can be easily accessed by other programs. This has become very important to me. I write with AppleWriter, save LogoWriter 2.0 programs as text and import them into my AppleWriter document, and then occasionally load the assembled file into MultiScribe on my GS and "prettify" it before printing. I also send ASCII files created by AppleWriter to Sharon Yoder which she transfers into Microsoft Word on the Macintosh for editing. She then loads the Word files into PageMaker to so she can typeset them for the *Logo Exchange*!

While this interchange of files has been possible with LogoWriter text files and other word processors, as well as with LogoWriter and some graphic programs, a similar connection between LogoWriter and Terrapin Logo pictures has not been so easily achieved until now.

Now Terrapin has a new ProDOS Logo called Logo PLUS, which I first played with at the Logo Institute this summer. What interested me about Logo PLUS was that a picture saved in LogoWriter can be loaded into Logo PLUS, changed, then saved again (this time in Logo PLUS), and finally be loaded back into LogoWriter to be animated with multiple turtles. Why, you might ask, would you want to do this? Read on....

New Features in Logo PLUS

Logo PLUS has a variety of text styles, something not available in LogoWriter. The text is displayed as graphics, comparable to LogoWriter's LABEL mode. Since the operating systems of Logo PLUS and LogoWriter are the same (Apple ProDOS), it is possible to access special features of either version by saving a picture and then loading it into the other version of Logo for additions. For a LogoWriter user, this means that the various text styles of Logo PLUS — boldface and italic, a different style of plain text, as well as options to underline or use color — are now available for labeling pictures, creating signs and the like. The final version of Logo PLUS has a font editor as well. Think of the wonderful maps and city scenes you can make now!

The Switch

The interchange process between LogoWriter and Logo PLUS is much easier than what one must do to import or export between Logo and Print Shop (see Judi Harris, *September 1988 LX*, pp. 15 - 17) — and every bit as much fun!

Here's all you need to do to get a LogoWriter picture into Logo PLUS:

With LogoWriter loaded and your picture on the screen, type:

```
SAVEPIC "name.PICT
```

The .PICT is essential for Logo PLUS, which identifies picture files by the .PICT at the end of the file name. Without .PICT Logo PLUS cannot identify your saved picture as a picture file.

Press RETURN.

Your picture is now saved as a binary file on your disk.

Load Logo PLUS.

Type:

```
READPICT "name.of.picture
```

Omit the .PICT here; it serves only as an identifier for pictures in Logo PLUS.

Press RETURN.

Behold! A LogoWriter picture on a Logo PLUS screen. Now you can add all the fancy text you want using Logo PLUS's special commands for adding text.

When you finish adding the features you need from Logo PLUS, it's time to save the picture again:

```
SAVEPICT "name.of.picture
```

You don't add .PICT this time; Logo PLUS does this automatically. Unlike LogoWriter you can use the same picture name in Logo PLUS. Logo PLUS will overwrite the old picture with the new version.

Now reload LogoWriter. Type:

```
LOADPIC "picture.PICT
```

This time you need the .PICT since LogoWriter takes its file names literally. If you catalog the disk on which you saved the picture — use SHOW or PRINT FILELIST if you are in LogoWriter — you will see the picture listed with its name and .PICT appended. By the way, you can save your pictures

Logo Ideas -- continued

on any disk formatted with ProDOS. You don't have to save your pictures on a LogoWriter or Logo PLUS disk!

Go ahead and use your new picture!

Notice that Terrapin READs and LogoWriter LOADs a picture. Just a little confusing, but you can manage. Terrapin also uses PICT and LogoWriter uses PIC: SAVEPICT or READPICT for Logo PLUS, LOADPIC or SAVEPIC for LogoWriter. If you associate "T" with Terrapin you'll get it right every time.

Two Printing Sizes in Logo PLUS

Another interesting possibility is presented by Logo PLUS's two screen printing sizes: a standard picture size and a reduced small screen version. When I tried Logo PLUS I thought immediately of how useful this interchange could be to a confirmed LogoWriter user (yes, I still am — but that's another issue.) Think of the illustrated stories students have written. Now they can be printed in small size, text can be printed as well, and neat little books can be put together. True, you could also reduce your pictures on a xerox machine, but this is much more fun!

A Different Kind of Shape and Screen Editor

The shape editor for Logo PLUS allows you to make any size shape you want, even full screen size. Rather than filling in a rectangle with squares, Logo PLUS works more like Pixit than anything else I can think of. (Macintosh users would talk about editing "fat bits.") Much more finely detailed shapes can be created than with LogoWriter. You see the shape you are designing, as you do in the new IBM Logo-Writer.

While there is still only a single turtle, the shapes you design in Logo PLUS will rotate unless you use the LOCK-HEADING primitive which prevents the shape from turning. Using the VFLIP and HFLIP primitives, the entire screen can be flipped in the immediate mode or under program control (in procedures.) Shapes can also be flipped horizontally or vertically in the shape editor, making it easy to copy and reverse a shape at the same time.

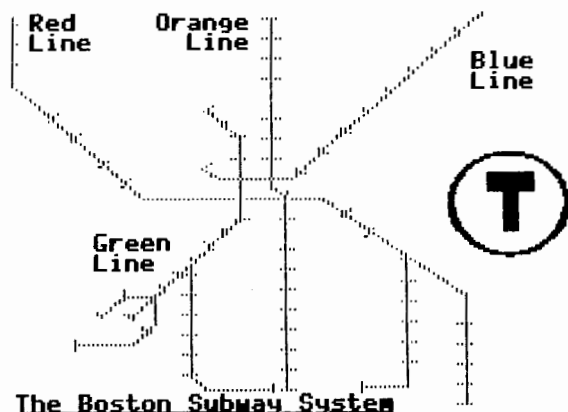
Shapes can be stamped with two different commands: STAMP and STAMPXY, which takes x- and y-coordinates as inputs. The Logo PLUS shape editor is not fully developed in the pre-release version I have been playing with, but it is to be a screen editor as well. Imagine saving a student's work and then giving him the capability of fine tuning a messed-up picture. In addition you can take whatever is already on the screen and turn that into a shape as well.

If a Terrapin user wants to use stamped LogoWriter shapes in a picture, it is perfectly possible to do so by saving the Logo PLUS picture, loading it into LogoWriter, creating

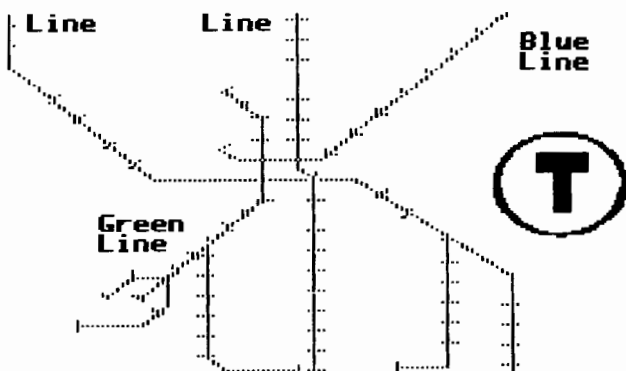
and stamping appropriate shapes, and then saving the picture (with a new name) to reload into Logo PLUS. A LogoWriter picture can also acquire added shapes from Logo PLUS if you can be satisfied with stamped shapes. Unfortunately, the two shape systems are not compatible. You cannot create shapes in Logo PLUS and animate them in LogoWriter. The reverse is possible, however. Save a page with LogoWriter stamped shapes, read the page into Logo PLUS, use special Logo PLUS editing commands that allow you to edit the whole screen and turn a part of the LogoWriter stamped shape into a Logo PLUS shape, and there you are!

Some Examples

As I thought about all these possibilities, I decided to try some things for myself. Here, for example, is a screen print of a Logo Plus map of the Boston T (Boston's subway system), which Dorothy Fitch of Terrapin, Inc. created as a starter for a group project at the Logo Institute this summer. I have added text to show the bottom of the Logo PLUS screen:



Here is the same map after loading into LogoWriter:



Notice that the top and bottom lines of the text are lost in the transfer to LogoWriter. That would need to be taken into account when working back and forth between versions.

Here is a sample of a LogoWriter page which was saved and then loaded into Logo PLUS. The italicized lines show you what can be added, in Logo PLUS, to a full-screen LogoWriter picture.

This line is added from Logo PLUS!!
 ➤ An animated plane flight
 with three turtles.
 TAYLOR GRANT
 Grade 4 ➤



And this line is added from Logo PLUS!!

And here is that same picture saved from Logo PLUS and loaded into LogoWriter. It looks just as it did originally; the added Logo PLUS lines do not show.

➤ An animated plane flight
 with three turtles.
 TAYLOR GRANT
 Grade 4 ➤



THIS IS HOW THE PICTURE SAVED
 IN LOGO PLUS LOOKS WHEN LOADED
 INTO LOGOWRITER. THE TOP AND
 BOTTOM LINES OF TEXT ARE GONE!



Think of the Possibilities!

I have a number of students whom I know will be fascinated by the possibility of interaction between LogoWriter and Logo Plus. Here are some potential projects which might interest them — or you:

A STORYBOOK

Make a picture story in LogoWriter, design a beginning reader or an alphabet book — use graphics only.

Save the pictures.

Load Logo PLUS and, one at a time, load the LogoWriter pictures and work on them.

Use Logo PLUS to add simple text: a single line at top and bottom if your LogoWriter picture has filled the entire screen. (You could leave additional space at top or bottom for addition of more text from Logo PLUS.)

Print the pictures in the smaller size to make a book.

AN ANIMATED STREET SCENE OR MAP

Take your favorite LogoWriter street scene or map.

Save the picture.

Load the picture into Logo PLUS.

Add fancy text labels to your picture.

Save the picture.

Load the picture into LogoWriter and add your animation.

(Or, if you are a Logo PLUS partisan, make wonderful new shapes with Logo PLUS and animate one of them.)

HIGHWAY SIGNS

Design a billboard.

Create shapes to stamp on the billboard.

Decide on your text.

Use the variety of text styles available in Logo PLUS and the LogoWriter text to create an interesting sign.

Make the LogoWriter text flash!

A MAP-MAKING PROJECT

Start a class map-making project.

Design shapes to indicate topography.

Leave some labeling of the maps and perhaps some of the topography symbols for Logo PLUS.

Save the LogoWriter picture.

Load into Logo PLUS.

Add the rest of the text for the map, in appropriate colors and type styles.

Save the picture again.

Load back into LogoWriter.

Add some flashing signs at key points of interest.

Program an animated "tour" of your map:

An interactive program might allow you to "drive" through the map, a nice project for both LogoWriter and Logo PLUS users.

An automated tour might give you tour guide information in the LogoWriter command center. Use TYPE to place text in the command center. Use TYPE CHAR 13 to add a carriage return at the end of the text.

So, if you're a LogoWriter user, it will be great fun to have a few copies of Logo PLUS around. If you are a Terrapin user, switch to Logo PLUS and get a few copies of LogoWriter to try this whole process in reverse!

Now, if only we could have different text sizes too
 ... or multi-colored shapes

... or a version of Logo that make use of the sound
 and color capabilities of the GS or

Is anyone out there listening?

Note of thanks: I want to thank Dorothy Fitch of Terrapin, Inc. for allowing me to use a test version of Logo PLUS to write this article. Her help on technical questions was invaluable.

Eadie Adamson, Allen Stevenson School
 132 East 78th Street, New York, New York 10021

Logo LinX

Subprocedure Anti-APPENDectomies

by Judi Harris

I once heard of a chemistry professor that immediately erased with his left hand what he wrote on the chalkboard with his right. The idea fascinated me, although I was thankful that I wasn't in his class. Logo work in the immediate mode is a bit like the professor's quick-to-disappear chemical equations. The result may be admired for its precision, beauty, or conceptual power, but the process by which it was formed has disappeared into chalkdust or computer ether.

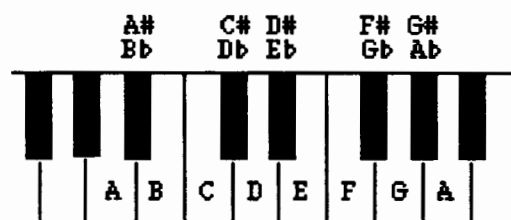
Logo novices may be hesitant to do what you are about to suggest: write procedures that can be stored in memory and invoked with a single command. The leap from immediate mode doodling to preplanned, hierarchically structured procedure writing can be a large one.

