

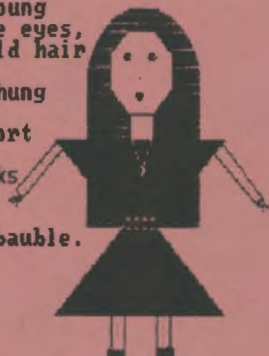


LOGO EXCHANGE

MAY 1989

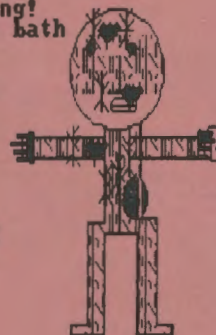
VOLUME 7 NUMBER 8

Elonwy is a young
woman with blue eyes,
and reddish gold hair.
She has a
crescent moon hung
from a chain,
She wears a short
white robe,
and silver links
around her
waist and
She carries a bauble.
by Priya Mohan

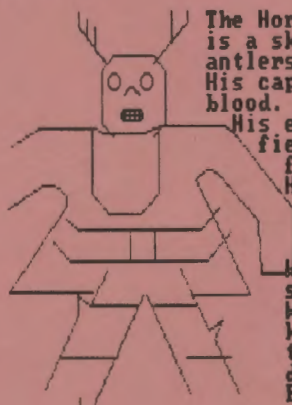


Oh there you are Gurgi
you dirty little thing!
You are in need of a bath
you know.
Gurgi you smell
like a wet wolfhound
you know.
Gurgi you
found Hen Men
you know.

GREGORY
NOCE



The Horned King's mask
is a skeleton with two
antlers.
His cape is as red as
blood.



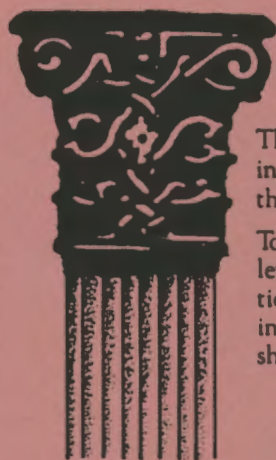
His eyes are as
fiery as a
furnace.
His legs are
metal armor.
He is as evil
as midnight.
He and his band
search for Hen
Men for only she
knows the one
thing that can
destroy him.
BY DANIEL MOORE

International Council for Computers in Education



Publications

ICCE



About ICCE

The International Council for Computers in Education was founded by Dr. David Moursund in 1979 as an organization that would foster appropriate instructional use of computers throughout the world.

Today ICCE is the largest professional organization for computer educators at the precollege level. It is nonprofit, supported by 12,000 individual members and more than 50 organizations of computer-using educators worldwide. These organizations are statewide or regionwide in scope, averaging 500 members each. Approximately 84% of ICCE's individual membership is in the United States, 12% is in Canada, and the remainder is scattered around the globe.

About The Computing Teacher

ICCE publishes *The Computing Teacher* journal. *The Computing Teacher* provides accurate, responsible, and innovative information for educators, administrators, computer coordinators, and teacher educators. The journal addresses both beginning and advanced computer educators through feature articles, columns, software reviews, and new product and conference listings. Contributors to *The Computing Teacher* are leaders in their fields, consistently providing the latest information in computer education and applications.

Publications, Special Interest Groups

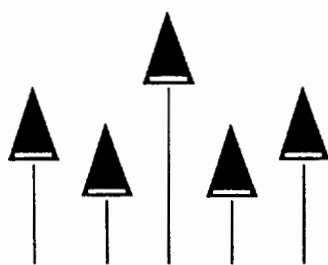
In addition to *The Computing Teacher*, ICCE provides a number of publications to computer-using educators. ICCE's Special Interest Groups provide in-depth information for computer coordinators, teacher educators, computer science educators, and Logo-using educators. *C.A.L.L. Digest* is published nine times per year for ESL teachers. ICCE committees address a variety of ethical and practical issues important to the computer-educating community.

Independent Study Courses

ICCE offers graduate-level independent study courses, designed to provide staff development and leadership. These courses have been approved by the College of Education at the University of Oregon and carry graduate credit from the Oregon State System of Higher Education. Participants correspond with instructors by mail.

Write for information and a free catalog today!





LOGO EXCHANGE

Volume 7 Number 9

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

May 1989

Founding Editor

Tom Lough

Editor-In-Chief

Sharon Burrowes Yoder

International Editor

Dennis Harper

International Field Editors

Jeff Richardson

Jun-ichi Yamanishi

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 Wetzal, 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 — But I Can't Write!

Sharon Burrowes Yoder

2

Monthly Musings — Turtle Wordle World

Tom Lough

3

Logo Ideas — Hypertext in Logo Writer

Eadie Adamson

4

Logo II Tools for ProDOS — Disk Management

Michael Muir

7

Little Kids and Logo — Mud, Marvelous Mud!

Leslie Thyberg

8

Logo Quilts: A Different Approach

Judy Zeiter

10

LogoLinX — Metaphysical LinX

Judi Harris

12

Logo Connections — Say WHEN!

Glen Bull and Gina Bull

17

MathWorlds — The Council for Logo in Mathematics Education...an Update

by Ihor Charischak

22

Assessing Logo Learning in Classrooms — Managing Workspace and Files

Dan Watt

24

Search and Research — Representing and Monitoring

Douglas H. Clements

28

Global News

Dennis Harper

31

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 Talbot Bielefeldt.

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

But I Can't Write!

As I sit here in Eugene writing this editorial, the sun is shining (yes, even in Oregon!) and the daffodils are blooming. At the coast, tiny wildflowers are appearing in our driveway, heralding the arrival of spring. It is a time of new beginnings. And yet, this year of *LX* is drawing to a close, much as the end of your school year is approaching as you read this. The last issue of *LX* for the year brings with it the same bittersweet feelings that come with the end of a school year. You are glad for the break, for the freedom from pressures of day to day responsibilities, but at the same time you'll miss your students and colleagues. For me, it will be good to not have a new deadline every week as the production of *LX* marches relentlessly onward. But I will miss the day to day interaction with columnists and authors as we prepare for each new issue.

But there is really very little time for me to reflect — plans need to be made for next year. Not long after you read this editorial, the September deadlines will be upon me and the responsibility for filling the pages of *LX* will again be an ongoing part of my day to day life. So now, as I bring the publication year to a close, I again need to turn to you for help so that I can bring a high quality, useful journal to you month after month. As before, I want to encourage you to contribute to *LX*. There are four areas where you can help.

1. **Send cover art.** Believe it or not, I never seem to know from one month to the next whether I will have student work for the cover. Right now my files are empty! I know many of your students have produced delightful products that others would love to see. All I need from you is a dark, printed copy, the name or names of the students who produced the submission, and a brief paragraph describing the project/assignment that led to the creation of the material that you enclose.
2. **Write a letter.** Do you have a question? How about a comment on some article or column in *LX*? Share your thoughts. If you would like your voice heard, I'll be glad to publish any letter of reasonable length.
3. **Send some local news.** What is your school doing with Logo? What interesting Logo projects are going on in your area? Are you a member of a local Logo SIG? What are they doing?
4. **Write an article.** We need to hear from practicing teachers. What are you and your colleagues doing with Logo? What interesting projects did you do in your

classroom this year? Do you have a new twist on an old idea? How about ways to integrate Logo into different areas of the curriculum? Tell us about the progress of a particular student or group of students.

From many of you I can almost hear "But no one would be interested in what I am doing. Besides, I can't write like those people who write for *LX*." Don't stop now — keep reading.

Take a close look at this issue of *LX*. It has quite a large variety of material: letters to the editor, reports, announcements, articles by teachers, and columns. I'd like next year's *LX* to have variety such as this in each issue. This will only happen if you share what you are doing. Certainly you have had experiences similar to those in this issue of *LX* — let me hear about them. When I ask what changes people would like to see in *LX*, the most common suggestion is to include more practical ideas that can be used in the classroom.

But, you repeat, "I can't write!" Not to worry. That's part of my job. You send me your ideas and I'll work with you to produce a publishable product. Very seldom does an article you read in *LX* look exactly like it did when it came to me — just ask my columnists! Even the most experienced professional writers are subject to editorial intervention.

So, when school is out and you are cleaning out your desks and cupboards, take time to collect some student work to send to me. As you plan your summer vacation, set aside some time to share with me your successes and failures. Thrill your students by having their work published or by writing an article about them.

Drop me a note or call ICCE and we'll send you guidelines for submission of articles and covers. In my March editorial I included publication deadlines and a description of our theme issues for next year: Just for Beginners, Logo Connections, and Extra for Experts. Remember: while you are enjoying the summer sun, I'll be looking for material for the September issue.

Enjoy your summer. Take time for rest and relaxation and for professional renewal. I'm certainly going to take time to walk on the beach...and I may yet find an agate!

Sharon Burrowes Yoder
SIGLogo/ICCE

1787 Agate Street, Eugene, Oregon 97405
CIS: 73007,1645 BITNET: YODER@OREGON

Monthly Musing

Turtle Wordle World by Tom Lough

Because of Logo's close relationship to LISP, a widely used artificial intelligence language based on LIST Processing procedures, the list is often billed as the fundamental element for Logo. However, there is another Logo element which should also receive a share of attention: the word.

The other day, I looked at the phrase "word processing" as if I were seeing it for the first time. I had been thinking about list processing and the various operations using Logo primitive procedures such as FIRST, BUTFIRST, and the like. So when I saw "word processing," I didn't think about the familiar use of the word processor, as I usually did; instead, I thought about applying FIRST and BUTFIRST to words. I was amused that I had not made this connection before. Naturally, this caused me to begin thinking more about words.

Many teachers have become aware of the wonderful potential of Logo to promote word play of one kind or another. For example, in the February 1985 issue of the *National Logo Exchange* newsletter, Cathy Frank delighted NLX readers with her feature article, "Wordles and Turtles." [For LX readers not having access to back issues, her article is reprinted on page 331 of *Logo: Theory and Practice* by Dennis Harper, Brooks/Cole Publishing Co., 1989.] Cathy suggests, "A wordle is an incomplete word or group of words whose full meaning is implied by the way the words or letters are written." Here is a sample wordle procedure. Can you figure it out?

```
TO WORDLE.SAMPLE
PRINT [STAND]
PRINT [I DO NOT]
END
```

Glen Bull and Paula Cochran created wonderfully long Hawaiian words using their word tools in the September 1985 NLX (also reprinted in the Harper book, page 244), and suggested some interesting ideas about language and words in general.

I have selected these two word examples because they contain elements of a project I would like to suggest to you this month. Cathy's "wordle" reminds us of "turtle." Glen and Paula's project suggests words in other languages. Together,

they form the basis of an international Logo project Dennis Harper and I would like to sponsor with your help.

Dennis and I are collecting the word "turtle" in as many different languages as we can find! Wouldn't it be interesting to have a special Turtle Dictionary, containing the word "turtle" in Spanish and Cambodian and Swahili and hundreds of other languages?

Once the collection is complete, then the words can be used for many other projects, such as making a turtle design out of the words for a t-shirt or other object. Maybe a music class in an elementary school could write an international song using as many of the words as possible. As an introduction to statistics, students could explore how often various letters of the alphabet are used throughout the collection. Teachers could use the word collection as a bridge to geography lessons, as a facilitator for international pen pal projects, or for any number of other activities.

Would you be willing to help? If you know the word "turtle" in a language other than English (or can find out from someone who does), please send it to us, along with the name of the language. If the language is based on letters or symbols other than those of the English alphabet, please enclose an approximate transliteration of the word; also send the word written in its native language, if possible. If the language is one spoken only in one particular part of the world, please include any details you might know.

Perhaps your students might like to get involved in the collection effort. They could check the foreign language dictionaries in their school library, interview relatives and visitors from foreign countries, or write to their pen pals overseas; what an interesting research activity this could be!

We are excited about the potential of this project, and look forward to hearing from Logo Exchange readers around the world. Thank you in advance for your participation.

FD 100!

Tom Lough
PO Box 394
Simsbury, CT 06070

PS: The answer to Cathy's wordle is "I do not understand."