It is possible to bridge these two experiences with a simple Logo tool that I call APPEND. When added to all student- or teacher-defined procedures, APPEND constructs a sequential record of subprocedures as they are used in the immediate mode. This list can then be defined as an operable superprocedure with another tool procedure called TITLE.

A Melodious Microworld

Consider, for example, a project that a Pennsylvania fifth-grader conceived and coded. The idea of Logo music intrigued him, especially because he was taking clarinet lessons, and reasoned that if he wrote Logo procedures to play practice pieces for him, then he would be sure to play the pieces correctly on the clarinet.

He proceeded to make eighteen subprocedures, one for each note in the chromatic scale beginning with A below middle C. All shared a common local variable, TIME, used to express the duration of the notes.



```
TO A :TIME
TONE 440 :TIME
END
```

```
TO B :TIME
TONE 494 :TIME
END
```

```
TO C :TIME
TONE 523 :TIME
END
```

```
TO D :TIME
TONE 587 :TIME
END
```

```
TO E :TIME
TONE 659 :TIME
END
```

```
TO F :TIME
TONE 698 :TIME
END
```

```
TO G :TIME
TONE 784 :TIME
END
```

```
TO A2 :TIME
TONE 880 :TIME
END
```

```
TO A# :TIME
TONE 466 :TIME
END
```

```
TO Bb :TIME
TONE 466 :TIME
END
```

```
TO C# :TIME
TONE 554 :TIME
END
```

```
TO Db :TIME
TONE 554 :TIME
END
```

```
TO D# :TIME
TONE 622 :TIME
END
```

```
TO Eb :TIME
TONE 622 :TIME
END
```

```
TO F# :TIME
TONE 740 :TIME
END
```

```
TO Gb :TIME
TONE 740 :TIME
END
```

```
TO G# :TIME
TONE 830 :TIME
END
```

```
TO Ab :TIME
TONE 830 :TIME
END
```

(Note: LCSI Logo II and Apple Logo II use the command TOOT instead of TONE, but the frequency numbers are the same.)

The student could then easily combine the subprocedures into superprocedures that played practice songs.

```
TO CHROMATIC.SCALE
```

```
A 24
A# 24
B 24
C 24
C# 24
D 24
D# 24
```

```

E 24
F 24
F# 24
G 24
G# 24
A2 24
END

TO TWINKLE
A 12 A 12 G 12 G 12 A2 12 A2 12 G 24
F 12 F 12 E 12 E 12 D 12 D 12 C 24
G 12 G 12 F 12 F 12 E 12 E 12 D 24
G 12 G 12 F 12 F 12 E 12 E 12 D 24
A 12 A 12 G 12 G 12 A2 12 A2 12 G 24
F 12 F 12 E 12 E 12 D 12 D 12 C 24
END

```

But when the student's younger brother stopped by after school, it was clear that he was not ready to write tune procedures like the one above. Instead, he began playing with the note subprocedures in an immediate mode format. His frustration was, of course, that a particular sequence had to be recorded on paper or memorized to be repeated.

Subprocedure APPENDages

An APPEND tool at the end of each of the note procedures would have automatically built a sequenced list of the notes that the younger brother used in his melody experiments.

```

TO APPEND :COMMAND
MAKE "CREATION LPUT
:COMMAND :CREATION
END

TO A :TIME      TO A# :TIME
TONE 440 :TIME  TONE 466 :TIME
APPEND "A       APPEND "A#
APPEND :TIME    APPEND :TIME
END             END

TO Bb :TIME
TONE 466 :TIME
APPEND "Bb
APPEND :TIME
END

```

In the APPEND tool, CREATION is a global variable name that refers to the record of commands, in list format, that forms sequentially in the computer's memory as the music program user builds a melody. CREATION must be set to [] (an empty list) before the musical record of user keystrokes can be started.

```
MAKE "CREATION [ ]
```

Another simple tool procedure, TITLE, takes the current value of CREATION and defines it as a procedure with a title that the user chooses. It then restarts "CREATION, so that the next tone sequence may be recorded separately.

```

TO TITLE
CLEARTEXT
TYPE [TITLE?]
DEFINE READWORD LIST [ ] :CREATION
MAKE "CREATION [ ]
CLEARTEXT
END

```

You're probably thinking, "but what happens if the user types a note, then wants to change it?" As a devoted facilitator, I'd reply, "that's a fascinating programming challenge! How might you go about coding a tool to allow users to alter the string of commands in the computer's memory?"

INSTANT History

One of the most popular examples of action-recording microworlds is INSTANT, a single-keystroke adaptation of Logo graphics commands for new or physically challenged users. Most versions of INSTANT allow users to enter turtle commands with single letters (i.e., F(oward), B(ack), R(ight), and L(eft)). This limits the number of commands that can be created to the number of characters on the keyboard, and further constricts inputs to these commands. Most single-keystroke INSTANTs, for example, make the turtle move 10 steps forward when the user presses an F, and turn the turtle 10 or 90 degrees to the right or left when users type R or L. Angle estimation and length approximation become much less conscious activities when users don't have to type numbers as inputs to movement or heading commands.

If users are physically able to press the Enter or Return key after typing a command, then command options can be greatly increased. Microworld function names can be one or more letters long, can accept variable input, and if APPEND is used, an automatic record of user actions can still be maintained. Moreover, new commands (in the form of short procedures) can be added to the microworlds by either students or teachers without having to change existing code.

APPENDix of Ideas

Although the appropriateness of using INSTANT-like microworlds has been debated (see the articles listed in

Logo LinX -- continued

the References at the end of this column), user-based, self-recording microworlds need not be limited to INSTANT-like programs. Microworlds are best used when they are integrated into subject-centered study. Here is a list of possibilities. Hopefully, these ideas will spark others that can be infused into curricular study in your classroom.

- rebus maker (language arts)
- hieroglyphic typewriter (social studies)
- chemical element diagram-maker (science)
- pattern block microworld (mathematics)
- animation tool (SETSHAPE sequences; art)
- Lego Logo obstacle course path for a car (science)
- sound effect sequences (music)
- braille, Morse code, or sign language translator (language arts)
- choreography tool (music/dance)

If you would like to see examples of how procedures like APPEND and TITLE can be used in subject-centered microworld construction, I would be glad to send you a copy of a pattern blocks/parquetry blocks microworld that I constructed for elementary school use and/or an upgraded version of INSTANT, written for kindergarten students, that I call DRAW. Send a blank, double-sided diskette and a disk mailer with sufficient return postage to the address listed at the end of this article. I would also love to hear from you if you create or revise microworlds using procedures like APPEND and TITLE.

References

- Bull, G. L. & Cochran, P. S. (1986). Instant tools. *National Logo Exchange*, 4(4), 8-11.
- Lough, T. M. (1983). Slow turtle moves clearly. *National Logo Exchange*, 2(1), 7-8.
- Tipps, S. (1983). The issue of instant. *National Logo Exchange*, 2(4), 6-10.
- Watt, D. & Watt, M. (1986). Starting Logo with young children? Slow down the turtle! *Logo Exchange*, 5(1), 3-5.

Judi Harris taught students in Philadelphia-area elementary through graduate schools to use computer in teaching and learning for six years. She now does similar work at the University of Virginia, where she is completing her doctoral work in Instructional Technology. She can be reached at

Judi Harris
621F Madison Avenue
Charlottesville, VA 22903
CIS: 75116,1207 BitNet: jbh7c@Virginia

Creative Writing

Developing Creative Writing Projects with Text Files in Terrapin Logo

by
Mel Levin

Using Logo as a Text Editor

The Logo language is designed to read and save files that contain Logo procedures. By modifying the system, one can use Logo to read and save text files (and thus be able to use the Logo editor purely as a text editor).

The advantage to using the Logo editor as a text editor rather than as a procedure editor is two-fold. First, writing text in this manner uses very little memory in the workspace. This is helpful if you are writing a program which contains a lot of text material (instructions, explanations, written passages, etc.) If you wrote the text as procedures, you would be using primitives which take up lots of memory in the Logo workspace. If your program is large, there is danger that you might encounter the dreaded "no storage left" message. Using the Logo editor as a text editor helps eliminate the problem.

Secondly, you are using the Logo editor as a quasi-word processor, which means that you write text as you would in English and not as Logo code. This means there is no code syntax problem (no hassles with PRINT, SENTENCE, WORD, [,], etc.). You also have the freedom to format the text as you would with a word processor. This is not to imply that the Logo editor has terrific word processing capabilities. It is, in fact, quite limited as a word processor. The point, however, is that text can be formatted relatively easily within the restrictions of the Logo environment.

Normally, to enter the editor you type TO followed by the name of a procedure to be created or edited. To work with just text in Logo, it is necessary for the edit buffer in memory to be empty. Entering the edit buffer is achieved by typing TO followed by Return. At this point text is written and edited the same way as a procedure is written and edited. You use the same cursor keys to move around the editor. In order to save the text, it is necessary to exit the editor. Instead of typing Control-C, which defines a procedure, Control-G is used. This exits the editor without modifying what is in the edit buffer. Thus, the text will be printed as written, and not reformatted as would a procedure. Next, you must save the buffer as a file on the disk. At this point, you must not edit any procedures or enter the graphics mode before you save the buffer or it will be lost. While you can save the text with any name, you cannot save in the usual way.

Fortunately, there exists a group of procedures that allow you to save text easily. The procedures, called TEXTEDIT, from Terrapin Logo are on the Utilities I disk that comes with the Logo language.

Here's how to use the procedures to write and save text files:

1. First read the TEXTEDIT files into computer memory.
2. Type TO followed by Return. This allows you to use the Logo editor as a text editor.
3. Type text as you would in a word processor.
4. Exit the editor with Control-G.
5. Type SAVETEXT "FILENAME immediately.

Armed with the TEXTEDIT procedures and the knowledge of how to use them, let's explore the concept of developing a creative writing project.

Writing An Interactive Text Adventure

Logo can be incorporated across many curriculum areas. In the area of Language Arts, for example, Logo can provide an interesting forum for developing creative writing. One way the Logo editor and the accompanying procedures can be utilized is to write interactive text stories. Students can write the stories in Language Arts, and then use Logo code to produce interactive text stories on disk.

The concept of interactive creative writing is to permit the reader (user) to be an active participant in the story. As s/he reads, s/he makes decisions that affect the story line.

The first step in this project is for students to prepare their stories off the computer. I have had teams of students write the story together as a group. They decide on the story plot and write each of the sections. It is important that students follow a prescribed writing process. The Language Arts teacher can be a collaborative support person here providing the various techniques and strategies. We then put the pieces together to complete the story, making sure each branch goes where it should (use tree diagrams, etc. to graph the story board).

Plan each screen of text as a separate page in a book. Give each page a name, title, or number to identify it. At the end of each page, direct the user to make a decision. This will send the user to a different page of the story, thus creating many different story plots.

When you have students develop their interactive writing, it is important that you have them chart the course of the story. This can be achieved by introducing the concept of

tree diagrams, flow charts, etc.. It will make things easier when you begin to code and certainly aid in the writing itself.

After the story flows correctly, we then begin to code it in Logo and save it on disk. Theoretically, you can develop a complete library of stories on disk for everyone to share and enjoy!

Lastly, we need to write a number of procedures to act as a template, in order to control the correct flow of the text files. Unless the students have a great deal of experience in Logo, they will not be able to write them themselves. However, you can write the template for them.

Sample Interactive Story

This creative writing project was written by a team of four sixth grade students a Prince Hall School they called their story *The Attic Adventure*. (See the procedure tree on the next page.

Template Used

Here is an example of a template of procedures that I used for the creative writing project.

```

TO START
MAKE "STORYEND? "FALSE ;—>
    Sets a flag that will be
    checked in subprocedure SELECT
CLEARTEXT
MAKE "PREFACE [INTOR]
MAKE "INTRO [STAYOUT CASTLE]
MAKE "CASTLE [NO.APPLES APPLES]
MAKE "APPLES [WITCH NO]
MAKE "NO.APPLES [EAT NOT.EAT]
MAKE "WITCH [ESCAPE DELAY]
MAKE "DELAY [RED.DOOR BLUE.DOOR]
MAKE "BLUE.DOOR [CHEST SHUT]
MAKE "FILE "PREFACE ;—> Makes
    the variable name "FILE have the
    value "PREFACE. This is the name
    of the beginning page.
SHOWFILE "PREFACE ;—> Prints
    the file on the screen
SELECT ;—> Subprocedure
    that asks for the user to respond
    with a decision.
END

TO SELECT
MAKE "CHOICE READWORD
TEST :CHOICE = "QUIT
IFTRUE CLEARTEXT PRINT [HOPE YOU
    PLAY AGAIN!] STOP

```

Developing Creative Writing Projects with Text Files in Terrapin Logo -- continued

```

TEST MEMBER? :CHOICE THING :FILE
IFFALSE SELECT.AGAIN
CLEARTEXT
IF MEMBER? :CHOICE STORY.ENDER
  MAKE "STOREND? "TRUE
SHOWFILE :CHOICE
IF :STOREND? = "TRUE STOP
MAKE "FILE :CHOICE
SELECT
END

TO SELECT.AGAIN
PRINT [WHAT DO YOU WANT TO DO?]
PRINT SENTENCE THING :FILE
  [OR QUIT?]
MAKE "CHOICE READWORD
IF :CHOICE = "QUIT CLEARTEXT PRINT
  [HOPE YOU PLAY AGAIN!] TOPLEVEL
IF MEMBER? :CHOICE THING :FILE STOP
SELECT.AGAIN
END

TO STORY.ENDER
OTUPUT [EAT ESCAPE CHEST NOT.EAT
STAYOUT RED.DOOR NO SHUT]
END

TO READWORD
MAKE "CHOICE REQUEST
IF :CHOICE = [] OUTPUT READWORD
OUTPUT FIRST :CHOICE
END

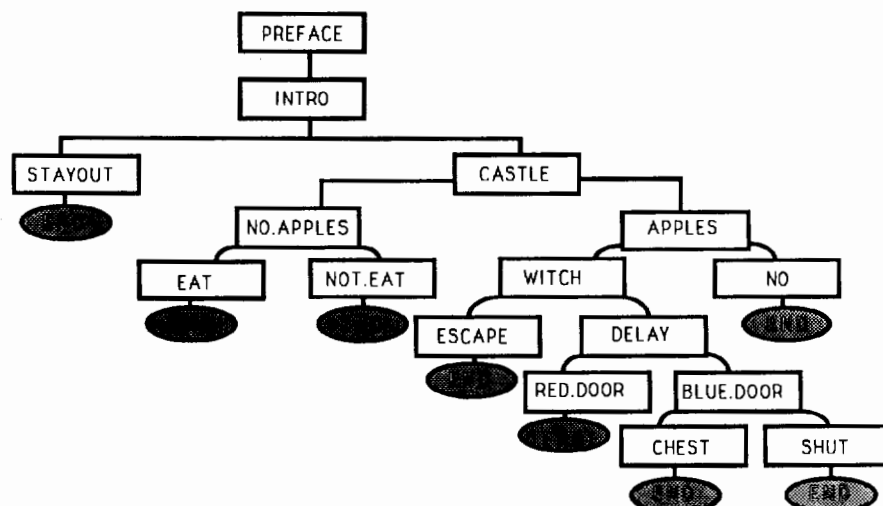
```

A Couple of Final Thoughts

- You can stop the flow of text across the screen with Control-W to aid any slow readers. Pressing any key will resume the flow of text.
- The students don't really have to understand how the procedures in the template work to enjoy and get involved in writing creative stories and then using Logo code to develop an electronic library. The tool procedures can be viewed as mysterious black boxes that are to be enjoyed because they do the job (i.e., radios, TV's, automobiles, etc.) without worrying how they operate.
- Students can add Logo graphics to enhance or embellish the story by plugging a procedure or two into the SELECT procedure. (I leave that problem for you to play with.)
- The idea of having their work electronically preserved excited and delighted the students involved.
- The final product was read over and over again, even after the students knew exactly where each branch led them.
- Try to create your own interactive text adventures. You might surprise yourself with the excitement and sense of satisfaction that results!
- You can write directly to me for complete listings (including one possible resolution for including graphics) at:

Prince Hall School
 Godfrey and Gratz Avenue
 Philadelphia, Pennsylvania 19141
 CompuServe: 76146,306

TREE DIAGRAM of THE ATTIC ADVENTURE



Logo Connections

Software Drivers: Electronic Glue by Glen Bull and Gina Bull

Languages follow an evolutionary path. They improve or die (except for FORTRAN). The BASICs of today are nothing like the Altair BASIC that first appeared on a microcomputer. The first BASIC we used fit in 4K (kilobytes) of memory. We still have it on a paper tape. Micro-Soft QuickBASIC and Borland's TurboBASIC have forced each other to improve as they have quickly moved through versions 1, 2, 3, and 4, with version 5.0 on the way.

The same has been true for Logo. Logo Computer Systems, Inc. (LCSI) developed Apple Logo, followed by Apple Logo II, followed by LCSI Logo II for the Apple. Similarly, Terrapin has developed Terrapin Logo versions 1.0, 2.0, and 3.0 for the Apple, and now a Terrapin ProDOS version (Logo PLUS) is available for the Apple. Each of these new versions has been a distinct improvement over its predecessor.

The refinement and enhancement of Logo has also continued on the Macintosh. Both LCSI and Terrapin have developed versions of Logo for the Macintosh. Coral Object Logo is also an exciting version of Logo available for the Macintosh. A new version of Object Logo has been released that makes it possible to compile a Logo program and give it to someone else, even if they do not own a copy of Logo.

There has been a steady stream of improvements in the Logos available for every computer—except the IBM. At first, there were many versions of Logo for the IBM. Most of these excellent Logos, such as Digital Research's DR Logo, are no longer sold. The official IBM Logo was halted at Version 1.1 many years ago. Even though this is an excellent Logo which has held up very well over the years, it does not reflect many of the newer developments in the IBM world.

This is unfortunate, since the majority of microcomputers in the United States are either IBM or IBM compatible machines. Even though most computers in the schools are still Apples, and the Macintosh dominates desktop publishing and graphics applications, IBM compatible machines are found everywhere else. This means that a child whose family has a computer at home more than likely has access to an IBM compatible computer. The failure to develop newer versions of Logo for the IBM both hampers the teacher who wishes to recommend an educational language for home use, and undermines the claims of Logo as a viable language.

That is why we were very pleased to see the Version 2.0 of LogoWriter issued for the IBM world as well as for Apple. The Apple version of 2.0 was available first, but Version 2.0 for the IBM did not lag far behind. As we reported in last month's column, LogoWriter 2.0 makes possible the interchange of programs between the IBM and the Apple, via using text files created with LOADTEXT and SAVETEXT commands. At that time we did not actually have the new IBM version, but we now have it in our hands as we type this. (Actually, one of us is holding the manual while the other is typing.)

We are pleased to report that at first glance, LogoWriter 2.0 for the IBM appears to be very good. The new version has many desirable features, such as the ability to access subdirectories and external files. The new version also takes advantage of features in the new IBM Personal System 2 (PS/2) series, such as Models 25 and 30. On these machines, Version 2.0 of IBM LogoWriter can display (count 'em) 256 different colors. That's more colors than we can name.

You may *think* that you don't need 256 different colors, or that they serve no educational purpose. However, that's only because you haven't seen them. They are simply glorious. How many times have you attempted to convince a third-grader that she didn't really need red or yellow for a picture, and that orange was an adequate substitute for either? Dozens of colors open the door to psychological experiments, artistic discovery, and sheer playfulness. Combined with a very good FILL command, ornate stained glass designs are possible. It is possible to change the color of the turtle, the line with which it draws, or the background. Of course, LogoWriter 2.0 can only take advantage of capabilities present in a computer display, and cannot add new ones. Therefore, on older IBM models, such as the IBM XT and AT, it will only be possible to see four colors at a time, just as before. (It is possible to upgrade older models with Enhanced Graphics Adapter capabilities, known as EGA graphics, so they can display more colors.)

Charting the Colors

LogoWriter 2.0 encourages the interchange of programs between the IBM and the Apple, via exchange of text files created with LOADTEXT and SAVETEXT commands. (You will need a telecommunications package or a computer equipped with a special card to do this. See the November 1988 "Logo Connections" column for an overview of the process.) However, if you are transferring a LogoWriter program for an IBM PS/2 computer to an Apple, there is an immediate and obvious problem. Let's compare the colors: Color 2 is green on the Apple, but turquoise on the IBM Personal System 2 computer.

Logo Connections -- continued

Number	Apple	PS/2
1	White	White
2	Green	Turquoise
3	Purple	Purple
4	Orange	Red
5	Blue	Dark Blue
6	Black	Orange
7	White	Green
etc. (through 255)		

There are three ways to handle this problem, but one is much better than the others. One possibility is to leave the color numbers unchanged. This means that when a program is changed from IBM to Apple, some of the colors will change.

Another approach is to manually edit each procedure. For example, everywhere it says SETC 7 in IBM LogoWriter, the procedure could be changed to say SETC 2 in Apple LogoWriter. (Green is 7 in IBM LogoWriter, but 2 in Apple LogoWriter.) This process works, but is time-consuming and tedious.

There is a third and much better approach which takes advantage of an idea built into LogoWriter — Logo tools. To take advantage of this method, you must develop a special page in the LogoWriter scrapbook called COLORS for each version of LogoWriter. The procedures on this page of the LogoWriterscrapbook should look like this:

Apple	IBM Personal System 2
TO BLACK	TO BLACK
OUTPUT 0	OUTPUT 0
END	END
TO WHITE	TO WHITE
OUTPUT 1	OUTPUT 1
END	END
TO GREEN	TO GREEN
OUTPUT 2	OUTPUT 7
END	END
etc.	etc.

The procedures for BLACK and WHITE are the same for both Apple and IBM LogoWriter. However, the procedure for GREEN is different, because the numbers for green are different in Apple and IBM LogoWriter.

Use the LogoWriter GETTOOLS command to access the color tools:

```
GETTOOLS "COLORS
```

Once you type this command, the color tools will be available in memory, but will not appear in the LogoWriter

editor. To use the LogoWriter color tools, substitute the name of the color for its number. That is, instead of:

```
SETC 2
```

you should say:

```
SETC GREEN
```

Color tools are not new. They are an obvious idea that many teachers use, and we have talked about them in previous columns. However, now color tools have the additional advantage of making it possible to transfer Logo procedures from the Apple II to the IBM Personal System 2 without changing all the color numbers in the procedures. The GETTOOLS command in LogoWriter makes it particularly easy to adopt this approach. Of course, if equivalent versions of LogoWriter 2.0 were not available for both the IBM and the Apple, there would be no point in discussing the issue.

Universal Controllers

Color tools exemplify a concept we have wanted to talk about for some time: the idea of *software drivers*. This idea is a common one in computer science. Seymour Papert suggested this concept was applicable to Logo long before Logo was available for microcomputers. Papert envisioned a Logo that would be used to control many external devices, such as floor turtles, robotic arms, and scientific instruments. Nearly 20 years ago, Papert wrote,

The hardest problem has been getting the computer to communicate with the device. The approach we have developed centers around the concept of a 'universal controller'... to use it you would program the computer to print a piece of text which might read:

```
!!(!(!(!(!(!(!
```

Knowing that the controller will turn '!' and '(' into turtle signals whose effect will be to cause FORWARD and LEFT steps respectively. Thus any programming language, running on any operating system ... can be used to control a turtle." (Papert and Solomon, 1971, page 39)

Today, we are using exactly the approach that Papert described to use Logo to talk to devices such as speech synthesizers, videodisc players, temperature sensors, and touch screens. The need for a software driver is encountered when Logo code must be developed for different brands of speech synthesizers (such as Echo and Votrax), or different brands of videodisc players (such as Pioneer and Sony). A problem arises when similar devices respond to different

codes. For example, the pitch on one speech synthesizer might be set by sending P1 for a low pitch and P3 for a high pitch, while another speech synthesizer might be adjusted by sending the code F3 for a low pitch and F1 for a high pitch. A table for the two speech synthesizers might look like this:

Pitch	Synthesizer 1	Synthesizer 2
LOW	P1	F3
MEDIUM	P2	F2
HIGH	P3	F1

As you can see, the table of codes for the two speech synthesizers is very much like the color table found earlier in the column.