Logo Ideas

Hypertext in LogoWriter by Eadie Adamson

HyperCard, a much published implementation of the ideas of hypertext, is the cause of the excitement behind articles in computer magazines and the popular press. However, HyperCard only runs on a Macintosh computer. For many schools a Macintosh is only a dream. You many have seen demonstrations or read articles about using HyperCard to run CD-ROM and videodisc players. Such sophisticated equipment is even further beyond the reach of most school systems. The idea of hypertext need not remain a dream. There are ways in which you too can show your students a little about hypertext using LogoWriter.

A Bit of History

The idea of hypertext is not new. In 1945 in an Atlantic Monthly article, "As We May Think,"¹ Vannevar Bush envisioned a system he called "memex," in which pieces of text, filmstrips, pictures and other information could be organized by the reader into a preferred order. Bush didn't call his idea hypertext, but memex was the beginnings of the idea. (The term hypertext was first used by Ted Nelson in the 1960's.) Bush even spoke of buttons and pointers, envisioning a kind of workstation which would contain all this information on microfilm and which could be organized according to need and called up at will. It was one idea for solving the problem of proliferation of information, making it possible for an individual to organize things in a useful fashion.

Several years ago Mark Guzdial sent me a copy of REFTOOL3, a Logo hypertext application he had been developing for the Macintosh, Apple II and IBM Logos. Mark experimented with a history project, an adventure game, a short story, and even used REFTOOL to write his documentation. His reference manual is an excellent resource for anyone interested in looking into the background of hypertext systems.

A recent novel, *The Dictionary of the Khazars* by Milorad Pavic², uses an approach to storytelling which is remarkably close to the idea of hypertext. There are three sections to the novel; it is arranged like three small encyclopedias. Within each entry, words are flagged with special symbols indicating a reference to that word can be found either in the section you are reading, in one of the other two sections, or in all three sections. Reading the book becomes a very individual experience. After a few excursions you run out of fingers or bookmarks to mark the place just left. You simply invent your own continuity as you read along.

Within the last year or so a number of us in the Logo community have explored the idea of hypertext using LogoWriter. In fact, it was not until I began working on a version of hypertext with LogoWriter that I became aware of Andy David's work on a similar tool, far more sophisticated than mine. Andy's tool was shown at the Great Lakes/East Coast Logo Conference in Cleveland in May of 1988. Michael Tempel and I have recently shared our differing versions and design outlooks on hypertext with LogoWriter at workshops in New York and at the West Coast Logo Conference in Los Angeles.

Why LogoWriter Hypertext?

A simple version of hypertext seems to capture the imagination readily. What I hoped to do with LogoWriter hypertext was just that: give my students a tool to play with, to ask them to "think in footnotes" and to link those footnotes to their main text so that the reader could follow their thinking. In particular, I wanted my students to be able to experience this different way of thinking about text. I was also curious about their response. LogoWriter seemed a good vehicle.

Hypertext has a great deal in common with the LogoWriter adventure stories, time lines, and simple data bases, all of which are familiar to seasoned LogoWriter users. LogoWriter's page metaphor puts us quite naturally into this manner of thinking about the pages (files) stored on our disks. One LogoWriter activity card even encourages students to make a slide show of the pages on their disk, metaphorically asking them to use LogoWriter in the way Bush envisioned his system: determine the order in which your pages are to be read or viewed; write a procedure which displays the pages in that order.

Students planning an adventure story or time line with LogoWriter often set up a plan or diagram from which to begin. If we think of the story path or paths as a trail, the analogies to hypertext systems are quite apparent.

Last summer at the Logo Institute, Elise Brabeck and I joined forces and began working on a version of hypertext in LogoWriter. I had not fully explored all LogoWriter's text processing commands. A hypertext project seemed just the thing. Elise was a wonderful partner and a genius at problem solving; a truly indispensable ally in this project.

SELECT Is the Key

Understanding how SELECT works is the key to developing a LogoWriter hypertext tool. The hypertext tool needs to activate the selection of text through some kind of pro-

grammed control. LogoWriter has words which function in the same manner as pressing the Select, Cut, Copy and Paste keys. They are, interestingly enough: SELECT, CUT, COPY and PASTE. For the purposes of hypertext we are most interested in the word SELECT, which will turn on the highlighting. As a simple example, put a few lines of text on the page and try typing this in the Command Center:

```
TOP
SELECT
```

Notice that not much has happened, other than that the cursor moves to the top left corner of the page and then disappears.

Now type:

```
BOTTOM
```

All your text is highlighted, or SELECTED. The word SELECTED is important for our investigations. Now type:

```
SHOW SELECTED
```

You should see all the highlighted text appear in the Command Center. To undo the highlighting, type:

```
UNSELECT
```

In order to investigate how SELECT works, the cursor must move one character at a time and we must check what has been selected as it moves. Try putting the cursor in the middle of a word. Type:

```
SELECT
SHOW SELECTED
```

You will see that nothing has been selected so far. Now move the cursor using another command which may be new to you, CF (cursor forward):

```
CF
SHOW SELECTED
```

One letter should be returned to you in the Command Center. Play around with this idea for a while, moving the cursor forward, CF, or back, CB, and use the command SHOW SELECTED to get information on what is happening. Watch the cursor disappear when you back up or move forward to the place in the text where you began SELECT.

This selection process is the basis for developing a hypertext program in LogoWriter. There's a little more to it than that, since you must take account of extraneous characters, such as asterisks, marks of punctuation and so on. We'll see that it is helpful to make a list of them and to test for them as a word is selected.

How Will It Be Operated?

The second part of a hypertext system involves the user interface. How could we imitate a mouse with LogoWriter? I chose the idea of using a READ program, allowing the reader to move the cursor through the text under program control. Arrow keys changed direction and pressing return selected a word. Pressing a space bar caused the program to look for a page matching the selected text, and to go to it. Otherwise the user is informed (in the Command Center) that there was no reference page for the text. Adjustment could be made to run procedures instead of merely selecting a page. Later, fresh from my reading of *Dictionary of the Khazars*, I added a place-keeper which returned the cursor to the place in the main text from which the program had moved to select another page.

Further Revisions

This version of hypertext doesn't keep a trail yet; that's a project for this summer. It gives students the germ of the idea, however. I spent some time polishing it, adding such touches as a release key which unselects a word chosen by mistake and making simple instructions so that a student could easily incorporate the tool into his work. I have now used it for projects with several groups of students this year.

How Hypertext Is Used

A hypertext program needs to be loaded as tools so that the user can move freely from page to page. Since tools are loaded into memory there is no need to put the procedures on each page. A STARTUP procedure loads the tools. (A STARTUP procedure on any LogoWriter page runs as soon as the page is accessed.) I suggest for my students that they create a STARTUP page containing a title page for the reader to look at while the tools are loaded. (A STARTUP page is a page that is automatically loaded when LogoWriter starts.) The STARTUP procedure on this STARTUP page not only loads the tools into memory but starts the read program, puts the instructions for using the tools in the Command Center, and moves to the first page of text when the user presses a key:

```
TO STARTUP
GETTOOLS "HYPERTOOL  Waits for the reader
NAME READCHAR "KEY   <— to press a key.
```

Logo Ideas--continued

```
GETPAGE "first.page.of.text
INFO      <- Puts instructions in Command Center.
READ      <- This activates the cursor so the reader
END        can move through the text.
```

Signaling a Reference Word

We decided to capitalize words for which there was a reference. Since I added a "There is no further information on that now" statement when a page was not found, the program does not stop if an unreferenced word is selected. Making the reference words uppercase simply saves the reader considerable frustration, especially if a story is long.

While Using Hypertext, Procedures Run Too

Using a READ program such as the one I developed for this version of hypertext does not preclude running procedures. Although the system is keyed to getting pages corresponding to the selected word, any page chosen can have its own startup which runs any number of procedures before returning control to the reader. READ does not need to be invoked again, so animation can be part of the fun, too!

Hypertext Projects

The very idea of hypertext sparks student interest, I've found. Some of my students are planning to follow up data base projects by writing hypertext interpretations and expansions of their research findings for the data bases. Instead of asking students to write a traditional, sequential story, ask them to try thinking in hypertext. Then show them how to apply this idea and see what they dream up. A travelogue or a geography project could also be created. History projects of all kinds might be fun in hypertext. An English teacher would probably enjoy using hypertext as an exercise in concise writing. Students could write a long descriptive piece first, then create a hypertext document by using CUT to remove long descriptions and PASTE these on pages of their own. Hypertext might also be useful for vocabulary in any language.

Two Versions of Hypertext Available to You

Michael Tempel and I are building a collection of LogoWriter hypertext projects. We have prepared a LogoWriter 2.0 scrapbook disk which contains, on one side, Michael's version of hypertext together with sample projects, and on the other side, my version with a hypertext history in hypertext plus some sample projects using this version.

If you send us at least one idea for using hypertext, along with a blank disk (specify IBM or Apple 2.0 or 1.1), mailer, and return postage, we will send you a copy of this scrapbook disk. Send your idea and disk to:

Michael Tempel
Logo Computer Systems, Inc.
330 West 58th Street - Suite 5D
New York, N.Y. 10019

Michael and I are collecting sample LogoWriter hypertext projects. If you use either version of hypertext, please send us samples to add to the collection so that we may share them with others.

References

1. Bush, Vannevar (1945). "As We May Think." *Atlantic Monthly* July 1945. Reproduced in CD Rom: The New Papyrus (1986). Lambert, S. and Ropiequet, S., eds. Redmond, WA: Microsoft Press (pp. 3 - 20).
2. Pavic, Milorad (1988). *The Dictionary of the Khazars*. New York: Alfred A. Knopf.
3. Guzdial, Mark (1986). "Wandering in a Sea of Text: REFTOOL, A Logo Hypertext" Paper and Presentation at Logo 86: The Third International Logo Conference, MIT.

Eadie Adamson is Computer Coordinator at
The Allen-Stevenson School,
an independent school for boys, K - 9,
in New York City.

Her summer mailing address is
1199 Park Avenue, Apt. 3A, New York, N.Y. 10128.

Summer course opportunities!

Bellevue Public Schools in Bellevue, Washington (near Seattle) is offering courses in LogoWriter, Intermediate Logo, and Lego/Logo. For more information contact:

John Newsom
P. O. Box 90010
Bellevue, WA 98009-9010
206-455-6200

Logo II Tools for ProDOS

Disk Management

by Michael Muir

Do you have problems remembering what you named your Apple Logo II or LCSII Logo II data disks? Are you frustrated with ProDOS pathnames? Does SETPREFIX just seem to keep you from working on your programs? Did you not find out until too late that it is a lot easier to name your data disk /LOGO? Do you get frustrated having to erase files before you can save them again? Then the short procedures in this article can help you.

UPDATE

The first procedure is UPDATE and is used to update a file which you have been working on. It takes the name of a file as input and if the file already exists erases the old version before saving the new version. In many versions of Logo, you cannot SAVE a file if it already exists on your data disk.

```
TO UPDATE :FILENAME
  IF FILEP :FILENAME
    [ERASEFILE :FILENAME]
  SAVE :FILENAME
END
```

Say you have been working on procedures which are currently saved in a file called TOOLS and are now ready to save your changes. Normally you would have to use ERASEFILE to remove the old TOOLS file from your disk and then use the command SAVE "TOOLS. Using the above procedure, you need only type UPDATE "TOOLS and the procedure UPDATE takes care of the rest. Since TOOLS is a file already on the disk, UPDATE erases it before saving the revised version.

Using SETPREFIX

The next two procedures use the primitive SETPREFIX. SETPREFIX is a primitive which, among other things, lets you tell the computer the name of the disk on which you want to save your programs. If you switch disks a fair amount or if you forget the name of your data disk, saving files can become very frustrating. One solution is to name all your data disks /LOGO. Another solution is to use the procedure called SETDISK. It works with the help of the operation (reporter) ONLINE.

ONLINE outputs the names of the disks which are currently in the disk drives. It behaves a little strangely, however. If there is a disk only in the first drive, ONLINE returns a list containing the name of that disk. If there are disks

in both drives, ONLINE returns a list of the names of both disks, but the names are reversed. The name of the disk in Drive 2 is listed first and the name of the disk in Drive 1 is listed second.