We would like Logo commands such as PITCH HIGH and PITCH LOW to have the same effect on both speech synthesizers, just as we would like the command SETC GREEN to have the same effect in any version of LogoWriter. Here's how we can do it:

Synthesizer 1	Synthesizer 2
TO PITCH :NUMBER	TO PITCH :NUMBER
.OUT ASCII "P	.OUT ASCII "F
.OUT ASCII :NUMBER	.OUT ASCII :NUMBER
END	END
TO LOW	TO LOW
OUTPUT 1	OUTPUT 3
END	END
TO MEDIUM	TO MEDIUM
OUTPUT 2	OUTPUT 2
END	END
TO HIGH	TO HIGH
OUTPUT 3	OUTPUT 1
END	END

The GETTOOLS command can be used to get appropriate speech tools for different speech synthesizers attached to different computers. Tools permit Logo commands such as PITCH HIGH to be used in place of specific codes such as P3 or F1. The effect is to spare users the necessity of changing their Logo procedures every time they change speech synthesizers.

In the example above, the LogoWriter command .OUT is used to send the letter "P" and the frequency of the pitch out to the speech synthesizer. Curiously, in the LogoWriter manual the ".OUT" command is documented for the IBM 2.0 version of LogoWriter, but not for the Apple 2.0 version of Logowriter. Despite this omission in the documentation, we can attest to the fact that the .OUT command does work in the Apple 2.0 version of LogoWriter, since we are using

the Apple with both speech synthesizers and videodisc players. Incidentally, it has been argued that inclusion of the .OUT command is not important in versions of Logo intended for younger children. We believe the opposite is true. We think it is very important for younger children to have access to versions of Logo that talk — and .OUT or its equivalent is required to send commands to the speech synthesizer.

Electronic Glue

Commands that control a speech synthesizer, videodisc player, or printer are termed *software drivers*. This term is used because the software is used to drive (that is, control) the peripheral device. A software driver is the electronic glue that provides the interface between a programming language and a peripheral.

Versions of Logo that control video images via videodisc players, offer speech capabilities, and have displays with many different colors are inherently desirable. Not long ago most terminals only printed letters on a roll of paper. At that time, Papert had the futuristic vision of using electronic graphics displays in place of teletypes, and said,

A final word about the cost of doing all this. Turtles, music boxes, computer-controlled motors and the like are less expensive than teletypes. Displays are slightly more expensive but becoming rapidly cheaper. So if computers are used in a school, there is no good economic argument for accepting the narrowness of the pure teletype terminal. (Papert and Solomon, 1971, page 39)

The same logic, a few years later, could be used to argue for inclusion of speech synthesizers, and videodisc players, and displays with dozens of colors. The .OUT command provides the tool needed to construct procedures for controlling peripheral devices. The GETTOOLS command makes it possible for LogoWriter to employ these tools in an elegant way.

Reference

Papert, S., & Solomon, C. (1971). Twenty things to do with a computer. *Artificial Intelligence Memo No. 248, Logo Memo No. 3*, Cambridge, MA: Massachusetts Institute of Technology.

Glen and Gina Buill

Curry School of Education, Ruffner Hall
University of Virginia, Charlottesville, VA 22903
Glen's BitNet address is GLB2B@Virginia
Gina's BitNet address is RLBOP@Virginia.

MathWorlds

Edited by Sandy Dawson

Logo and Mathematics Education (LME4) at the International Congress of Mathematics Education: A report from Joel Hillel and Celia Hoyles

The recently held International Congress of Mathematics Education (ICME, Budapest, July, 1988) had a special Working Group on Logo and Mathematics Education. The sessions were co-organized by Joel Hillel (Montreal) and Celia Hoyles (London). This column is a summary of reports from the four sessions held at ICME, as submitted by Laurie Edwards (USA), Tom Kieren (Canada), Uri Leron (Israel), and Richard Noss (England). The sessions within the working group focussed on four key strands within the field of Logo and mathematics education.

Strand One: Logo and Teachers

The four presentations in the first strand raised a number of common themes in addressing the use of Logo in teacher education. These ranged from a consideration of the personal experiences and attitudes of the individual teachers to the social and institutional contexts of computer use.

Joao Matos (Portugal), in describing the MINERVA education project for secondary school teachers, emphasized the importance of creating for the teachers the opportunity for personal mathematical development. This was also evident in the Inset Logo course for primary teachers presented by Candida Moreira (Portugal/UK), and in the Microworlds Project, described by Celia Hoyles, Richard Noss, and Rosamund Sutherland (UK). The Microworlds Project involves research and evaluation of a 30 day course for secondary mathematics teachers in which significant shifts in attitude toward pupils and classroom practice have been detected. As in the Inset programs described by Matos and Moreira, particular attention is paid to the social context of the teachers' learning experience—the importance of discussion, cooperation, reflection and debriefing in the process of building new Logo/math activities. The social context of teacher education was also a focus of the Sunrise School project (Liddy Neville, Australia) which is investigating the nature of effective teacher interventions in a technologically rich environment for inner city children, and attempting to address the question of the influence of social setting on learning.

In summary, the session brought to light a number of issues important in mathematics education generally: the need for time and commitment to both personal and profes-

sional teacher development, the influence of the social context of learning, and questions of pedagogy related to teacher intervention and learner autonomy.

Strand Two: Logo and Geometry

Three common themes of the second strand were (a) extending turtle geometry beyond its usual familiar domain, (b) using turtle geometry to explore geometrical transformations, and (c) linking Logo to school geometry.

Jean Cesar (France) presented a study in which tools for constructing three dimensional solids were supplied. Pupils can construct solids by specifying their faces using ordinary plane turtle geometry, then assemble these using special new procedures. Johnny Lott (USA) reported using list processing to implement a matrix-based approach to transformational geometry. Using these tools, students can experiment with transformations and, in particular, obtain all transformations as products of reflections. Medat Rahim (Canada) applied Logo to Dissection Motion Geometry, in which geometrical figures are dissected, and their parts are transformed and re-assembled to construct new figures of equal area. Chronis Kynigos (UK / Greece) has used Loethe's Post/Direction/Distance microworld as a bridge between the intrinsic turtle geometry and Euclidean geometry.

Several presenters illustrated links between school geometry and Logo. Laurie Edwards (USA) presented a microworld for transformational geometry and reported a teaching experiment using this microworld in which the pupils were involved in studying a single transformation, invariants, group properties (composition, inverse) and creative use of these transformations. Elizabeth Gallou-Dumiel (France) presented research on the learning of reflection and point symmetry. She explained how a structured Logo environment was helpful for learning of the two transformations, and how it modified pupils' mathematical representations in these two cases. Joel Hillel (Canada) described an environment for learning about the properties of several geometric figures, particularly those in an isosceles triangle, and reported how pupils learned to use the tools given them to come to understand the relations and geometric concepts embedded in the tasks.

Strand Three: Recursion and Recursive Thinking

In recent years there has been considerable interest in both the Logo and mathematics communities in the idea of recursive processes and the kinds of thinking which are generated by the construction and use of recursive Logo programs. These issues formed the basis of the third strand.

Tom Kieren (Canada) began the discussion by pointing out that although Logo offers children and adults the opportunity to represent mathematical ideas by sophisticated multileveled recursion, there is also a tendency to represent these same ideas through concatenative recursion and systematic replication—both essentially iterative in nature. Trevor Fletcher (UK) proposed the need to build bridges between Logo activities and existing mathematical curricula, and took as an example a Logo procedure for a binary tree and a traditional piece of combinatorics at grades 11 and 12 focusing on binary arithmetic. Rosamund Sutherland and Teresa Smart (UK) reported the results of a teaching experiment to investigate the nature of grade 11 and 12 pupils' mental models when approaching a problem with a linear recursive solution. Their approach hinged on encouraging pupils to develop a recursive frame which they could then instantiate when presented with the task. Al Cuoco (USA) presented an example of his work with advanced high school students; he reported the activities of two students who worked on the combinatorial problem of determining the number of ordered partitions of a finite set. Bojidar Sendov (Bulgaria) outlined a new version of Logo which has been enlarged by the inclusion of primitives to solve problems in plane geometry (such as CIRCLE, POINT, and SEGMENT). Finally, Uri Leron (Israel) addressed the fundamental question "What makes recursion hard?" He pointed to a complexity gap between recursive procedures and the processes they generate and argued that unless we specifically wish to understand the behavior of the computer (as in debugging), it is often best to suppress process and move directly to the Logo procedure as a description of the product.

Strand Four: The Logo / Mathematics Environment

De Corte et al. (Belgium) reported on a research project, *Computers and Thinking*, in which sixth graders were systematically taught a planning strategy within a Logo learning environment. The researchers identified a strong relationship between the quality of the planning and the time needed to achieve a positive outcome. Hoyles and Noss (UK) reported two studies based around a parallelogram microworld. They found a mismatch between pupils' initial intuitions and formalized definitions, which was at least partially resolved as a result of Logo activities. They also pointed to the importance of synthesis between visual and symbolic modes of representation, and described instances of the computer offering scaffolding for pupils to make sense of partially comprehended geometrical relationships. Bruno Vitale (Switzerland) argued for the importance of mastering the tools of informatics as a way of enriching learning, by introducing procedural thinking through exploration of the strengths and limitations of a formal language. Vitale also

described a psycho-cognitive analysis of pupils' perspectives of what the computer can offer.

In Conclusion

The theme group illustrated the rich variety of research, teacher education, and curriculum development that currently characterizes the international Logo mathematics community. In particular, clear progress has been made in a number of key areas: in understanding the potential of Logo tools for the construction of mathematical representations; in clarifying the role of turtle geometry as offering not only a context for geometrical activities, but also as a medium for expressing and exploring a range of algebraic and more general mathematical ideas; finally, in identifying the significant features of the pedagogical context in which children's Logo work is situated.

Note: Watch this column in future issues of *LX* for information on where to get proceedings for this conference.

Joel Hillel is a professor of mathematics at Concordia University in Montreal, Quebec, Canada. He is one of the founding members of the Logo and Mathematics Education (LME) research group which has met annually since 1985.

A. J. (Sandy) Dawson is a member of the Faculty of Education at Simon Fraser University in Vancouver, Canada. His Compuserve number is 76475,1315. He can also be reached electronically through Bitnet as userDaws@SFU.BITNET

About the Cover

This Christmas Tree was made by third grader Natalie Taylor with help from classmate Teresa Walker. Both girls attend Timothy Ball School in Crown Point, Indiana and were members of the school corporations LogoWriter Club. Their teacher was Rhonda Rawlings.

Assessing Logo Learning in Classrooms

III. Mathematics of Turtle Geometry - Using 360 Degrees by Dan Watt

This is the fourth of nine columns based on a research project which Molly Watt and I have been carrying out with support from the National Science Foundation, "Exploratory Research on Critical Aspects of Logo Learning." In this project, we collaborated with a group of experienced Logo teachers to identify critical aspects of Logo learning and group them under eight headings which were listed in September's column.

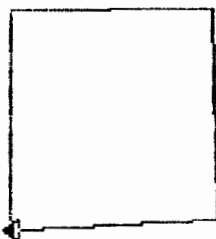
This month I will talk about what we mean by each of the subheadings included in the third cluster of critical aspects, Mathematics of Turtle Geometry - Using 360 Degrees, and will illustrate this cluster with examples of student work. For a fuller sense of what we mean by critical aspects of Logo learning, and our rationale for this approach to assessing Logo learning, please read the September '88 column in this series.

Can We Get to Mathland From Here?

Remember Mathland, Seymour Papert's vision of "... a context which is to learning mathematics what living in France is to learning French?" By "speaking mathematics" to the turtle, and solving geometric problems in an environment with math-speaking peers and teachers, Logo users have the opportunity to learn mathematics in the process of drawing pictures and designs with the turtle. Logo's turtle geometry provides a natural entry point into the exploration and use of important mathematical ideas.

Many schools use Logo to foster computer literacy, problem solving, creativity and personal expression — all highly appropriate educational objectives. But how often is Logo used as a math activity? All too often I have observed that the mathematical understandings needed to use Logo as powerfully as possible are missing from Logo students' work. This month we'll look at some mathematical ideas involving 360 degrees; next month we'll consider some of the other mathematical ideas that arise in turtle geometry.

Here's an example of what I mean. Have you ever seen a "square" like this one?



I remember wincing when a third grade teacher I'll call Gary, proudly showed me that his students had discovered how to draw a square using a rotation of 89 degrees, and then used that idea all year in their other drawings. Gary was an outstanding teacher, who had embraced the "Logo philosophy" enthusiastically, by allowing his students the freedom to discover their own ideas. He supported their efforts by accepting and praising whatever they created. "I never correct my students in their Logo work," he explained. "They'll come up with a 90-degree square when this square no longer serves their purposes." But they went through an entire year without ever finding a need for another type of square, or being asked by their teacher to see if they could find one!

I winced because Gary had missed an important teaching opportunity. The process of drawing a simple Logo square could have been a terrific starting point for his students to explore and eventually understand one of the most useful ideas in turtle geometry: the importance of the total turtle trip, a rotation of 360 degrees, and the usefulness of rotations that involve factors of 360, such as 90 degrees. Gary's students may never use an 89-degree rotation again; but a 90-degree rotation could have been an extremely powerful tool for them, leading to success in many other Logo projects. Understanding the rotational symmetries involved in the total turtle trip, and the interesting modular arithmetic of the 360-degree cycle opens the door to much more powerful use of Logo, and to some of the core mathematical ideas of geometry, trigonometry and physics, and other disciplines using mathematics.

Why is 360 such an important number? Why didn't the mathematicians who invented degree measurements select a nice round number such as 100 or 1000 to be the number of units in a complete rotation? I've read that 360 degrees originated with the ancient Sumerian astronomers who estimated the length of a year — the number of days for the Earth to complete one full orbit or circle around the sun — to be 360 days. But I think this value has been a standard for thousands of years because of another property: 360 has more integer factors than almost any other number you can think of less than 1000.

In our work, we have identified five different levels in the use of 360 degrees by students in grades 4 - 6. They are arranged roughly in the order in which many students learn to use them.

A. Understanding the Relation: Rotation x Number of Sides = 360 for a Regular Polygon.

Again, the square comes first. Students easily see that a square is made up of four 90-degree turns. Some extend this immediately to predict that a triangle would have three

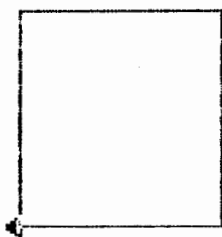
120-degree turns, a hexagon, six 60-degree turns, and so on. A classic Logo circle, REPEAT 360 [FORWARD 1 RIGHT 1] is another well-known example of this principle. Students who don't discover this relation on their own, can readily learn it through directed exploration. The same reasoning can be applied to draw complete spinning designs, that is, designs which repeatedly rotate a given shape through a fixed angle, until the turtle is back where it started. Once students have learned the "Rule of 360," a nice collection of special angles, all factors of 360, becomes available for them to use in other activities.

B. Distinguishing Between Complete and Incomplete Rotations

It is important to distinguish between a complete shape, that is a shape in which the turtle draws a figure and returns to its original state (heading as well as position), and an incomplete shape in which the turtle draws a figure but does not return to its initial state. A technical term for a complete shape is a *state transparent shape* — a shape that ends with the turtle in the same heading and position that it started. A good simple test for a complete shape is this. Invoke the same command twice. If the turtle draws the second shape exactly on top of the first, the procedure is state transparent. Otherwise it is not.

Many students begin by drawing incomplete figures: the turtle returns to its starting position, but not to its original heading. For example,

```
TO SQUARE1
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
END
```



This draws a square, but leaves the turtle rotated 90 degrees from where it started. Either of the next two procedures draws a complete square:

```
TO SQUARE2
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
END
```



TO SQUARE3

REPEAT 4 [FORWARD 50 RIGHT 90]

END

It is important to help students recognize the difference between complete and incomplete shape procedures, and to see the usefulness of procedures such as SQUARE2 or SQUARE3, which can easily be used as modular pieces in other projects.

C. Recognition and use of special angles (factors of 360 degrees and their multiples):

90, 180, 30, 60 120, 45, 135, ...

Students commonly begin by using 90 degrees to make a square. Many generalize to using 90 degree turns to move the turtle easily around the screen. Students begin to use other special angles in a variety of ways: using 180 to reverse the turtle's direction; using 45 as half of 90; combining turns of 30 and 60 degrees to make a right angle; finding that 30, 60 and 90 go together well in making drawings; adding 60 and 60 to get 120, or 90 and 45 to get 135; using 120, 60 and 45 to draw regular polygons, and so on. Eventually some students may realize that some of these angles are factors of 360. Even if they don't discover this for themselves, students who use combinations of special angles can create turtle designs more easily than students who have to work out all their angles by trial and error. And there's no reason why a teacher shouldn't seize the opportunity to show students that the special angles they are using have this interesting relationship to 360 degrees.

D. Using the Total Turtle Trip to Solve Problems Involved in Constructing Figures:

In its fullest sense, the concept of the total turtle trip has two parts:

1. Every time the turtle makes a journey of any kind and returns exactly to its starting position and heading, it will have turned through a total rotation of exactly 360 degrees, or a multiple of 360 degrees (including possibly, 0 degrees).
2. If you repeat a fixed set of turtle commands indefinitely, the turtle will eventually return to its starting point, having made a total turtle trip.

Students who understand these two ideas can use them in their work in a variety of ways. Some designs become easier to complete by first calculating the total angle the turtle has already turned, and subtracting that amount from 360 degrees to see how much it must still turn. In other cases, students can use these concepts to figure out how many repeats they need to complete a figure. In combination, these

Assessing Logo Learning in Classrooms -- continued

two ideas provide a powerful reasoning tool that can be used to solve complex problems that otherwise have to be solved through trial and error. One common problem that lends itself to this type of thinking is finding the turns necessary to draw a right isosceles triangle; another is building a leaf, using two symmetrical arcs.

E. Using State-Transparent Shape Procedures as Modules for Larger Projects

A state transparent shape procedure draws a shape, and leaves the turtle with its position and heading unchanged. SQUARE2 above is an example of such a shape. Although it sometimes takes more forethought to construct a state transparent shape, it is much easier to use such shapes as part of a larger project, than to use shapes like SQUARE1.

Consider. If you want to place a lot of squares in different positions around the screen, a state-transparent procedure lets you clearly separate the procedures that move the turtle from place to place, from the procedure that draws each part of the drawing. With this approach, you never have to think about where the turtle is going to end up, in order to plan the next move.

Most students do not discover this technique for themselves. You'll have to teach it to them, and show them why it's a good idea for their own work, or they'll never use it. Regular polygons and spinning designs provide good introductory examples of this type of procedure.

A Note About Modular Arithmetic and the Logo Turtle

The Logo turtle always has a heading between 0 and 360 degrees. If it rotates more than 360 degrees, Logo subtracts multiples of 360 from the total rotation to place the turtle in its new position. For example, RIGHT 60 and RIGHT 420 produce exactly the same effect, a rotation of 60 degrees to the right. But it is difficult for students to learn this because the turtle does not actually rotate on the screen. It simply redraws the turtle at a new heading calculated by the computer. Students who repeatedly issue commands with large numbers such as

```
REPEAT 1000
  [FORWARD 1000 RIGHT 1000]
```

have no way of observing that a rotation of RIGHT 1000 is the same as RIGHT 280 ($1000 - 360 - 360$).

Similarly, students do not automatically discover that RIGHT 280 has exactly the same effect as LEFT 80

($360 - 280$), unless it is pointed out to them directly or indirectly (through a structured exploration). One approach we have used is to substitute teacher made procedures which slow down the turtle, for the conventional RIGHT and LEFT commands (Watt and Watt, 1986b).

Examples of Student Work Using Ideas About 360 Degrees

1. Andy's polygons and spinning designs (Grade 5):

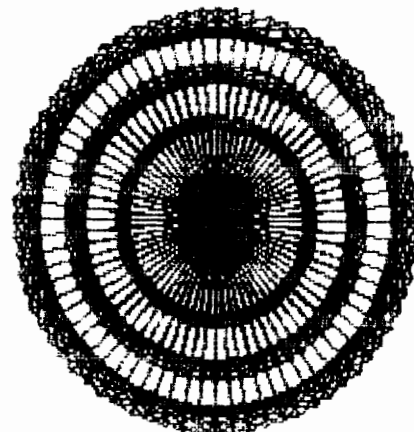
Andy did a lot of work with geometric designs, and used triangles and squares for many of them. When he worked on other polygons, he had difficulty generalizing a rule, although he was attempting to discover one:

```
TO PENTAGON
  REPEAT 5 [BK 50 RT 72]
END
```

```
TO HEXAGON
  REPEAT 6 [FD 60 RIGHT 82]
END
```

(The "rule" Andy was trying was: "Add 1 to the number of repeats and add 10 to the distance and angle). Later, Andy repeatedly rotated his TRIANGLE procedure LT 5, until he got the closed shape shown in Figure 4. He asked his teacher to print out a dribble file of this session for him. (The class was using IBM Logo, and the teacher was collecting dribble files of student work as one form of record keeping.) He then counted the number of rotations, and came up with the following procedure which is not complete, but which looks complete with the turtle hidden.

```
TO LOTS
  REPEAT 70 [TRIANGLE LT 5]
END
```



2. Martha's and Vinh's supershapes and Flag (Grade 5):

Martha's and Vinh's teacher showed them how to use REPEAT to make supershapes, using regular polygons and stars:



```
TO SSTAR
  RT 45
  REPEAT 5 [STAR RT 144]
END
```

```
TO STAR
  REPEAT 5 [FD 30 RT 144]
END
```

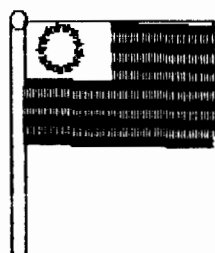
```
TO SPENT
  LT 135
  REPEAT 5 [PENT RT 72]
END
```

```
TO PENT
  REPEAT 5 [FD 20 RT 72]
END
```

```
TO SQ1
  REPEAT 4 [FD 20 RT 90]
END
```

```
TO SSQ1
  REPEAT 36 [SQ1 RT 10]
END
```

In a later project, they applied the same idea (on their own!) to make a flag with 13 stars.



```
TO FLA
  REPEAT 13 [M2 STAR]
END
```

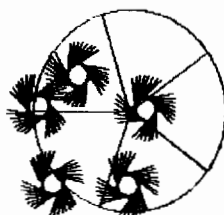
```
TO M2
  PU RT 27.6 BK 7 PD
END
```

```
TO STAR
  REPEAT 5 [FD 6 RT 144]
END
```

When their teacher asked how they decided to rotate the turtle 27.6 degrees in the M2 procedure which is used in FLA, they told her that they had divided 360 by 13! Here we have an example of students applying what they had learned in an assigned project, supershapes, to a student generated project, using the rule of 360!

3. Douglas' PENTACIRCLE (grade 4):

This was one of the classic examples in our research project. Douglas gave this project to his teacher with the following note: "(I wish that the design would go in each part of the circle.)"



```
TO PENTACIRCLE
  HT
  PU LT 90 FD 50 RT 90 PD
  REPEAT 360 [FD 1.5 RT 1]
  REPEAT 5 [RT 90 FD 85.1734 BK 85 LT 90]
  REPEAT 72 [FD 1.5 RT 1]
  PU RT 90 REPEAT 5 [FD 30 LT 90 FD 40 PD]
  D REPEAT 30 [REPEAT 2 [FD 1.5 RT 1] FD 1]
  BK 35 LT 90 PU LT 180 FD 30 RT 180
  LT 72
END
```

Douglas clearly knows a lot about using 360 degrees. He uses it in several contexts in this project. He uses REPEAT 360 [.. RT 1] to draw a circle, and REPEAT 5 [.... REPEAT 72 [.. RT 1]] to divide the circle into five equal parts. And he even used REPEAT 5 [lots of commands including two nested REPEAT's] LT 72], to draw his design and repeat it five times. What was Douglas' problem? And how could his teacher help him? I'll leave these questions for you to explore for yourselves.

4. Jerry's castle (grade 4):



```
TO SIDE.TOWER
  PC 1
  FD 80
  LT 90
  FD 10
  RT 130
  FD 50
  RT 100
  FD 48
  RT 135
  FD 10
  LT 95
  FD 82
END
```

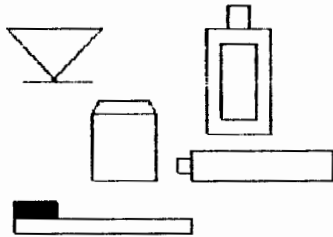
```
TO MIDDLE.TOWER
  FD 70
  LT 45
  FD 20
  LT 90
  FD 20
  LT 45
  FD 70
END
```

Two of Jerry's subprocedures, SIDE.TOWER and MIDDLE.TOWER, show us how he's using angles to draw roofs on his towers. In SIDE.TOWER, the turtle starts by turning LEFT 90, and then makes a total rotation of 365 degrees (RIGHT 130, RIGHT 100 RIGHT 135). Jerry compensates for the extra 5 degrees by turning the turtle LEFT 95 to get it heading straight down. (We can't possibly determine from reading the procedure whether he figured this out by trial and error or by mathematical reasoning.) In MIDDLE.TOWER however, Jerry omits the eaves, and limits the turtle to left turns of 45, 90 and 45 degrees, leaving the turtle positioned correctly for a downward trip to complete the tower. Jerry seems to know something about special angles, but has not completely sorted out how to use them effectively. (A simpler set of rotations to make the roof of SIDE.TOWER might have been 135, 90, and 135, if Jerry had known to use them.)