SETDISK

SETDISK sets the ProDOS prefix to the volume name of the disk in Drive 1. Although SETDISK can be used when you have disks in both disk drives, strange things may happen if both disks have the same name. The computer cannot determine which disk by the given name is to be used for saving. SETDISK works best when you are just using one drive and don't want to be troubled with remembering the name of each disk. Don't forget to use SETDISK each time you swap disks.

```
TO SETDISK
  IF ( COUNT ONLINE ) > 1
    [SETPREFIX FIRST BUTFIRST ONLINE]
    [SETPREFIX FIRST ONLINE]
  END
```

SETDRIVE

The procedure is easy to use. To change to a disk with a different name, or with a name you cannot remember, just put the disk in Drive 1 and type SETDISK. SETDISK finds the name of that disk and changes the prefix to that name.

The next procedure allows you to determine on which drive to save your programs. SETDRIVE takes the numbers 1 or 2 as input. First, the procedure finds out the names of the disks in the the drives. If there are two disks and both disks have the same name, SETDRIVE prints a message to that effect and takes no further action. As was stated above, if two disks have the same name, the computer is unsure on which to save. If the names are different and 1 is the input, the computer sets the prefix to the name of the disk in Drive 1. If the input is 2, SETDRIVE first makes sure that there are two disks on line. If there are not, an error message is printed. If there are two disks, the prefix becomes the name of the second disk. The assumption is made that if you are using only one disk, it is in Drive 1. Note that SETDRIVE 1 is functionally the same as SETDISK.

```
TO SETDRIVE :DRIVE
  MAKE "VOLS ONLINE
  IF (COUNT :VOLS) = 2
    [IF (FIRST :VOLS) = (LAST :VOLS)
      [(PRINT [BOTH PREFIXES ARE
        CALLED] FIRST :VOLS)
        STOP]]
```

Logo II Tools for ProDOS--continued

```

IF :DRIVE = 1
  [IF (COUNT :VOLS) > 1
    [SETPREFIX FIRST BUTFIRST
      :VOLS]
    [SETPREFIX FIRST :VOLS]]
IF :DRIVE = 2
  [IF (COUNT :VOLS) < 2
    [PRINT [CHECK DISK DRIVE 2]]
    [SETPREFIX FIRST :VOLS]]
END

```

If you are a more experienced programmer, you may wish to include these procedures in a STARTUP file and then BURY them. This way you can use them as if they were Logo primitives. Hopefully, these procedures will make your disk management a little easier and you can spend your time focusing on your own program writing.

Michael R. Muir
RR 1, Box 3359
Burnham, ME 04922

Attend the ICCE Summer 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 - 22, 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 the
 Leadership and Logo for Leaders Workshops!**

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

Little Kids and Logo

Mud, Marvelous Mud!

by Leslie F. Thyberg

There is an age old nursery rhyme that goes "April showers bring May flowers." In Pittsburgh, April showers also tend to bring mud — lots of it. Like many members of the baby boomer generation, I am now the parent of a preschooler. My husband and I are alternating weeks attending and participating in a parent/child art class at the Carnegie Museum. Last week was my first turn with our son, David. This particular class was devoted to an art experience based on the traditional desert architecture from West Africa and Southwest Asia. Translation? Mud houses! The school teacher in me was fascinated with the variety of activities being made available for the three and four year olds, all revolving around the theme of mud. The Logo lover in me, of course, immediately turned the experience in to a Logo activity! What has mud got to go with Logo? Plenty!

Several years ago, Seymour Papert wrote a wonderful, landmark article called "The Computer as Mud Pie" (Papert, 1984). Learning playing and playing learning are the avenues for the construction of the mind. True learning takes place only when children experience personal interactions with knowledge. Tactile play and exploration are the means that children have for reflecting and making meaning of their worlds. Preschools today universally utilize play dough, and sand and water play as a part of children's exploratory activities. Why not extend this notion for exploration and construction to the computer--and what better avenue than Logo?!

Mud is a simple technology. The Art Express class in which my son and I were participating went on a field trip to the hall of architecture at the Carnegie Museum of Natural Science and History. There we viewed a photo collection of a wide variety of mud buildings. I was impressed with what I saw, foremost because of the potential for teaching children to learn respect for beauty in simplicity. The photographs were of gentle buildings — expressive, ecologically sound and wonderfully made. Mud is not a backward building material at all. In fact, it is a great asset. Its resistance to temperature changes against the heat of the day and the cold of the night make it a far better material than many modern day materials.

The instructor sent each parent/child team on a treasure hunt of sorts. Our mission was to search for specific shapes in the buildings we were viewing: semi-circles, triangles, arcs, rectangles, squares and circles. We had to find a building that looked like a sand castle starting to wash away. We were to search for a roof that had one big point and another roof with

a rounded top like a hill. We were to study walls and find some with lots of holes cut out and others with lots of vertically directed rectangles in the front. My son's favorite was the building with spines like a porcupine. Many of the mud houses have sticks bristling from their walls which serve as ladders for ease of maintenance. We saw thatched roofs, ribless valuts and domes made without the benefit of scaffolding. The focused engagement of parent and child alike was delightful. Squeals of "Look what I found!" and "Look what I see!" rang throughout the hall. The abacus-like colonnades shaped like donuts and bars received many an "ooh" and "ah." They too were my favorite because they teased my visual perceptions of weight with their applied column supports. (A good Logo puzzler.)

Needless to say, my three year old did not come away with the same knowledge acquisition or experience as did I. He is thankfully much more right-brained and experientially oriented than I. Upon completion of our field trip, we returned to the children's art room at the museum and proceeded to spend the remainder of the morning with our sleeves rolled up and our hands busily shaping and pounding mud into edifices of our own design. We were given interesting tools with which to construct and design our clay buildings. Square and round dowel rod pieces were provided, along with wooden molding to make curlicue designs. Colorful flags were given to us to affix to the top of our buildings. (In the Muslim culture these flags were flown at the spring equinox, the Afghan New Year, or to commemorate special occasions, or to mark miracles. This seemed especially fitting to me — the spring equinox and mud do go together!) What was particularly enjoyable for me in the midst of all this, was that parents and children were all busy making their own creations. When we placed them all together as a group, we had a magnificent village that looked surprisingly like a miniature of the photo collage that we had just witnessed. No two mud houses were the same.

When our class was completed, we wandered back out of the museum through another intriguing exhibit — the Polar Hall. I immediately stopped in front of the life-size igloo to study it more closely. Its beehive-like structure was equally as gentle and utilitarian as its mud-house counterpart. We watched a brief movie on how to build an igloo. We discovered that "igloo" is the Inuit word for any kind of house. A snowhouse is an "igloovigak." There are natives to western Alaska who actually build sod houses. More mud! The igloo is remarkably similar to the West African mud house. Like the mud house, igloo entrances vary from flat roofed tunnels to series of domes and combinations of tunnels and domes. They are both practical and environmentally worthy shelters that

take advantage of nature without abusing it. A mud house is usually constructed from molded bricks that have been baked by the sun. As the walls go up, bricks are joined together by mud for mortar. The Inuit actually stamp out a near perfect circle with their heels. They cut some forty to fifty blocks of snow to set around the baseline of their shelter giving each a slight inward tilt. There is a fascinating math and movement dynamic to its construction. A right-handed Inuit constructs his igloo in a counterclockwise direction. A left-handed Inuit constructs his dwelling moving in a clockwise direction. Why? They shave off a half dozen squares or so on the baseline to form a ramp that creates an upward winding spiral. They build their house from the inside, tilting each square until the final block is nested in on top. They, too, chink their cracks to make the structure weatherproof. Like the mud hut, the igloo is strong enough to bear a person's weight. Can you tell that I was fascinated by the similarities?

Are you still asking, what does this have to do with Logo? Plenty! Why not send your students on a shape hunt? I've had innumerable students over the years comment that they never really noticed the patterns in their bathroom tiles so well as when they began studying Logo! What a wonderful way to use Logo in your social studies classes. Field trips are terrific, but not necessary. Our exposure to mud houses was in the form of photographs. National Geographic is always a good classroom resource. Have an architect come in and speak to your class. Turn your social studies groups into city planners. Every primary level classroom studies some variant of "My Neighborhood" or "My City." Some children can be the urban developers for "Circleburgh." Others may design "Rectangle Row." Clay or play dough models can be built. Use Legos to build a model (it can even be built to scale) before putting the plans on the computer! The construction of a simple house or of a complex community both offer the opportunity of introducing or further developing the concepts of super- and sub-procedures. To streamline the "construction" activities, you can build a shape tool kit for them to use (or get some older students from the upper grades to do this as their Logo project). If part of your goal is vocational awareness, assign children in your groups different roles to perform (foreman, carpenter, general contractor). You can play the part of the chief building inspector! The creative dramatics or dress-up corner can have a wide assortment of construction tools, toys, and hats placed in it. Or, transfer it to a different culture. If it is the Muslim culture, assign the girls as the artists and the boys as the builders. Sound sexist? Only the women do the decorating and embossing of the structures in Muslim portions of West Africa and Southwest Asia. Their reliefs include stars, waves, grids, and rosettes. Peacocks are often added as a symbol of good luck. Ostrich eggs are used as symbols of

Little Kids and Logo--continued

fertility and prosperity. Think of what an interesting Logo tool kit could be devised for such a unit of study! Still another possibility is to focus the study on the pueblo dwellers of the American Southwest. Why? In the Eighth Century, the Arabs introduced their intricate mud-building techniques to Spain. The Spanish, in turn, brought these skills to the American Southwest. I was amazed to discover that the Pueblos and adobe dwellings indigenous to the Southwest have their roots in Arabic culture! It's a small world. Similarities and differences--an abstract but significant understanding for children to develop. "Art-chitecture" can serve as a fascinating vehicle to build values--aesthetic and ethical--for the future citizens of our global village. What better way to begin than with the computer as mudpie?

References

Papert, Seymour. (1984, January). Computer as mudpie. *Classroom Computer Learning*, 37-40.

Dr. Leslie F. Thyberg
5637 Rippey Street
Pittsburgh, PA 15206
AppleLink ALS 038

SIGLogo Meeting

Don't miss the annual SIGLogo meeting at NECC. This is your opportunity to make your views known, get to know your new officers, and share with other Logo enthusiasts.

Day: Tuesday, June 20

Time: 7 - 9 P.M. (Birds of a Feather time)

Place: Check bulletin boards for the room

Note: If you have not yet returned your ballot for the new officers, please do so immediately!

Logo Quilts

A Different Approach:

by Judy Zeiter

Quilt patterns have been used in a variety of ways in teaching Logo as well as other classroom activities. However, at the Greenhill School of Dallas, Texas, we think we have a new and exciting approach to using quilt activities.

One of the special features of the Greenhill School is the focus on the integration of subject areas. I began teaching computers at Greenhill in 1987, as did our present art teacher. Our preparations for the Pioneer Week in fourth grade brought about the development of a project in which students designed quilts in their Logo computer classes and then actually quilted their designs in art classes.

While reading an art magazine, our art teacher noticed an article about computer generated designs. She had begun a traditional quilt-making project with our fourth graders, who were studying about pioneers. She showed the article to me and asked if it was anything that I could use to help integrate my classes with the pioneer unit. I was teaching the concept of variables, and saw the opportunity to illustrate this concept by having the students make quilt designs in Logo. The designs that the students made were beautiful.

After comparing the Logo designs created in computer class to the quilt designs made in art class, the art teacher and I decided that next year we would combine our students' talents to produce an integrated project to be shown during Pioneer Week.

The integrated project began in Logo class. The students wrote procedures to make simple pattern block shapes: the square, triangle, and kite. The procedures used variables so that these shapes could quickly be made in any size. The square and triangle could be drawn in either direction. Many other shapes could be used. Our only limitation was time. Some of the students did combine shapes previously made in other projects with the shapes made specifically for the quilts.