5. Sheila's cup and (dental) floss drawings (grade 4):

Sheila also uses special angles to draw her CUP and FLOSSY. Sheila knows about angles like 45, and even though she does not seem to know 135, she knows that she can get the turtle heading in the direction she wants by repeating RIGHT 45 three times. She also uses 30 and 60 appropriately in making the top of her dental floss pack. Her procedures are not state-transparent, however.

Assessing Logo Learning -- continued



```

TO CUP
  LT 45 FD 50 RT 45 RT 45 RT 45 FD 67 RT
  45 RT 45 RT 45 FD 50 RT 45 FD 20 BK 50
END

TO FLOSSY
  SEW FD 45 RT 30 FD 10 RT 60 FD 20 FD 15
  RT 60 FD 10
END

TO SEW
  REPEAT 4 [FD 45 RT 90]
END

```

References

Note: Many Logo books discuss the role of 360 degrees in constructing regular polygons. The references listed below go significantly beyond that in dealing with the topics described above.

- Abelson H., and diSessa, A. (1981). *Turtle Geometry*, MIT Press, Cambridge MA. (A formal math text for serious, college or graduate level mathematicians. See Chapter 1 for formal proofs of theorems related to the total turtle trip and 360 degrees.)
- Martin, K. and Bearden, D. (1983). *The Rule of 360*, privately published.
- Papert, Seymour (1980). *Mindstorms*, Basic Books, New York NY. (See Chapter 3, "Turtle Geometry, a Mathematics made for Learning.")
- Papert, S. diSessa A., Watt, D., Weir S. (1979). *Final Report of the Brookline Logo Project*, Logo Memo # 53, MIT Logo Group, Cambridge MA.
- Watt, D. (1983, 1983, 1984, 1985). *Learning With Logo, Learning With Apple Logo, Learning with Commodore Logo, Learning with IBM Logo*, Byte Books, McGraw-Hill, New York NY
- Watt, M. and Watt, D. (1986a). *Teaching With Logo*, Addison-Wesley, Menlo Park CA.
- Watt, D. and Watt, M. (1986b). Starting Logo With Young Children? Slow Down the Turtle!, *The Logo Exchange*, Vol. 5 No. 1, September 1986.

The work described here was conducted at Education Development Center (EDC), 55 Chapel Street, Newton Massachusetts, and supported in part by the National Science Foundation under grant # MDR 8651600, Exploratory Research on Critical Aspects of Logo Learning. The ideas and opinions expressed are those of the author and do not necessarily reflect the views of EDC or the National Science Foundation.

Logo: Search and Research

Research on Teaching

Problem Solving

by Douglas H. Clements

Did you monitor your own problem solving as you were working with Logo since last month? Did you find that your own problem-solving activity involved "figuring out what to do when you don't (immediately) know what to do?" Could you identify any metacomponents, the *executives* of your mental company, directing this effort? Recall that they include:

- deciding of the nature of the problem, including the use of basic schemes,
- deciding on a representation,
- deciding on a strategy, which is involves many abilities for the "good strategy user," and
- solution monitoring.

If you didn't, do so soon. Such self-reflection will make all future discussion of problem solving more meaningful and useful. This month, we will discuss research-based methods for helping students develop these abilities. These methods tend to focus on mathematical problem solving. This reflects not only my own interests, but also the numerous research efforts that have been made to understand mathematical problem solving. Given the emphasis mathematics educators place on problem solving (e.g., National Council of Teachers of Mathematics, 1987), and Logo's mathematical structure, this is not a bad place to start. We should remember, however, that problem solving should underlie teaching in all areas. Certainly, people use their metacomponents to solve problems in all areas of their school and out-of-school lives.

As usual, there is no single best way to translate what little we know about problem solving processes into classroom practice. Research does tell us, however, that students benefit when we discuss, model, and prompt the use of these processes. There are no rigid "steps" to be followed in a lockstep manner—people jump from one process to another during all phases of problem solving. However, for planning instruction it is useful to employ four broad phases. Table 1 shows these phases and describes what both students and teacher should be doing in each. As you read them, picture how you might use them to teach a specific problem.

Table 1*

Students...**1. Understand the problem**

- Read the problem again
- Write what you know
- Tell the problem in your own words
- Look for key phrases
- Find the important, or missing, information
- Ask questions for clarification if necessary
- Tell what you are trying to find

2. Solve the problem

- Make sure you understand the problem
 - Tell each other what you know, what you are trying to find, and what you might do to find it
 - Agree among yourselves what the problem is all about
- Use an appropriate representation and strategy
 - Draw a diagram or picture
 - Act it out
 - Look for a pattern.
 - Make an organized list or table
 - Guess and test (systematically)
 - Solve part of the problem
 - Solve a simpler, but similar, problem
 - Work backwards from the solution

- Monitor the solution process

3. Check and answer

- Check (final monitoring)
 - Check if the right question was answered
 - Check if all relevant information has been used
 - Check if the answer makes sense
 - Check computations
- Write the answer in a complete sentence

4. Extend knowledge of problem solving

- Discuss solutions (your own and others')
- Analyze what worked and why
- Solve the problem in another way
- Make up and solve a similar problem
- Identify generalizations

Teachers...**Help students understand the problem**

- Present the problem to the class
- Answer questions about the problem
- Lead a discussion about the nature of the problem
- Focus on the questions and the given information
- Discuss implicit as well as explicit information
- Ask: What is the problem?
- Ask questions that can be answered mentally
- Discuss possible solution strategies (but avoid evaluating them at this time)
- Organize students to begin work (small groups are often best)

Help students further understand and solve the problem

- Encourage students to share and argue about ideas
- Analyze difficulties
- Observe and question students to identify "where they are" in the problem-solving process.
- Provide prompts and hints
- Pose questions that encourage monitoring and reflection. Consistently pose the following three questions to groups or individuals (from Schoenfeld, 1985).
 1. What exactly are you doing? (Can you describe it?)
 2. Why are you doing it? (How does it fit into the solution?)
 3. How does it help you? (What will you do with it when you're done?)

They may not be able to answer you well, especially at first, but it will serve as a powerful prompt for them to monitor their progress. They should be challenged to critique their own strategies. Eventually, they will internalize the questions and begin to query themselves.

Help students check and answer**Ask students if they have checked their solution**

- Was the right question was answered?
- Was all relevant information used?
- Does the answer makes sense?
- Are computations correct?

Help students extend their abilities

- Allow students (about 3-5) to show their solutions
- Name the solution strategies used
- Encourage analysis
- Have students make up similar problems and identify possible generalizations.
- Help children to make connections between these problems and others. They should identify other problems that can be solved with similar strategies.
- Have children apply what they have learned to a variety of contexts.
- Even if students are taught a strategy...even if they use it, they might now use it the next time. To encourage this transfer, ask them (and as soon as possible have them ask themselves) Pressley's three questions:
 1. Which strategy worked for me?
 2. Why was it best (i.e., in this situation)?
 3. What will I do the next time?

* From Clements, 1989; loosely adapted from Charles & Lester, 1984.

Search and Research -- continued

For many problems, these phases might all take place during a single lesson. For others, and for many Logo-based problems, the first phase might occur with the whole class on a Monday, say, and the third and fourth at the end of the week when all students had a chance to work on the problem.

Research has supported the effectiveness of such an approach (Charles & Lester, 1984). Several conclusions in a review of research on problem solving by Suydam (1982) support, and serve as a partial summary of, what we have said:

- Problem-solving heuristics, or strategies, can be taught and they help students achieve correct solutions.
- There is no single optimal strategy for solving problems. Students need a wide repertoire of strategies.
- Problem solving is best improved by integrating it into the curriculum.

Teachers should:

- include a variety of problems so they develop creative approaches to problem solving;
- encourage multiple solution strategies and emphasize that mathematics is more than facts and algorithms;
- provide sufficient time for discussion, practice, and reflection; and
- provide wait time. Whenever you ask a question, wait at least three seconds before doing anything else (asking another question, asking another student, giving a hint, etc.).

In addition, research indicates that it is important to:

- Recognize the importance of domain-specific conceptual understanding.

It may help to teach students to use heuristics, but we must realize that they are never divorced from conceptual understanding. That is, the heuristic "make a drawing" is of little use apart from some knowledge about what to draw! This is another way of saying that students' schemes must be developed simultaneously with heuristics and other problem-solving processes. It may be that increases in problem-solving performance that follow training in heuristics cannot be accounted for just by using new heuristics. Possibly they are the result of increased understandings about underlying concepts, which in turn is the result of activities like drawing pictures or searching for patterns (Lesh & Zawojewski, 1988). This is good news. Because students must organize their own knowledge, problem-solving experiences can

contribute to knowledge development as much as knowledge development contributes to problem-solving ability.

- Be a good model of a problem solver.

Work out some problems in front of the students (Shoenfeld, 1985). Do not do them "perfectly." A reasonable "stream of consciousness" might be: "Let's see. We have to find out _____. We know _____ and _____. We could try _____, or _____, or _____. What do you think is worth a few minutes try at first?" and so on.

At other times, serve as the "monitor" only, writing down students' ideas and raising general questions ("Do we all understand the problem before going on?", or, in the middle of a solution attempt, "Is this working? Are we getting somewhere?")

Encourage creative thinking.

Davis (1985) believes that math must be creative. Why?

- It's impossible to teach rules for every problem, every situation. Students must be able to solve problems creatively.
- Students are creative if they are allowed and encouraged to be. We shall see examples of kindergartners creating significant mathematics, on and off the computer.
- Approaching mathematics as a creative activity increases students' motivation.
- Originality is a goal of all education, including mathematics education.
- Mathematics in the real world is not rule-based and routine.
- Mathematics is not static. More new mathematics is created every month!

Certain types of questions stimulate creativity. For example: "Can you think of a problem just like this one that you have solved before?" "Can you create and solve a similar problem?" "Can you solve the problem in a different way?" "Can you solve a more general problem?" Each of these questions asks students to "Keep thinking!"

- Address students' belief systems.

What students believe about mathematics and mathematical problem solving strongly influences what they do. Most students believe the following (Shoenfeld, 1985).

- Mathematics is facts and rules with one way to get one right answer. You find the rule and get the answer.
- If you don't solve a problems in 5 minutes, then you'll never solve it. Give up.
- Only geniuses discover or create (or re-create) mathematics. So if you forget something, you'll never be able to figure it out on your own. Also, you can't really understand why things (e.g., algorithms) work the way they do.

Work to change these beliefs. Have students solve problems in a myriad different ways. Have them discuss—and argue about—their solutions and ideas. Work in groups on problems that take some time to solve. Respect student's ideas, regardless of whether they are "right" or "wrong."

One more suggestion: Do not (unconsciously) give in to common (sometimes unconscious!) student behaviors that dilute problem-solving activities. When such activities become challenging, students often complain. "We don't understand." "No one else expects us to do such difficult work." "You aren't teaching us how to do this." They often get their parents behind them, especially on the last one. It is often easy to simplify, explain, or omit activities in the face of such negative reactions. Resist such temptations. Meet the resistance squarely. With some groups, you might even discuss it. After meeting your challenges successfully...after changing their beliefs about school, about mathematics, about learning, and about the satisfaction of solving problems, these negative reactions will taper off. And you will have affected a change with more powerful ramifications than any other.

- Develop an accepting, trusting atmosphere.
- Make most of your lessons problem solving lessons.

In the next column, we'll turn to research that specifically examines Logo and problem solving processes. Before then, try out the phases in Table 1 with your students. What works for you and for them? What doesn't? Why?

References

- Charles, R. I., & Lester, F. K. (1984). An evaluation of a process-oriented instructional program in mathematical problem solving in grades 5 and 7. *Journal for Research in Mathematics Education*, 15, 15-34.
- Clements, D. H. (1989). *Computers in elementary mathematics*. Englewood Cliffs, NJ: Prentice-Hall.
- Davis, R. B. (1984). *Learning mathematics: The cognitive science approach to mathematics education*. Norwood, NJ: Ablex.
- Lesh, R., & Zawojewski, J. S. (1988). In T. R. Post (Ed.), *Teaching mathematics in grades K-8: Research based methods* (40-77). Boston: Allyn and Bacon.
- National Council of Teachers of Mathematics. (1987). *Curriculum and Evaluation Standards for School Mathematics (draft)*. Reston, VA: Author.

- Shoenfeld, A. H. (1985). Metacognitive and epistemological issues in mathematical understanding. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 361-379). Hillsdale, NJ: Erlbaum.
- Suydam, M. N. (1982). Update on research on problem solving: Implication for classroom teaching. *Arithmetic Teacher*, 29(6), 56-60.

Douglas H. Clements, SUNY at Buffalo
Department of Learning and Instruction
593 Baldy Hall, Buffalo, New York 14260
CIS: 76136,2027 BitNet: INSDH@UBVMSA

Enjoy a summer week or two in scenic Oregon!
Increase your leadership and Logo skills.
Attend the ICCE workshops:

Logo for Leaders: Beyond Turtle Graphics

Topics:

- Planning the implementation of Logo on a building or district level
- Integrating Logo into the curriculum
- Logo and problem solving
- Comparison of versions of Logo
- Beyond the beginning: Logo syntax and grammar
- Hands on time for development of Logo projects
- Assessing Logo learning
- Writing for publication

Staff: Sharon Burrowes Yoder and Dave Moursund

Dates: July 16 - July 22, 1989

Location: Eugene, Oregon

Cost: \$300 before May 1st; \$350 after May 1st; covers 2 quarter-hours graduate credit, materials, and instruction.

Leadership Development

Topics:

- Long range planning for computers in schools
- Effective inservice and staff development
- Computer/brain/problem solving
- The computer coordinator
- Frontiers and the future

•Grant writing and other writing for publication

Staff: Dave Moursund and Sharon Yoder

Dates: July 9 - July 15, 1989

Location: Eugene, Oregon

Cost: \$300 before May 1st; \$350 after May 1st; covers 2 quarter-hours graduate credit, materials, and instruction.

Receive a 10% discount if you attend both!

For more information, contact Dave Moursund or Sharon Yoder at ICCE, 1787 Agate Street, Eugene, Oregon 97403 503-686-4414

Global News

Edited by Dennis Harper
University of the Virgin Islands
St. Thomas, USVI 00802

Logo Exchange Continental Editors

Africa	Asia	Australia	Europe	Latin America
Fatimata Seye Sylla	Jun-ichi Yamanishi	Jeff Richardson	Harry Pinxteren	Jose Valente
Lab Informatique et Ed	Faculty of Education	School of Education	Logo Centrum Nederland	NIED
BP 5036 Dakar	Toyama University	GIAE	P.O. Box 1408	UNICAMP
Senegal, West Africa	3190 Gofuku	Switchback Road	BK Nijmegen 6501	13082 Campinas
	Toyama 930 Japan	Churchill 3842	Netherlands	Sao Paulo, Brazil
		Australia		

1988 comes to an end and thanks to the winter and summer Olympics North Americans are once again reminded that there are other countries in the world. Just as many countries enjoyed success in the Olympics (some even in "American" sports), so too are other countries enjoying success in using Logo in schools. Readers of this column are aware of the advanced state of Logo usage in countries such as Australia, Israel, Iceland, England and the Netherlands. This month's Global Comments describes Logo activities in two countries where the language has expanded quite a bit during the past year — Finland and Brazil. The Finnish report is about one classroom's experience with using Logo to promote traffic safety while Jose Valente describes three Logo projects centered at the University of Campinas.

The following text was submitted by the Kari Hämynen who is a primary school teacher in Finland. He has utilized a computer in many subjects in his school which is an experimental school of computer science supervised by the National Board of General Education. As coauthor of the report, Hannu Korhonen teaches mathematical subjects in the normal school of the University of Jyväskylä. At the moment he works as a research officer at the University's Institute for Educational Research, specializing in the use of Logo in mathematics instruction.

Enter the Electronic Girl

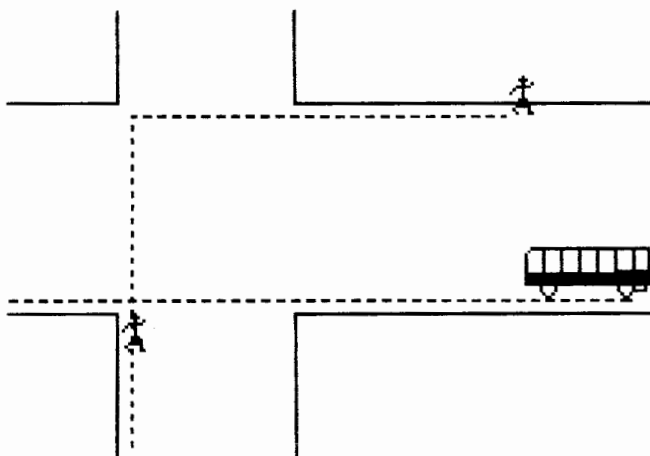
A girl, a mother with a baby, a grandfather, and a city bus on an adventurous tour are on a computer screen in Kypärämäki school, Jyväskylä, Finland. The electronic girl



is on her way to school. She is guided by a real girl using the computer, who gives her commands in her own language

(Finnish) to proceed and turn and wait. Logo interprets the words of the real girl as instructions for the electronic girl.

The route of the electronic girl corresponds to the map of the real girl's road to school. While guiding the electronic girl, the real one has to contemplate on the dangerous spots along the school route and the rules for safe walking. There is plenty of time to think about the various situations, because the electronic girl and bus have the patience to wait for the next instructions.



An erroneous selection does not result in a bloody accident. Thinking is not hampered by the many nonessential factors that occupy the attention of a little walker in a real traffic situation. This is a means of using a tool with which the child can safely plan her movements on the way to school.

The Commodore-Logo we had at our disposal made it possible to create our own creatures, with sprites, and to guide them with ordinary Logo commands. In the end our "traffic Logo" had six different creatures: a girl, a boy, an old man, a mother, a car, a bus and a bike. Some of them already existed on the auxiliary program disk while others were created by the students using the sprite-editor.

The sprite turtles were given names using small programs that allowed us to identify them:

```
TO GIRL
TELL 4
SETH 90
PC 7
ST
END
```

The bus was enlarged with the commands BIGX and BIGY. Furthermore, it had to be turned to move horizontally and painted the real color.

```
TO BUS
TELL 1
BIGX
BIGY
SETH 90
PC 2
ST
END
```

In fact we also tried to turn maps into Logo pictures, but we did not have sufficient knowledge. The drawing of street intersections with turtle graphics could of course have been possible, but it seemed far too laborious because the school vicinity has many crossings. Nor were we able to convert the maps digitized by a video camera into Logo pictures. That is why we ended up with a simple poor man's solution. The maps were drawn on transparent plastic which stuck to the screen because of static electricity without paste or stickers.

A typical traffic situation is a girl crossing a road. She stops on the side of the road to wait, lets the bus go past, crosses the street and turns to walk the left side of the intersecting street. In Logo the instructions look like this:

```
TO SITUATION1
GIRL
WALK 50
WAIT 5
BUS
DRIVE 150
GIRL
WALK 40
RIGHT 90
WALK 80
END
```

It is possible to create a lot of activity in the intersection by combining the programs made by several pupils. The traffic Logo produces movement that is missing in the textbook illustrations without the disturbing factors of real

life traffic situations. The traffic Logo differs from a video film taken in real life situations in that the pupil is not an outside observer, but she participates in the traffic in the shape of an electronic creature. She is the one who walks on the screen-street and contemplates her movements and turns as well as the effects of traffic signs and the various factors to be observed in the surroundings.

Logo was not the only instructional material used during the traffic instruction phase. The lesson was started by coloring the safety-cautious people green and the accident prone people red in the textbook. Safe movement was then planned with the help of the traffic Logo. During the second lesson we listened to a school radio program, followed by the traffic Logo situations produced by classmates, and talked about traffic. After all this, all the children left for home along a now far safer route.

For more information contact Hannu Korhonen, Institute for Educational Research, University of Jyväskylä, Seminaarinkatu 15, SF-40100 Jyväskylä, Finland.

Research in Brazil

Our Latin American correspondent tells us about some Logo research activities in Brazil.

The research objective of the University Estadual of Campinas' Laboratory for the study of informatics applications to education is to study new educational practices which can take advantage of the computer presence. This work includes the development and evaluation of environments for learning concepts in science, engineering and the arts. The work also addresses the practical issues of providing training and continuing support to teachers involved in the Laboratory's work.

The Laboratory has several research projects which involve the use of Logo in public schools and in special needs settings. The results of this research and experiences in the schools are made available to the public on an ongoing basis. What follows is a summary of the principal Logo projects and research in development:

1. EDUCOM (COMputers in EDUcation). This project has four major areas of investigation:

- to develop a Logo methodology to be used with secondary students,
- to use the Logo environment to investigate the differences in cognitive and programming styles,
- to study how Logo can help students to learn the concept of variable in mathematics, physics, and biology
- and to study the effectiveness of physics and biology microworlds.

Global News -- continued

Two public schools are involved in this project. Seven teachers from each school were selected. These teachers chose to work with a 5th and a 12th grade classroom (approximately 80 students per school.) A laboratory with 20 computers was set up in each school and the students spent 6 to 8 hours per week developing Logo activities in mathematics, physics, biology, chemistry, and Portuguese, at the high school and mathematics, Portuguese, English, science and art at the junior high level. This project is supported by the Ministry of Education and government research funding agencies.

2. Informatics in Special Education.

The objective of this project is to investigate the use of Logo with hearing impaired and physically handicapped children. Two institutions, one for physically handicapped and another for hearing impaired are involved in this project. A laboratory with five computers was set up in each institution and Logo is used with 30 physically handicapped children from 5 to 12 year olds, and 7 hearing impaired from 5 to 7 years old. A Logo educational methodology was developed and several investigations are taking place specifically in the area of using Logo as a tool to diagnose these children's intellectual development:

- spatial knowledge in physically handicapped children,
- abstract reasoning in hearing impaired children, and

3. Training in Computers in Education

The Laboratory offers courses to undergraduate and graduate students from the University of Campinas, and teachers from public and private schools interested in using Logo as an educational tool. These individuals have a chance to get involved in the Laboratory's projects and to develop specific research questions on these topics. This project is supported by the Laboratory of Informatics Education.

For further information, please write to our Latin American correspondent.

I Didn't Teach Spelling Last Week

by Bill Craig

In 1986, I organized a small group of teachers in our school division to investigate the uses of Logo in the elementary school curriculum. We continued to meet during the 1986-87 school year and began to look at the advantages of using LogoWriter with our students. All of these teachers are enthusiastic about Logo's potential but are also faced with tremendous demands on their instructional time. The difficulty these teachers constantly face is how to both provide Logo instruction and meet their curriculum requirements. During a discussion of this problem, one maverick in the group volunteered that she had not taught spelling the previous week so that her students could have some Logo time. For teachers who are unwilling to give up spelling instruction, the challenge of Logo users becomes identification of Logo applications which combine Logo with instruction in spelling, mathematics, social studies, science, or any other subject. A modification of one of the projects from the LogoWriter project books provides such an application.

The second LogoWriter project book, *Word Adventures*, describes an interesting programming project in which students write procedures to code and decode words. The project is based on the procedure, CODE.

```
TO CODE
TOP
REPEAT TEXTLEN [CF INSERT "A]
END
```

The primitive TOP moves the cursor to the top of the text screen. TEXTLEN reports a number equal to the number of characters on the page. CF moves the cursor forward one space and INSERT places a character at the position of the cursor without a carriage return. In CODE, the cursor is first moved to the top of the page, and for every character on the page (as counted by TEXTLEN), the cursor moves forward one space and inserts an A.

If the word BILL is CODEd, the output is BAIALALA. The result of CODEing HI THERE is HAIAATAHAEEARAEA. More complex codes can also be invoked as illustrated by CODE2:

```
TO CODE2
TOP
REPEAT TEXTLEN/2 [CF CF INSERT "Z]
END
```

CODE2 inserts a Z after every second character. If CODE2 is applied to BILL, the result is BIZLLZ. The result of applying CODE2 to HI THERE is HIZ TZHEZREZ. Notice the use of TEXTLEN/2 which counts the number of characters to be coded and divides the number by 2.