The following procedures were used to make the quilts.

```
TO SQR :S
REPEAT 4 [FORWARD :S RIGHT 90]
END
```

```
TO SQL :S
REPEAT 4 [FORWARD :S LEFT 90]
END
```



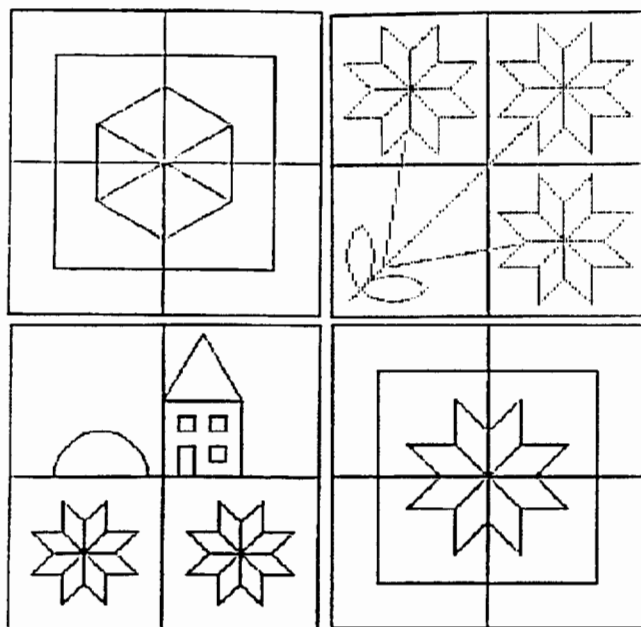
```
TO TRIR :S
REPEAT 3 [FORWARD :S RIGHT 120]
END
```

```
TO TRIL :S
REPEAT 3 [FORWARD :S LEFT 120]
END
```

```
TO KITE :S
REPEAT 2 [FORWARD :S RIGHT 45
FORWARD :S RIGHT 135]
END
```

Some picture examples of quilt patterns were placed in the computer room to give students ideas of what quilts look like. From this point on, they used their creativity, with technical advice from the computer teacher, to complete their computer designed quilt patterns.

These computer generated quilt patterns were taken to art class and colored with fabric crayons. The colored designs were transferred to fabric by ironing. In the tradition of the pioneer days, the students then had a quilting bee in art class, sitting with needle and thread in hand and quilting their designs.



Before beginning the project, the computer and art teachers met and decided on specifying the following design requirements:

1. To meet size requirements for the fabric used in art, the students were to make either one square 200 steps on a side or four rotated squares 100 steps on a side to form the base for their designs. This procedure was then entered in the Logo editor.
2. The quilt pattern had to fit within the boundaries of the base design.
3. The students had to be careful not to overlap the designs, creating small areas that would not clearly transfer when using the fabric crayons.
4. The students worked in pairs on the computers. We did not want repetitious designs. Therefore, after a student and his/her partner completed one design, they were required to make minor changes in their original quilt design to create a new design for the partner. For example, some simple changes could be degree rotations of the design, or the use of different numbers for their variable shapes.

Overall, the project was a great success. The students could see that the use of variables made their programming much easier. They had a definite purpose for computer designs, in this case, to create quilt patterns. On Pioneer Day when the students proudly displayed their work, the computer-art display captured everyone's attention.

The quilt pattern project gave the students an opportunity to combine several areas of study. The integrated unit allowed the students to extend and enrich the learning experience and strengthen the relationship of academic, fine arts, and scientific classes.

Judy Zeiter
Greenhill School
14255 Midway Road
Dallas, TX 75244

Logo LinX

Metaphysical LinX

by Judi Harris

In the beginning, there was Consciousness and only Consciousness. Consciousness knew that microworlds provide excellent learning opportunities. It decided, therefore, to create an ILLUSION microworld. Consciousness created four microworld primitives: PERCEIVE, MANIFEST, COMMUNICATE and EMOTE. The procedures looked something like this:

```
TO PERCEIVE :FORM
RUN [KNOWLEDGE]
END
```

```
TO MANIFEST :FORM
RUN [[PHYSICAL FORM]]
END
```

```
TO COMMUNICATE :FORM
RUN [INTERACTION]
END
```

```
TO EMOTE :FORM
RUN [FEELING]
END
```

Now that the five tool procedures were written, Consciousness created a key with which to invoke any of the procedures. It was the **LOVE** key.

But Consciousness wasn't satisfied. It had created the basic expression of the four-square physical universe (aren't you glad that it didn't call it TO SQUARE?), but the superprocedure from which all these are called had yet to be coded.

The ultimate goal, Consciousness reasoned, is awareness. The only true path to awareness' door, it intuited, is realization. So it coded:

```
TO REALIZE :TRUTH
TEST PERCEIVE :TRUTH DEFINEDP
[KNOWLEDGE]
IFTRUE [OUTPUT [CREATIVE OUTLET]]
TEST MANIFEST :TRUTH DEFINEDP
[[PHYSICAL FORM]]
IFTRUE [OUTPUT "ABUNDANCE"]
TEST COMMUNICATE :TRUTH DEFINEDP
[INTERACTION]
IFTRUE [OUTPUT "RELATIONSHIPS"]
```

```
TEST EMOTE :TRUTH DEFINEDP [FEELING]
IFTRUE [OUTPUT "HEALTH"]
END
```

Still, Consciousness sensed that something was wrong. When it commanded,

```
REALIZE :TRUTH LOVE
```

a single expression of All That Is would be produced...and then the illusion would disappear. Well, Consciousness knew that ILLUSION was valid for itself and certainly wasn't nothing...so it looked for another programming technique to employ.

It tried iterative structures:

```
REPEAT (RANDOM 10000000000000)
[REALIZE :TRUTH] LOVE
```

that just took longer to disappear,

It tried variable inputs:

```
REALIZE JUDAISM LOVE
REALIZE CHRISTIANITY LOVE
REALIZE ISLAM LOVE
REALIZE BUDDHISM LOVE
```

that output different expressions of the same truth, and then disappeared,

It even tried turtle graphics:

```
FORWARD :TRUTH
```

which just created another type of illusion, which also disappeared.

Consciousness looked all around itself at the illusions that were being expressed. It looked deep into the crevices of the Nautilus shell, the golden-white depths of the Lotus flower, and the reflective structures of the quartz crystal. It attempted to code their patterns. Eureka! They were, in deepest essence, the same.

After learning from its experiments, as is encouraged in a microworld, Consciousness realized that illusion is expressed in self-referential patterns, and that patterns of illusion are created recursively.

Logo LinX--continued

```

TO REALIZE :TRUTH
TEST PERCEIVE :TRUTH DEFINEDP
  [KNOWLEDGE]
IFTRUE [OUTPUT [CREATIVE OUTLET]
  STOP]
REALIZE :TRUTH + :EXPERIENCE
TEST MANIFEST :TRUTH DEFINEDP
  [[PHYSICAL FORM]]
IFTRUE [OUTPUT "ABUNDANCE STOP]
REALIZE :TRUTH + :EXPERIENCE
TEST COMMUNICATE :TRUTH DEFINEDP
  [INTERACTION]
IFTRUE [OUTPUT "RELATIONSHIPS STOP]
REALIZE :TRUTH + :EXPERIENCE
TEST EMOTE :TRUTH DEFINEDP [FEELING]
IFTRUE [OUTPUT "HEALTH STOP]
REALIZE :TRUTH + :EXPERIENCE
END

```

EXPERIENCE, you see, is a variable with a globally-defined value. Its value, of course, is inestimable, but equally so for all experiencers.

What language did Consciousness use to write its ILLUSION microworld? The language of ultimate self-realization, which doesn't really have a name, since it is an unequivocal experience; but some entities call it ENLIGHTENMENT.

Won't you please beta-test Consciousness' instructional efforts? OOPS! I guess you already are.

At the West Coast Logo Conference in Los Angeles in February of this year, Seymour Papert spoke about *instruction* versus *construction*. Logo encourages the latter. I looked down into my bookbag, and saw the copy of Dr. Willis Harman's *Global Mind Change* that I had just finished reading. It was more than a coincidence, I realized, that Harman, too, speaks of construction....but on a much larger scale.

Harman's eloquently-supported thesis is that we are in the midst of an age of pervasive change of consciousness. We are rapidly discarding notions of form being ultimate reality, and therefore creating mind, in favor of a more intuitively satisfying, albeit impossible-to-"prove" metaphysic: that Consciousness creates form. Harman cites many examples from disciplines as disparate as quantum physics, medicine, and economics to support his hypothesis.

Isn't programming in Logo a metaphorical representation of this process? When our students experiment with recursive structures, could they not be bringing some cosmically familiar creative drama into conscious awareness? Could that be one of the many reasons why Logo is so essentially satisfying and addictive?

Some LinX, dear readers, are metaphysical.

Reference

Harman, W. W. (1988). *Global mind change*. Indianapolis, Indiana: Knowledge Systems, Inc.

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

Mark Your Calendars Now!

ECCO announces The Great Lakes/East Coast Logo Conference

Where: Cleveland Airport Marriott Hotel
Cleveland, Ohio
When: May 3rd - 5th

Pre-conference workshops will be held on Thursday, May 3rd. Conference activities will begin on Thursday evening with a reception at the Marriott Hotel. The opening session will take place on Friday morning and will be followed by two days of sessions. There will be a banquet on Friday evening.

Co-chairpersons for the conference are:
Sharon Yoder, program
Glenn Looman, local arrangements
Rosemary Sutton, preconference

For conference information, contact Alice Fredman at ECCO:

1123 S.O.M. Center Road
Cleveland, OH 44124
216-461-0800

Watch the pages of *LX* for the call for presentators.

Teaching Logo Operating Commands with a Cardboard Computer

by James H. Wiebe and Malorie L. Wiebe

Although Logo is a very friendly language, certain aspects of the language are difficult to teach, whether to elementary students or to their teachers. Among these are workspace and file management, and differentiating between what is displayed on the monitor, what is in the computer's memory, and what is stored in a disk file. For example, many students think that when you clear the screen, the procedure(s) that were displayed there have been deleted and can no longer be retrieved. Others believe that when they have created a picture, the command SAVE "SCENE stores the actual picture on disk. Others, after loading a file called HOUSE from disk, are dumfounded when they get the error message, I DON'T KNOW HOW TO HOUSE after typing HOUSE to execute their routines (this error will occur if the file, HOUSE, contain no procedures of the same name).

In order to help students become successful Logo problem solvers, we recommend that memory and file management be specifically taught to students using models¹. If we move a toy turtle around a screen drawn on the blackboard or ask children move around a screen drawn on the floor to help them understand Logo graphics commands, then why not manipulate objects to demonstrate what happens when procedures are defined or saved?

When teaching students about memory and file management, a useful term to use is **space**, as in workspace. This term is appropriate because it is a general word referring to places where various forms of Logo procedures and related information can exist. In addition, the term space gives students a concrete referent that will help them visualize and manipulate these entities.

The most important space students need to learn about is **workspace**—the portion of memory containing procedures created by the user. Other spaces are **screenspace** (the screen on the computer's monitor), **diskspace** (the place where files are stored on disk), and **variablespace** (the portion of workspace where variable names and their values are stored).

Workspace

In order to help students understand systems commands like ER, PACKAGE, and POPS, and in order to help them avoid running out of memory and saving unwanted procedures to disk, it is suggested that manipulative activities with a model for the computer's memory be done soon after students have begun to write procedures. An appropriate model for the computer and its memory is the bottom part of

a cardboard box (approximately one foot square) with a keyboard and memory where procedures may be written. See Figure 1.

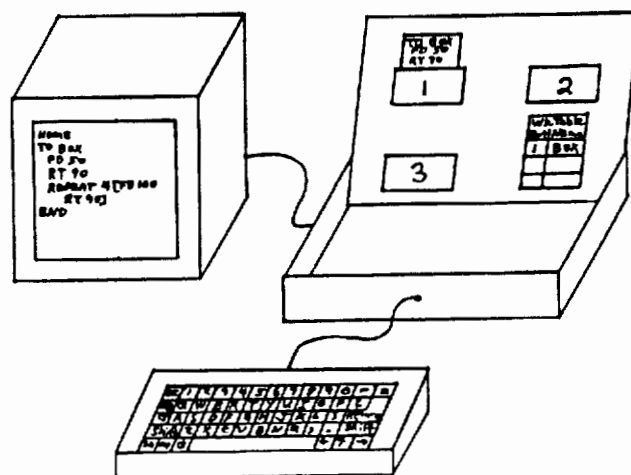


Figure 1

The memory (workspace) of the computer may be simulated with a tag board poster containing three or so slots or envelopes into which may be placed 4" by 6" cards on which short procedures have been written. Write a number on each slot so that the location of the procedures can be identified. Also included should be a region labeled "Workspace Table" for recording the names and locations of the procedures.

Activities

1. Write two or three short procedures on 4 by 6 cards and define them by placing them in the slots in the cardboard computer's memory. Record the names and locations of the procedures in the Workspace Table. You may wish to define these same procedures on a real computer at the same time. Ask your students to model on the cardboard computer what happens as you define several procedures on your real computer.
2. Show what happens when you delete a procedure from the workspace by erasing the procedure's name and location from the Workspace Table and by removing the card from the memory. Explain the command POTS (Print Out Titles) or corresponding command in your

version of Logo. Type POTS on the real computer and point out how that relates to what is written on the cardboard computer's Workspace Table.

3. Do a variety of activities with the students on the cardboard computer—adding procedures, erasing them, and asking students what will be printed when you type POTS. After students have answered each question, type POTS on the real computer to verify (or point out the error of) their answers.
4. Whenever you introduce another workspace management command, use the cardboard computer to help demonstrate what happens in the computer. ERALL, PACKAGE, and ER [PROC1 PROC2 PROC3] are examples.
5. Review workspace-management commands with the cardboard computer at regular intervals until you are sure that students thoroughly understand the concepts and commands.

Screenspace

A model for the screen is a cardboard box with paper taped on one side to model the screen and a string connecting the screen to the cardboard computer. You will not need to spend nearly as much time and effort with screenspace as with workspace; however, a model screenspace may be used to develop several important concepts. These are

1. the difference between what is displayed on the screen and what is in memory;
2. that when a procedure is displayed on the screen it is not removed from memory and placed on the screen (the screen is more like a "window" into the memory); and
3. that when a procedure is cleared from the screen, it is not deleted from memory.

Activities

1. Type PO "BOXES (or whatever command prints procedures on the screen in your version of Logo) on the real computer and discuss what happens in the computer using the cardboard model—the computer goes to the workspace table to determine where the procedure is stored, then goes to the appropriate place in memory and prints a copy of the procedure on the screen.
2. Have students demonstrate with the cardboard computer what happens when you type PO "BOXES, PO [PROC1 PROC2 PROC3], or POPS by writing what they think will be displayed on sheets of paper (this is an activity in which the entire class may participate at once).

3. Have one student tape her display to the cardboard monitor and type the appropriate command on the real computer to verify the student's answer. Discuss any discrepancies.
4. Erase or add procedures and have students model what happens in workspace and what is displayed on the screenspace when PO "(filename) is typed. For example, erase a procedure (e.g., ER "BOXES), then ask students to write on their screens what will be displayed when you type PO "BOXES).
5. Model what happens when you use commands that change the contents of the screen. For example, display a procedure on the screen (one that has already been defined) then type CLEARTEXT on the real computer and remove the paper from the cardboard screen. Point out to the class that the procedure is still in the workspace. To verify this on the real computer, type POTS or PO followed by the name of the procedure.

Diskspace

The idea of a disk file is, perhaps, even more difficult to understand than that of a procedure or workspace. Important commands and related concepts are SAVE, LOAD, ERASEFILE, and CATALOG.

To model diskspace, cut a circular disk from tag board or construction paper. Use a compass to draw several concentric circles on the tag board disk that are at least a half-inch apart. You may wish to include a square, shallow box to model the disk drive and place the cardboard disk inside when not in use. To model saving workspaces to disk, first draw two horizontal lines (or a single line in red) to mark the beginning of a new file. Next write the filename followed by each of the procedures in the workspace on one or more cardboard tracks. If a track is filled up, continue writing on the next smaller track. (See Figure 2).

Teaching Commands with a Cardboard Computer--continued

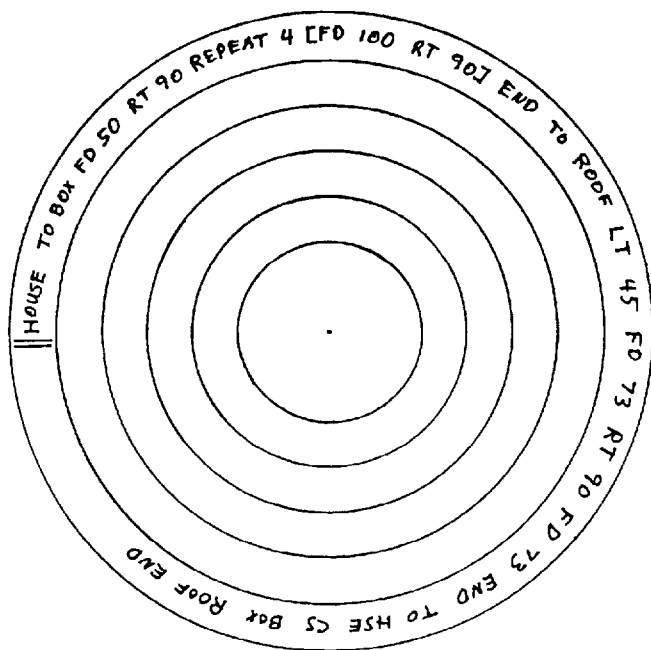


Figure 2

Activities

1. Create 2 short procedures named BOX and SUN, by placing them in the cardboard workspace. Save them to disk on the real computer and model what happens by first asking one pupil to write the file name (e.g., FILE) on the tag board disk. Next, have the pupil write the first procedure name followed by the procedure itself on the tag board disk and repeat for the second procedure. An alternative or follow-up activity is to make worksheets with circles modeling the disk and have the entire class write the file names and procedures to demonstrate what happens when workspaces are saved.
2. Create two more new procedures named RECT2, TRI2, etc., save them on the real computer, and place them in the cardboard workspace. Next, save them to disk under a new file name such as F2. Have students add the new file to their paper disks and discuss the contents of the diskette (e.g., "What files are contained on the disk?" "What is contained in the files?" "What is in the workspace?" "What would happen to the workspace if I typed ERALL?" "What would happen to the diskspace?") Save the workspace again under the old name (FILE) and discuss the contents of the two disk files and workspace.
3. Type CATALOG and discuss what appears on the screen in relation to what is in disk space.
4. Erase a file using the ERASEFILE command and model what happens using the cardboard computer.
5. Show students a model disk space with several Logo files (e.g., on the overhead projector) and ask them to model what happens when one of the files is loaded into the computer. For example, if the command LOAD "PYTHAGOREAN" is executed, and PYTHAGOREAN has three procedures — SIDES, PICS, and CALC — the students would need to place 4" by 6" cards containing the three procedures in memory and enter the procedure names in the workspace table.

We have found the above activities very useful in helping teachers and elementary students understand the complex concepts related to workspace management and files. We have also used similar activities to help students understand variables and other difficult concepts. When difficulties related to these concepts have been removed, students have been able to spend more time concentrating on the use of Logo for solving problems and for exploring mathematics and other subjects. We feel confident that they will be just as useful with other age groups and in a variety of learning settings.

1. A model is a simplified physical or pictorial representations of a complex idea or object. We have all seen the plastic or styrofoam models of molecules used in teaching science.

James H. Wiebe

California State University

5151 State University Drive, Los Angeles, CA 90032

Malorie L. Wiebe

Polytechnic School, Pasadena, California

Logo Connections

Say WHEN!

by Glen L. Bull and Gina L. Bull

This issue of the *Logo Exchange* contains our 62nd column. The total would have been 63, except that Tom Lough (the editor who preceded Sharon Yoder) allowed us a month off when our son, Stephen, was born. During the past decade Logo has increasingly grown in sophistication. Versions of Logo such as LogoWriter now contain features such as multiple turtles and interrupts. Multiple turtles are obvious, but what are interrupts? You probably have used them before, even if you haven't realized it. Interrupts provide a way of interrupting one program so that a second program can be run. For example, let's look at a very short LogoWriter program:

```
TO Race
FORWARD 1
Race
END
```

This program will run forever, or at least until the power is turned off. However, LogoWriter provides a way to interrupt the program so that it will stop. Run the procedure Race.

Race

Now press the Stop key (Open Apple-S). This key will interrupt the program, and stop the turtle.

LogoWriter will allow you to program your own interrupts, using the **WHEN** command. To demonstrate this, type the following short procedure.

```
TO Set.Keys
WHEN "R [SETSH 22]
WHEN "X [SETSH 0]
END
```

Then run Set.Keys and Race.

```
Set.Keys
Race
```

The turtle will begin racing across the screen. While holding down the Control key, press the letter "R". What happens when you type Control-R? The turtle should change into a rabbit shape as it moves across the screen. (That's because Shape 22 on the LogoWriter shape page is a rabbit.) LogoWriter has interrupted the first program — Race — in order to carry out a second command — SETSH 22.

Try typing Control-X. Did the rabbit change back to a turtle shape? The Race procedure should continue to move the turtle across the screen as this is happening. LogoWriter permits up to ten control keys to be programmed using the **WHEN** command:

N O P Q R and V W X Y Z

The control keys F (flip page), S (stop program), U (up), and D (down) are already reserved for use in the IBM version of LogoWriter, just as their Open Apple equivalents are reserved for use in the Apple version of LogoWriter. For that reason, you can not program these keys using the **WHEN** command.

It is not clear why some of the other control keys can not be programmed with the **WHEN** command. For example, it would be desirable to use Control-T to reset the shape to a turtle. However, Control-T cannot be programmed using the **WHEN** command, and therefore Control-X was substituted instead. For whatever reason — possibly a technical limitation — the **WHEN** command is restricted to use with only the ten keys listed above.

The rabbit shape from the LogoWriter shapes page faces left. Therefore we would like it to race across the screen from left to right. The following procedure will set the rabbit in the right direction.

```
TO Set.Rabbit
TELL
SETSH 22
SETH 270
PU
ST
END
```

As the rabbit races across the screen, a cage is placed above it. An interrupt will be used to drop the cage when the rabbit is below it. This procedure changes a turtle to the shape of a cage.

```
TO Set.Cage
TELL 0
SETSH 30
SETH 180
SETC 2
PU
ST
END
```

Logo Connections--continued

The next procedure raises the cage into place.

```
TO Raise.Cage
CT
TELL 0
SETPOS [0 60]
TELL 1
END
```

Try typing **Raise.Cage**. Was the cage positioned above the turtle? Finally a procedure to drop the cage is created. After the cage is dropped, the procedure checks to see if the rabbit is nearby. If it is, the word "Trapped!" is printed on the screen. The **Drop.Cage** procedure uses a second procedure, **Abs**, to help determine how far the rabbit is from the cage. (If you are using Version 1 of LogoWriter, substitute **Abs FIRST POS** for **Abs XCOR**.)

```
TO Drop.Cage
TELL 0
SETPOS [0 -40]
TONE 400 2
TELL 1
IF (Abs XCOR) < 10 [Restart]
END
```

When you type in the **Abs** procedure, be sure there is no space between the minus sign and the number one: "-1".

```
TO Abs :Number
If :Number < 0 [OUTPUT -1 * :Number]
OUTPUT :Number
END
```

When the rabbit is trapped by the cage, the **Restart** procedure provides directions on how to restart the program. The **Restart** and **Next.Line** procedures may contain LogoWriter commands which you have not used before. The **PRINT** command, of course, prints words on the graphics screen. The **TYPE** command, however, prints words below the graphics screen, in the Command Center. After a line has been printed in the Command Center, the **Next.Line** procedure moves the cursor to the next line. Finally, the **STOPALL** command stops all the procedures, including **Race**, so that the rabbit will stop racing across the screen after it has been trapped.

```
TO Restart
CC
CT
PRINT [Trapped!]
TYPE [Type STARTUP to restart the
```

```
program.]
Next.Line
STOPALL
END

TO Next.Line
TYPE CHAR 13
END
```

The cage can be raised and lowered by typing **Raise.Cage** and **Drop.Cage**. Interrupt keys provide an easier way to accomplish this. We have chosen Control-N to drop the cage and Control-O to raise the cage, but you can substitute any of the other keys which can be programmed with the **WHEN** command.

```
TO Set.Keys
WHEN "R [SETSH 22]
WHEN "X [SETSH 0]
WHEN "N [Drop.Cage]
WHEN "O [Raise.Cage]
END
```

Type **Set.Keys** to assign control keys to **Drop.Cage** and **Raise.Cage**. Then type Control-O and Control-N. Do these keys raise and lower the cage?