Whatever can be CODEd can be DECODEd and the LogoWriter project book suggests that the next programming assignment be the design of a procedure to decode coded words and messages. For example, DECODE will decode a message coded with CODE:

```
TO DECODE
TOP
REPEAT TEXTLEN/2 [CF DELETE]
END
```

Because this procedure erases instead of inserting characters, it need only be repeated TEXTLEN/2 times. Here is a decoding procedure for CODE2.

```
TO DECODE2
REPEAT TEXTLEN/3 [CF CF DELETE]
END
```

I was initially attracted to CODE and DECODE as interesting programming activities which introduce some text primitives in a unique way. But these procedures have potential as tool procedures beyond their value in programming instruction. My idea is to provide the CODE procedures with a list of words, such as weekly spelling words, and let students attempt to decode the words that the computer codes. The emphasis of the activity now shifts from programming to spelling and pattern detection.

I modified CODE so that it uses an input:

```
TO CODE :WORD
PRINT :WORD
TOP
REPEAT TEXTLEN [CF INSERT :A]
END
```

To CODE my name, I enter CODE "BILL .

I then wrote a main procedure, GENERATE. When GENERATE is entered, the computer randomly picks a word from a teacher entered list and codes the word using the CODE procedure. In its simplest form, the procedure looks like this:

```
TO GENERATE
CODE PICK WORD
END
```

GENERATE uses PICK.WORD which selects a word randomly from a list entered with the procedure.

```
TO PICK.WORD
OUTPUT PICK [COLLEEN ELSONIA
SHIRLEY SUE ELAINE DIANA]
END
```

The procedure now contains the names of teachers in our Logo group but any names, spelling words, or vocabulary words can be used. The procedure PICK.WORD uses the PICK cited in many sources. (Bull and Cochran, 1984).

```
TO PICK :LIST
OUTPUT ITEM ( 1 + RANDOM COUNT
:LIST)
END
```

GENERATE simply CODEs randomly selected words without checking students' responses for accuracy. The procedure can be written so that it will accept and check responses.

```
TO GENERATE
CT
RG
HT
MAKE "WORD PICK.WORD
CODE :WORD
PRINT []
PRINT []
INQUIRE
REPLY
END
```

```
TO INQUIRE
PR [WHAT'S THE WORD?]
MAKE "RESPONSE READLIST
END
```

```
TO REPLY
IF :RESPONSE = :WORD [PRINT [YOU GOT IT!] STOP]
(PRINT [THE WORD IS ] :WORD [.] )
END
```

Now GENERATE codes a word and asks students to enter the word in its uncoded form. The procedure checks the word that is entered, tells the student if the guess is correct, or prints the correct word if the guess is incorrect. Someone with more patience and/or programming skill could design a much more elegant checking routine.

GENERATE codes words using just one CODE. The procedure can be modified so that it picks codes, as well as words, randomly.

I Didn't Teach Spelling Last Week -- continued

```
TO GENERATE
CT
RG
HT
CHOOSE.CODE
RUN.CODE :WORD :NUM
INQUIRE
REPLY
END
```

```
TO CHOOSE.CODE
MAKE :WORD PICK.WORD
MAKE "NUM RANDOM 3
END
```

```
TO RUN.CODE :WORD :NUM
IF :NUM = 0 [CODE :WORD STOP]
IF :NUM = 1 [CODE2 :WORD STOP]
IF :NUM = 2 [CODE3 :WORD STOP]
END
```

```
TO CODE3
TOP
REPEAT TEXTLEN [CF INSERT [ZQ] ]
END
```

Another twist can be added by asking students to code a word using the same code that was used to disguise the word they just guessed. For instance, the computer might use CODE3 to produce the following word from Logo:

LZQOZQGZQOZQ

Once a student makes a guess and is given feedback, the computer picks another word from the list for the student to code using the same rule. The revised procedure, GENERATE2, adds this more difficult exercise.

```
TO GENERATE2
CT
RG
HT
CHOOSE.CODE
RUN.CODE :WORD :NUM
INQUIRE
REPLY
RECODE
END
```

```
TO RECODE
PRINT []
MAKE "WORD2 PICK.WORD
(PRINT [NOW CODE THIS WORD: ]
```

```
MAKE "TRY2 READLIST
PRINT [TYPE RETURN TO SEE THE
ANSWER.]
MAKE "TRY3 READLIST
CT
RUN.CODE :WORD2 :NUM
PRINT []
PRINT []
PRINT [HERE'S YOUR RESPONSE AGAIN, ]
PRINT :TRY2
END
```

Rather than telling a student whether he is correct, RECODE prints the correct answer and allows the student to compare his guess with the correct answer. One of the teachers in the group is a resource teacher for learning disabled students. She was enthusiastic about the procedures but believed that the activities were too difficult for students with visual discrimination problems. The CODE procedures can be modified so that symbols, rather than letters, are used to code the words.

```
TO CODE :WORD
PRINT :WORD
REPEAT TEXTLEN [CF INSERT "$]
END
```

```
TO CODE3 :WORD
PRINT:WORD
TOP
REPEAT TEXTLEN [CF INSERT [#*] ]
END
```

Work with procedures that code with symbols instead of letters might be a good initial activity for all students.

These activities are different in purpose than those described in the LogoWriter project book. The CODE procedures are no longer programming exercises but are tool procedures designed to provide something as mundane (and necessary) as spelling drill or as sophisticated as experience in pattern detection. My objective is to begin to provide a solution to the limited time dilemma of teachers interested in using Logo but burdened by curriculum requirements. My hope is that with these types of activities, teachers will no longer have to choose.

References

- Bull, Glen and Cochran, Paula, "Listful Thinking," *National Logo Exchange*, September, 1984.
Word Adventures, Learning With LogoWriter, Logo Computer Systems Inc., 1986, New York.



Logo Exchange

The Journal for Logo Activities Worldwide

Now published by the International Council for Computers in Education, *Logo Exchange* brings you exciting ideas from top Logo educators throughout the world.

Each of the nine issues includes:

- Logo articles *by* classroom teachers *for* classroom teachers;
- Columns which emphasize practical ideas for the use of Logo at the primary and intermediate level;
- Articles focusing on the use of Logo in language arts and math;
- Updates on Logo research;
- A look at Logo-like activities, software and hardware;
- Articles by Logo leaders from throughout the world;
- Reports on Logo activities world-wide.

The *Logo Exchange* is the journal for ICCE's Special Interest Group for Logo-using educators. SIGLogo members are invited to participate in local, regional, and national meetings and to contribute to the flow of ideas through the *Logo Exchange*.

Logo Exchange is published monthly September through May. SIGLogo membership is \$24.95 for ICCE members (include membership number on order) \$29.95 for non-members. Add \$5.00 for non-U.S. memberships. All billed orders are charged \$2.50 for handling.



To order, contact:

ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905; ph. 503/686-4414.

Get the Logo Alphabet on disk FREE!

The Logo Alphabet is a set of procedures that draws all of the letters of the alphabet in the size of your choice. They were developed by Tom Lough, founding editor of the *Logo Exchange*. Use this special coupon below to take advantage of this offer to get the Logo Alphabet free when you join SIGLogo or extend your current membership for a year.

☐ Yes, I would like to join SIGLogo and receive the Logo Alphabet disk free of charge.

☐ No, I do not wish to join SIGLogo at this time, but I want the Logo Alphabet disk. I am enclosing \$7.50.

Please check the version that you want:

☐ Apple Logo

☐ Apple Logo II or LCSII Logo II

☐ Apple LogoWriter 1.1

☐ Apple LogoWriter Version 2.0

☐ Terrapin Logo

☐ Terrapin Logo Plus

SIGLogo Membership (includes the Logo Exchange)

ICCE member

Non ICCE member

U.S.

\$24.95

\$29.95

Non-U.S.

\$29.95

\$34.95

ICCE Membership (includes The Computing Teacher)

1 year

U.S.

\$28.50

Non-U.S.

\$31.00

Add \$2.50 for processing if payment does not accompany your membership dues. VISA and Mastercard accepted. Add \$18.00 per membership for airmail shipping of *Logo Exchange* outside the U. S. and Canada.

☐ Payment enclosed. Amount \$ _____

Name _____

☐ Visa ☐ Mastercard

Address _____

Expiration Date _____

State _____ Zip/Postal code _____

Name on Card _____

Phone Number _____

Card number _____

☐ Bill me. Add \$2.50 for handling

ICCE Membership number _____

NECC '89 **Connections**

9th National Educational Computing Conference *Invites you to attend*

Papers, Panels, Projects, Exhibits, Full-Day Pre- Conference Workshops

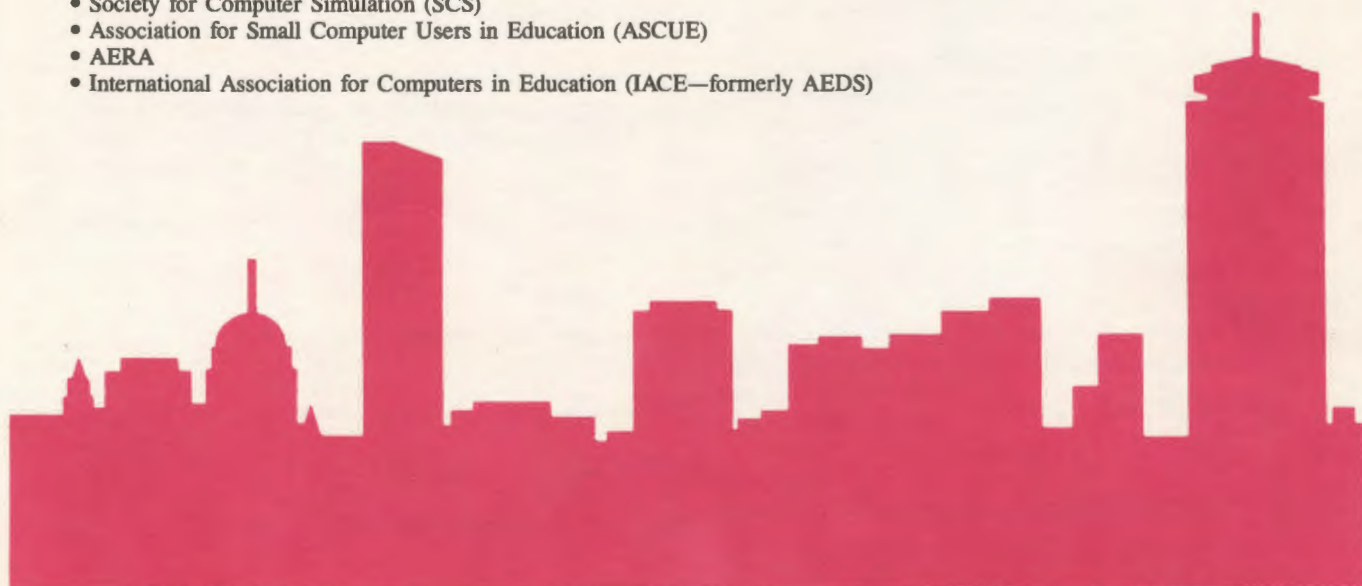
For information contact:
Susan Friel and Nancy Roberts
Lesley College
29 Everett Street
Cambridge, MA 02138-2790

For information on Exhibits contact:
Paul Katz
NECC '89 Exhibits
University of Oregon
Continuation Center
1553 Moss Street
Eugene, OR 97403-9905

NECC '89 is hosted by Lesley College and Bolt Beranek and Newman, Education Development Center, Massachusetts Computer Using Educators, and Technical Education Resource Centers.

NECC '89 is sponsored by the International Council for Computers in Education (ICCE) in cooperation with member organizations of the NECC Steering Committee:

- Association for Computers and the Humanities (ACH)
- Association for Computing Machinery (ACM) Special Interest Groups on:
 - Computer Science Education (SIGCSE)
 - Computer Uses in Education (SIGCUE)
 - University and College Computing Services (SIGUCCS)
- Educational Computing at Minority Institutions (ECMI)
- IEEE Computer Society
- EDUCOM/EDUNET
- Society for Computer Simulation (SCS)
- Association for Small Computer Users in Education (ASCUE)
- AERA
- International Association for Computers in Education (IACE—formerly AEDS)



Boston, Massachusetts

June 20-22, 1989