A special procedure, **STARTUP**, starts the program. **STARTUP** is a special procedure name built into LogoWriter. If a LogoWriter page contains a procedure named **STARTUP**, this procedure will be automatically run when you get the page. We have included some instructions in our **STARTUP** procedure, so that someone using the program for the first time will know how to trap the rabbit.

```
TO STARTUP
RG
CC
CT
Set.Keys
Set.Rabbit
Set.Cage
Raise.Cage
Instructions
Race
END
```

```
TO Instructions
CC
TYPE [Type Control-N to drop the cage
and]
```



```

Next.Line
TYPE [trap the rabbit.  If you miss,
    type]
Next.Line
TYPE [Control-O for another try.]
END

```

Type STARTUP to start the program. Use Control-N to lower the cage, and Control-O to raise the cage (if you miss the turtle). If you would like to make a special disk which automatically runs the program when LogoWriter starts up, name the page STARTUP as well. Then when the computer is turned on and LogoWriter is started, the computer will automatically go to that page when it starts up.

If you wish to make it easier to trap the turtle, you may wish to change the Drop.Cage procedure so that the cage must only be within 15 steps of the rabbit in order to trap it:

```
IF (ABS XCOR) < 15 [Restart]
```

On the other hand, if you wish to make the game harder, you may wish to change the Drop.Cage procedure so that the cage must be within 5 steps of the turtle in order to trap it:

```
IF (ABS XCOR) < 5 [Restart]
```

Those with a mathematical bent will have noticed that the line which checks to see if the rabbit is trapped is an algebraic inequality. This type of program could be demonstrated when inequalities are introduced. To simplify the inequality, the cage has been placed on the zero coordinate of the Y-axis. To increase the complexity of the problem, move the cage to one side or the other of the zero coordinate. Then challenge the class to find a way to determine whether the rabbit has been trapped.

Glen and Gina Bull
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.

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

Over the summer you may wish to catch up on your reading. If so, we would like to recommend our candidate for best Logo book of the year, and possibly the decade, *Visual Modeling with Logo: A Structured Approach to Seeing*

This book by James Clayson is the second in the "Exploring with Logo" series edited by Paul Goldenberg. The author teaches a course called "Problems in Visual Thinking" at the Parsons School of Design in Paris. This text grew from his experience in teaching the course. As he describes it,

My problem was to discover how to teach visual thinking to those students who had problems doing it naturally...The Logo computer language struck me as an appropriate medium of instruction — just enough of the visual and just enough of the analytical. (pp. xi-xii)

The text combines art with the analytic, from Mondrian to fractals. It does combine just enough of each. The book is caviar — the very best. It is published by M.I.T. press (Cambridge, MA 02142). Ask your bookseller to order a copy. You'll not want for summer entertainment.

Letters

[Once again, this letter was received via BITNET — what a delightful way to interact with colleagues throughout the world. We certainly welcome Uri's and Tami's comments! Ed.]

In response to your editorial "Logo: it's not just for little kids" (LX November '88), advocating the use of Logo beyond elementary school level, we would like to report to readers of the *Logo Exchange* on one such project — and one on a national scale. I am referring to the course (one year, 90 hours, 11th grade) *Introduction to Computer Science via Logo*, which the Israeli Logo Centre at the Technion (Israel Institute of Technology) has been commissioned by the Ministry of Education and Culture to develop and implement.

The course, which is part of a more extensive computer science curriculum for grades 10-12, is now being run experimentally in 20 classes (over 500 students) across the country. The participating population represents widely varied backgrounds: advanced and socially disadvantaged, urban and kibutz, Arabs and Jews. In developing the course, we relied on our experience in the teaching, research and development of Logo (we have been involved with Logo since 1980; the Logo Centre was officially established in 1984), as well as the experience and writings of others. We were especially influenced by the spirit of the following books, each excellent and exciting in its own way, but none quite suitable for the purpose and audience we were addressing: H. Abelson and G. Sussman: *Structure and Interpretation of Computer Program* (MIT Press, 1985); B. Harvey: *Computer Science Logo Style* vols. 1-3 (MIT Press, 1985-7); R. Pattis: *Karel the Robot* (Wiley, 1981); and J. Allen, R. Davis and J. Johnson: *Thinking About [TLC] Logo* (Holt, Reinhart and Winston, 1984) and, we need scarcely add, Papert's *Mindstorms*.

The course reflects modern trends in computer science as an independent intellectual discipline, in which applications and even the computer itself are no more considered the central objects of study. Nor is programming (i.e. the activity of writing programs that instruct the computer to do a prescribed task) in itself, sufficient. At center stage appear instead issues such as the *expressive power* of programming languages and the methods used to "control the *intellectual complexity* of programs" (Abelson and Sussman). These methods include abstraction, recursion, modularity and functional programming. As an illustration of the emphasis on expressive power, variables appear not as programming trick, not even as a useful general tool (though of course they are that as well), but primarily as enriching the language and increasing its expressive power. For example, you need variables to express in your language the *concept* rectangle. The book by

Abelson and Sussman, which is the textbook for the introductory computer science course at MIT, implements this approach using Scheme, a dialect of LISP. In our project this approach is adapted to the high school level, and to population of varying abilities, with Logo as an educational dialect of LISP.

Because of the emphasis on communication issues and on effective methods for dealing with complexity in view of the known strengths and weaknesses of the human mind, we tend to think of this as a human centered approach to computer science, an approach which is particularly suitable for the educational goals of a high school course. To us, this approach represents a fortunate junction where the value systems of the professional computer scientist and the educator meet.

Of course, the task of turning such lofty ideas into a working, real-life, high school course, taught by many teachers to many students, is beset by difficulties. We mention two main considerations that helped us cope. First, the computer which has been demoted to a minor role in our *theoretical* view of computer science, regains a central place as a *learning* tool. Here we prescribe to Papert's view that "children learn best by doing, and then thinking on what they did." The high-level concepts mentioned above are constructed gradually through reflection (usually a class discussion or written assignment) on carefully chosen activities. This makes for an interactive, lively, interesting learning and, incidentally, takes into account the interests of weaker students. They may fail to achieve the highest abstractions, but still enjoy the programming experience as a positive, worthwhile outcome of the course.

The second major consideration is that even if one is successful in developing a professionally and educationally sound course, the success of its *implementation* in the schools depends crucially on the teachers who are mostly unfamiliar with the innovative approach to both the subject matter and the teaching methods. This is added to the unfounded but widespread bias against Logo as a "language for babies," rendering a proper teacher preparation even more crucial. It was clear to us that whatever success or failure we have with the teachers on these fronts is likely to be reflected by a corresponding success or failure in their work with the students. We have therefore attached the utmost importance to (inservice) teacher education in the best tradition of the Logo spirit that we have been cultivating for years in our other Logo activities. Happily, the teachers reacted enthusiastically to both the new materials and the teaching methods emphasizing interaction, exploration, intuition and metaphors. One senior math teacher, for example, told us that not only did she enjoy the

workshop, but she thought her math classes would not look the same in the future.

As for the experimental classes it is too early in the year (and too late in this letter) to summarize it in detail, but half-way through the course (and with close contact with the participating teachers and students), the general feeling is that our goals are being achieved quite successfully.

Finally, if you are a "Logo purist," Brian Harvey's phrase "what happened to the revolution?" must have crossed your mind several times while reading this letter. Well, the way we saw it, that was the wrong question to ask in this case. Due to some rather special circumstances we had a chance to exert a strong influence on the national computer science curriculum, as regards both contents and teaching methods. Of course in the process we'd have to come to terms with the system, including some things we don't particularly like. Staying out of the system in order to preserve our purity, it seemed to us, would be too self-indulgent.

Uri Leron is an associate professor of math and computer science in the Department of Science Education; and Tami Lapidot is a graduate student. They can be reached at Technion - Israel Institute of Technology
Haifa 32 000, Israel
Bitnet: ur0128@technion

Attend the ICCE Summer Workshops:



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 (Burrowes) Yoder
Dates: July 9 - 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 the Leadership and Logo for Leaders Workshops!

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

About the Cover

An Interdisciplinary Approach to Teaching

The components of this fourth grade project at Hennigan School in Boston included the following: literature, *The Book of Three*, a unit on sound and LogoWriter as the tool to unite reading, unit reading, Language Arts and science.

Initially, the unit began with reading *The Book of Three* and studying the elements of a novel. Strong emphasis was placed on character development and analysis, plot and theme. Simultaneously, experiments with sound were being conducted in the classroom. Pitch and frequency were being explored. The computer provided a way to intertwine the important integral parts of this interdisciplinary unit.

Children selected their favorite character to depict on the screen. Since there were few illustrations in the novel the children needed to glean the book carefully for descriptive passages. They also needed to rely heavily on their own creativity to fill the gaps left by the paucity of illustrations.

The novel contained a character names Ffleuddur Fflam, a minstrel by trade. The children were required to become "minstrels" and create the words to a ballad for their chosen character. Each syllable was then assigned a tone on the computer.

The finished programs first drew the character and then almost simultaneously showed a line of the ballad on the screen as the music which accompanied the lyrics was heard. Reading, language arts, science, music and art were combined in a natural format. Modular programming was the mainstay of the children's work on the computer.

The computer provided a fundamental means to integrate varied subject matter in a natural way. Children need to see the relationships between and among subject matter. The computer was the tool which provided the means to a successful end.

Joanne Ronkin
36 Dunedin Road
Wellesley, MA 02181
617-235-3712

MathWorlds

edited by Sandy Dawson

The Council for Logo in Mathematics Education... An Update by Ihor Charischak, CLIME

NCTM affiliation - We made it!

As of last December, the Council for Logo in Mathematics Education (better known as CLIME) became officially affiliated with the National Council of Teachers of Mathematics (also well known as NCTM). One of NCTM's roles is to set the mathematical educational policy in this country. We have now entered its inner circle. What this means is that (1) we now get mail from people within the NCTM hierarchy telling us secrets that only Affiliated Groups (AG) are privy to and (2) we can now send a delegate to the Delegate Assembly that will be meeting at the annual NCTM meeting in Orlando, Florida next April. According to Kathy Layton, Chair of the Regional Services committee, we are entering a very exciting time in mathematics education. A committee has been hard at work developing a curriculum and evaluation standard for school mathematics which they will be presenting to the world very soon. Delegates to the Delegate Assembly in Orlando will have the opportunity to help guide the implementation of the *Standards*. George W. Bright (Univ. of Houston, Dept. of Curr. and Instruction, 4800 Calhoun Rd., Houston, TX 77004), who is our delegate representative, will be attending the meeting. Please call or write to him to share your thoughts and dreams about Logo and Math Education so he can convey them at the NCTM meeting.

Announcement: The Winter Issue of CLIME News will (oops..I mean *does*) focus on Math Microworlds.

It's strange to be writing about something that at present is not complete, but will be when you are reading this. After announcing this issue, a different kind of issue immediately came up. What exactly is a Logo microworld? And how can we tell if a Logo activity is really a microworld or not? To find an answer to the first question, I went to the source -- Seymour Papert's *Mindstorms* -- for a definition. According to Seymour a microworld is "a computer based interactive learning environment where the prerequisites are built into the system and where learners can become the active, constructing architects of their own learning" (p. 122).

From this definition I can infer at least three characteristics of a microworld:

1. A microworld contains a set of tool procedures that accomplish some task.

2. There are stated or implied goals or objectives which may take the form of problems to solve or questions to explore.
3. The context is interesting and encourages student exploration. So, it seems to me that a candidate for microworld status should contain these characteristics. Lets look at an example.

A "classic" example is POLYSPI (short for polyspiral).

```
TO POLYSPI :DISTANCE :ANGLE :CHANGE
  FORWARD :DISTANCE
  RIGHT :ANGLE
  POLYSPI :DISTANCE + :CHANGE :ANGLE
  :CHANGE
END
```

All the user has to do is type POLYSPI followed by three numbers or inputs and the procedure will produce some intriguing results. Students discover that stepping variables (holding some variables constant, while incrementing another) can be a powerful tool in testing hypotheses. Most experts agree that POLYSPI indeed is a microworld. But how can I be sure? When I posed this question to Ricky Carter (Lesley College, Cambridge, MA), he said that the issue for him is not whether an activity is a microworld or not, but rather how *microworldly* is it, and how we might make it more microworldly. For him microworlds involve a general (and developing) set of characteristics and no microworld has all of them.

Microworldly characteristics

Microworlds:

- are sufficiently bounded, yet sufficiently rich for discovery
- the structure (Logo procedures) can be transparent to the user
- are centered on set of powerful ideas
- involve a set of engaging or "neat phenomena"
- are easy to get started with
- have many ways to explore an idea or solve a problem
- entices student to ask questions
- encourage users to explore on their own, build new structures, and personalize their learning in some way
- can help students bridge the gap between intuitive and more formal understanding of mathematical ideas and principles

POLYSPI is an example of a very microworldly microworld.

A Clearinghouse of Microworlds

The last issue of the CLIME news announced the intention of CLIME to create a clearinghouse of Logo math microworlds and make them available to interested readers. The

theme of the next issue of CLIME news will be math microworlds. What follows is a summary of some of the microworlds that will be described in more detail in our newsletter and made available on disk. Hopefully, you will find that these activities have a high level of microworldliness. If you don't, please send us some you think are better and next year we will publish the "Best" of Math Microworlds for 1989.

Microworlds from Terrapin Inc. and Logo Computer Systems Inc.

Congratulations to our Logo makers for finally producing some microworlds for exploring mathematics! Terrapin recently published *Logo Innovations* which is a collection of 16 creative activities that students can use to learn in all curriculum areas. Most of the activities involve some mathematics, but the ones that I feel would be of most interest to a math teacher are:

Iso-The Amazing Isosceles Triangle

Explore introductory geometry with Logo

Logo Miniature Golf

Used the turtle as a golf ball in a game of estimation and strategy

Mandalas

Uses coordinates and variables to create interesting designs

Proportions

Explore mathematical relationships visually

Vectors

An activity for secondary math class

If you want the activities above and are interested in the others, the good news is that they all come on one disk at a very reasonable price. (Note: *Logo Innovations* requires *Terrapin Logo* or *Logo PLUS*.)

Terrapin Inc. will also be publishing another microworld called *Logo Math: Tools and Games*. This package written by CLIME member Henri Picciotto includes 14 programs that support the secondary math curriculum.

Logo Computer Systems Inc. published Brian Silverman's *The Phantom Fishbowl*. Inspired by John Conway's Game of Life, this microworld is inhabited by objects in the form of patterns on the screen and are given life by a set of rules that govern how old patterns give rise to new patterns. The user makes up a set of rules, puts some patterns on the screen and says "go". What happens irresistably evokes biological images and metaphors.

A Sample of CLIME Microworlds

Graphing - An Apple LogoWriter Tool (Ken Goldberg)

This microworld allows you to study data as it is being entered into the computer by displaying the data in a variety of ways (in its original raw form, in a table, and as a bar graph.) You can easily find the data's numerical count, mean, and standard deviation.

Logo Pattern Block Fractions (Judith Harris)

Children can explore fractional relationships on the screen using pattern blocks. Best used with real pattern blocks.

Geo Shapes (Ihor Charischak)

In the early days of Logo, Robert Lawlor created the BEACH microworld whose purpose was to get beginning readers to learn to read with a minimal direct instruction. GEO SHAPES was inspired by that microworld and its purpose is for children to become familiar with names of geometric shapes and to explore with them.

Quarts (Ihor Charischak)

A tool for solving the famous quarts problem as posed by George Polya.

Lottery (Bill Craig)

A model for exploring probability by playing the Lottery.

These are just a few of the titles. If you wish to receive the CLIME microworlds Disk, you need to do two things.

- (1) Become a member of CLIME. This will cost you \$10 (\$15 if you live outside of North America).
- (2) Send CLIME a self addressed, stamped 5 1/4 disk inside a mailer to:

CLIME
10 Bogert Ave.
White Plains, NY 10606

In return you will receive our special Microworlds issue and disk. If you want both the IBM and Apple versions, send two disks. (So far we only have Judi's program for the IBM. Hopefully, by the time you read this there will be more.)

Assessing Logo Learning In Classrooms

VIII. Managing Workspace and Files

by Dan Watt

This is the last 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 write about each of the subheadings included in the eighth cluster of critical aspects, Managing Workspace and Files. 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.

Although this is the last column in the series, I know that I've only scratched the surface of the topic. Assessment is just one tool for improving the teaching and learning of Logo. I believe that Logo teachers will teach more effectively if they regularly examine the work their students are doing. To assess Logo learning properly, I believe that teachers should:

- have a good idea about some of the important ideas their students might be learning from Logo;
- have ways to determine which of these ideas students are and are not using in their work;
- have a sense of how these ideas typically develop over time;
- have strategies for supporting students to use particular Logo ideas more effectively as their work progresses from one project to another.

Teachers who use assessment as part of teaching in this way will need to set their own priorities, create their own lists of objectives. My intention in writing these articles is to share some of the ideas we identified as important in our research project, and to offer some examples of student work which demonstrate whether or not those ideas are being used.

Two sad stories

We did not originally consider workspace and file management to be a critical aspect of Logo learning. It was the teachers in the project who insisted that it was an important topic to include. They had observed that students who could keep track of their work more easily could make more progress in other areas of Logo. It was as simple as that. Unfortunately,

we found that few teachers knew how to help their students learn to keep track of their work.

Sad Story #1: Once I asked a fourth-grade teacher whether her Logo students kept journals or files or printouts, or any other off-computer records of their work. "We don't have to do that in fourth grade," she explained. "In third grade, students don't write procedures, so they keep a journal to record what they have done. But in fourth grade they do learn to write procedures and save them on disk. This is much easier for them and takes much less time than keeping written records."

Sad Story #2: This same teacher taught in a self-contained classroom with five computers. Four of the computers were connected to a single printer through a switch box. I asked the teacher if she could show me any examples of children's Logo work, pictures or printouts. She apologized that she did not know how to print copies of Logo procedures or pictures. "What do you use the printer for?" I asked. "To print out their compositions when they use word processing," she explained.

There is an interesting history behind this story. Logo was the very first elementary computer application used in this school district. From the beginning, the district decided to put a computer in every classroom in grades 3-5, but could not afford to purchase printers to go with them. An elementary Logo curriculum was written which ignored the use of printers entirely! Later, when word processing was added to support writing instruction, more computers were purchased along with one printer per classroom. The teachers were trained to use the printers for word processing, but not for Logo. So what began as a reasonable compromise, to stretch computer resources as far as possible, deteriorated into a form of QWERTY, because when printers were added, teachers were not shown how to use them for Logo.

What's Critical About Workspace and Files?

In his famous polemic against our society's growing dependence on computers, *Computer Power and Human Reason* (W. H. Freeman, 1979), computer scientist Joseph Weizenbaum argued that when computers made it possible to store and retrieve more data than could ever be kept in a set of physical file drawers, social organizations grew too large for human beings to manage effectively, and thereby became both impersonal and inefficient. Logo was designed to do the opposite, that is, make computers more amenable to human control and understanding. And yet, when information is stored invisibly on magnetized disks, and when graphic images disappear, perhaps forever, when the power is turned off,

the resulting confusion about what's been done and how to recreate it and move on, can be as frustrating and dehumanizing to an individual as any computer problem with an airline reservation or telephone bill.

Imagine a file drawer without any folders — just a huge pile of papers of different sizes, scraps, notes, bills, and so forth, all stuffed in with no organization whatsoever. Or imagine a loose leaf notebook without dividers, and suppose that every time you write something, you add it at the end of the notebook, but whenever you use the notebook again, all the pages have been rearranged so that you can't remember what's there, or what order they were written in. That's what a Logo workspace or disk can be like for someone who doesn't know how to organize them.

Unlike using 360 degrees in turtle geometry, or using subprocedures in programming, understanding file and workspace management is not important because the intellectual ideas are especially powerful, or especially accessible through Logo. Rather, the ideas and strategies of workspace management have a negative power if you don't understand them. Put another way, if you do not manage your workspace and files effectively, you are likely to waste tremendous amounts of time searching for and recreating past work, and miss out on many significant learning opportunities, simply by not having easy access to the work and thinking you have already done.

As far as I can tell there's only one powerful idea associated with file and workspace management: If you're working with ideas and information it's a good idea to organize them so that you know what you've got, when you created it, and how you can find it again. Some of this can be done using the computer itself; however, I find that hard copy printouts, and a notebook or journal are essential for most people to keep track of their work. In the sections that follow, I will describe several stages that we have observed as students learn to keep their work organized.

A. Saving procedures on disk in an undifferentiated work space

This is an almost universal starting point — whether a student is using LogoWriter, which utilizes a page metaphor, or a version of Logo which uses SAVE and LOAD or READ to save and recall files. The natural tendency is to ignore the complexities of file management at first, and save a bunch of unrelated exploratory procedures on the same page or workspace. An advantage of this is that a Logo beginner is free to concentrate on learning to use the turtle, to write procedures and subprocedures, without having to worry about naming files and recalling them correctly.

Subtle differences between versions of Logo may influence how soon a learner breaks away from this extremely simplistic approach to saving work on disk. For instance, Terrapin Logo makes it easy to keep saving work in the same file, over and over again. So does LogoWriter, which saves automatically whenever you leave the page you're working on. Other LCSi versions of Logo won't let you save new material in an old file until you erase the previous version of the file. I suppose that this would lead students to use multiple files at an earlier stage of their work.

B. Saving work in more than one file

Once a student begins work on one or more specific projects, it can be useful to save work in more than one file. At first, a student may simply name each new file slightly differently. Brad, a fourth grader who worked for many weeks on a city project, changed his file names several times as he worked. His names included: CITY, CITY.2, GOOD.CITY, BIG.CITY, BIG,CITY (possibly a typo), and so on. All of his file names were also names of procedures in his workspace. Brad kept no written records about what was in these files, and soon became confused about which one was the most up-to-date. Sometimes he would load in several different files, looking for a particular version of his project, and then save the entire workspace using just one of the file names.

Two strategies might have helped Brad. First, he could have been encouraged to save files with sequential names: CITY1, CITY2, CITY3, and so on. In this way, he would know that his earlier work was still available, and that the file with the highest number was the last work he had done. It would have been even better if he had kept a journal, and had written down the names of new procedures, the dates they were written and the file they were saved in.

C. Distinguishing between file names and procedure names

Realizing that all your current procedures are saved when you save a file or a page is a key understanding for managing files and workspace effectively. Brad may have been suffering from a very common misconception among beginning Logo users. He probably thought that he needed to save each procedure he wrote in a file with its own name, and may not have been aware that all his procedures were saved every time he used the SAVE command. Ironically, this misconception is reinforced by some teachers (and even one early Logo textbook) who teach students to save each new procedure in a file with its own name, as a way of making sure that a student's work is saved continuously as he or she works.

LogoWriter users may be less likely to hold this misconception. It is clear that an entire LogoWriter page is saved

Assessing Logo Learning In Classrooms--continued

every time you leave the page. And whenever you move to the flip side of a page, you are aware of all the procedures written on it. This feature makes other aspects of workspace management more difficult, but it is extremely useful in helping students avoid the procedure name/file name confusion.

D. Using Logo commands to keep track of work

Most versions of Logo (other than LogoWriter) include special primitives that help users understand the contents of their workspace and disk. For example, students can be encouraged to type POTS (printout titles) before saving a file. This would help them see that all the procedures listed were being saved every time. If they want to see the text of a specific procedure, they can type PO CITY or PO "CITY (depending on which version of Logo they are using). (Note: LogoWriter does not have these features. A user who wants to know exactly what procedures are on a particular page must go to the Flip Side and page through several screens to see all the procedures written there.)

Similarly, the CATALOG command shows all the files on a particular disk. The LogoWriter CONTENTS PAGE performs the same function much more naturally, so that no special command is needed.

E. Making journal notes or hard copy printouts of workspace information

If students have a printer available during their working sessions, they can print out the current contents of their workspaces, procedures and disks, and save them in a file folder or paste them into a notebook or journal. Although time-consuming, this may be the best way to keep track of work done on a day-to-day basis. It's fairly simple to direct output to a printer, and then use POTS, PO (followed by a list of procedures), PO ALL or POALL, or CATALOG, to print the information needed. PO ALL or POALL prints out all procedures and variable names in a workspace, and may generate more paper and more data than is useful. And it prints procedures in a random order. It is often more useful to use PO to print out only those procedures which have been changed in a given session, or to print a particular set of related procedures in some useful order.

LogoWriter users can print the entire flip side of a page by typing PRINTTEXT or PRINTTEXT80 from the Command Center, but it is more difficult to print out individual procedures. Also, LogoWriter has no POTS command, and no way to printout the CONTENTS PAGE, unless your printer is equipped with a printscreen button.

If printers are not available, notes in a journal are the next best thing. Students may want to copy the list of procedures

generated by POTS and the list of files generated by the CONTENTS command.

F. Cleaning up workspace and files

One of the last workspace-related skills learned by most Logo users is cleaning up workspace and disk files. What I mean by "cleaning up" is using the ERASE command to eliminate unnecessary procedures from a workspace, and saving it in a less cluttered form, and using the ERASEFILE command to erase unneeded files from a disk. Students who regularly use POTS and CATALOG to track their work, are more likely to realize that a particular procedure or file is no longer needed. LogoWriter users can accomplish the same thing by selecting the text of an unneeded procedure and using the delete key, or selecting an unneeded page from the CONTENTS page as using the delete key.

It's a bit more complicated when you want to clean up a file by dividing it into two or more separate files. In such a case you'll need to save the file twice, with different names, and erase different procedures from each file.

Why is workspace management so difficult to teach?

I think there are three reasons why teachers find it difficult to support their students in managing their Logo files and workspaces effectively:

First, many teachers have not learned the concepts and skills of workspace management themselves. Many Logo training sessions for teachers, especially ones that consist of one or two after-school sessions designed to cover what their students are expected to learn, do not cover specialized commands such as POTS, PO and ERASE, or explain how to print hard copies of Logo procedures and graphics.

Second, many teachers are afraid that if they teach their students how to erase procedures or files, that essential work may be lost. This is a very real concern! Most computer users have had similar experiences at one time or another, and we want to protect our students from having the same frustrations we have had. One solution is to back up files regularly (as a teacher, I often do this for my students, making a back up copy of student disks once a week or so, and storing them in a safe place).

Finally — and I think this is the biggest problem — because computers are a scarce resource, we all want to maximize the benefits of computer access for our students. We want our students to devote as much of their time as possible to creative work and problem-solving, so we don't require that they keep notes or make printouts, because of the

time and mechanical effort involved. What we often don't realize is that huge amounts of time are wasted by students trying to find old work, remember what a particular procedure does or what file last week's work was saved in. A few minutes routinely spent in keeping records of each session, can save hours in the long run. In addition, students who make printouts of their work can examine and think about it while they are away from the computer, and this might make their computer time more valuable and effective.

Assessment is important for students as well as teachers

I've said throughout this series of articles that ongoing assessment of student learning is an essential ingredient of effective Logo teaching. In order to assess Logo work, teachers need data — especially printed examples of the procedures and graphics that their students are creating. If students learn to keep track of their own work and to print out procedures regularly, it could make a teacher's assessment work much easier.

But students may also benefit directly from keeping track of their work, especially if they keep printouts of their procedures and pictures. Students who examine their previous work as they think about what to do next, may very well become more effective problem solvers. And students who habitually keep track of their work have the opportunity to assess their own Logo progress, to become aware of how their own thinking is developing. Helping learners to think about thinking was one of the fundamental goals of Logo's developers. To the extent that it contributes to that objective, work-space and file management may involve powerful learning after all!

Dan Watt
Educational Alternatives
Gregg Lake Road
Antrim, New Hampshire 03440

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.

The Logo Institute, June 23 - 30 The Logo Institute Workshops, June 23 - 24

At Lesley College, Cambridge Massachusetts
Immediately following NECC (June 20-22)

What are your plans for professional renewal this summer?

Consider the Logo Institute (6th annual)

- experience a seven-day immersion in a learning environment for Logo-using educators from the US, Canada and other countries
- use the full scope and power of Logo and its educational philosophy to create programming, research, teacher education and curriculum projects
- (optional) receive three graduate credits from Lesley College

Participate with An Outstanding Faculty

- Dan Watt and Molly Watt, Logo Institute founders
- Ricky Carter, Lesley college Faculty
- Scholars in Residence:

Logo Exchange columnists Eadie Adamson and Gary Stager

- Faculty will offer lectures and demonstrations, lead short courses, provide technical assistance and coaching, and support participants in completing their own creative projects.

Attend the Logo Institute Workshops

(for those unable to attend the entire Institute)

- Two days of hands-on workshops led by Logo Institute faculty and other leading Logo educators
- Hands on workshops include:
- Supporting mathematical learning with Logo
- Supporting language learning with Logo
- Assessing Logo learning in classrooms
- Getting a jump start with Logo-Logo
- Workshop participants may register for one optional graduate credit

For further information and registration materials, contact:

Ricky Carter
The Logo Institute
Lesley College
29 Everett Street
Cambridge, MA 02138-2790
617-868-9600 Ext. 370

Logo: Search and Research

Representating and Monitoring by Douglas H. Clements

"We got it!" "Well, let's think and make sure."
"Put 70." "70? We already did 50...type FOR
WARD 20."

"Let's make a list of everything we tried and see
which ones [inputs] are best."


Recall the problem-solving metacomponents—the "executives" of peoples' "mental company" that are critical to successful problem solving. They included:

- deciding on the nature of the problem;
- selecting a representation;
- selecting a strategy for combining components (planning); and
- monitoring one's solution processes.

We've discussed several in previous columns. This month we'll discuss two more.

Which of the metacomponents are the children using in the three short vignettes above?¹

Building Representations


Logo can help students represent problems in ways that are meaningful to them. For example, they can relate to turtle geometry in terms of moving their bodies or drawing. Programming in turtle graphics also encourages multiple, linked representations. That is, students might represent their Logo project in several ways: as a drawing, as a verbalized solution, and as machine-executable programming code. More importantly, they can build bridges, or links, between these representations. Such links promote rich, integrated, understanding—in fact, being able to relate one representation (e.g., ) to another (1/2) is, to a large part, what understanding means.

Does programming in Logo effectively build such links? Unfortunately, research is not revealing. It is scant and its results are mixed (Clements, 1986; Lehrer & Randle, 1987; Silvern, Lang, McCary, & Clements, 1987). With so little guidance, we can only speculate as to how to realize Logo's potential in this area.

Most probably, we should encourage and support the use of a several linked representations, including diagrams, English descriptions (oral and written in journals), computer code,

patterns, and tables. We should discuss the relationships between them and the ways in which each representation can help solve problems.

We shouldn't neglect having students of any age continue to relate turtle geometry to their own movements. For example, one technique is to "chalk it, walk it, talk it." Students draw the desired design with chalk (inside, use masking tape on the floor). Then they walk the design and simultaneously "talk through it" (describe exactly what they are doing in turtle commands), recording these steps so that they can type them at the computer.

"Representing" a problem in turtle geometry often involves finding a good way to break the problem down. For instance, in planning to draw , many students initially plan to go around the outside (perimeter) of the figure. One teacher discussed how one could examine the drawing to find the most basic shapes. Her students determined that if you saw the drawing as composed of rectangles, then that was the only basic procedure you needed.

Complementary experiences might be combining and repositioning shapes to create new shapes. The goal is to help students construct cognitive building blocks that are important in representing spatial problems. For example, Davis (1984) describes the cognitive building blocks needed to determine the area of a rotated square on a geoboard. In addition to mental images of squares and triangles, he cites mental representations of the acts of rotating and translating triangles, of putting them together to make other shapes, and even of cutting apart squares to get triangles.

Monitoring Solution Processes

Research clearly shows that most students do not monitor their own problem solving, right up through the college level (e.g., Schoenfeld, 1985). After they begin working on a problem, they rarely pause to see if their chosen strategy will help them solve it. They do not check their work for mistakes. Many believe that they fail because they are "stupid"; most believe that errors are to be hidden and that little can be learned from them.

Why might Logo help? In computer programming, errors are unavoidable. Few if any write programs that are free of bugs on the first try. And when they occur, they don't hide themselves! Ideally,

Experience with computer programming leads children more effectively than any other activity to "believe in" debugging. ...children learn that the

teach too is a learner, and that everyone learns from mistakes. (Papert, 1980, p. 114)

So, the act of debugging Logo programs that do not do quite what was intended provides students with valuable experience in using their monitoring skills.

Does Logo actually help? Qualitative reports have suggested that students working with Logo do indeed "step back to think about their thinking." Kull (1986) observed students spending considerable time discussing "Why is this plan better than this other plan?" Students in the Clements and Nastasi (1988) study exhibited a higher percentage of monitoring behavior than those in a control group. For example: "No. You're gonna go off the screen [with that numerical input], I'm telling you!" (Recall also the statements at the beginning of the column.)

Transfer to non-computer tasks has also been reported. For example, in several studies, students were given a set of incomplete directions for tasks. In one task, eight alphabet cards were divided equally between the experimenter and the students. Directions given included each player laying out one card at a time, looking for the "special card." There is, however, no mention of what the "special card" might be. Logo students were more likely to realize that they did not (i.e., could not) understand such directions (Clements, 1986). Note that both Logo and these tasks involved finding errors in sequences of directions. Does this mean that students learned only this specific skill?

Apparently not. In another study, students were given problems that purposely misled via extraneous or irrelevant information. For example, "When Albert was 6 years old, his sister was 3 times as old as he. Now he is 10 years old and he figures that his sister is 30 years old. How old do you think his sister will be when Albert is 12 years old?" Again, Logo students were more likely to find and fix the error in the problem (Clements, in press). In fact, an increase in cognitive monitoring following Logo experience is one of the more consistent research findings (Clements & Gullo, 1984; Lehrer & Randle, 1987; Miller & Emihovich, 1986; Silvern et al., 1987).

What can we as teachers do to help? It is noteworthy that all these studies featured supportive, frequent adult-student interaction. So, there is a lot we can do to help students develop monitoring abilities. First, we can help them develop strategies for debugging proper—fixing Logo problems. Second, we can help them generalize this to other situations.

Regarding debugging within Logo, Carver and Klahr (1986) constructed a precise model of debugging—precise enough to actually run as a computer simulation. Students trained to use the model significantly increased their debugging skills and transferred these skills to a different task. Simplified, the model is:

1. Check for bugs. Run the program. See if there are bugs.
2. Identify the bug.
 - A. Describe the discrepancy between your goal picture and the turtle's drawing.
 - B. Based on this, propose a specific type of bug that might be responsible. For example, if the orientation is off ("This is going over here instead of down"), an angle/rotation is probably buggy.
 - C. If more than one possibility exists (e.g., "These are spread apart too far" could mean a distance bug or an angle/rotation bug), describe the discrepancy more precisely.
 - D. When only one possibility remains, go to the next step.
3. Locate the specific bug.
 - A. Represent the structure of the program to investigate the probable location of the bug. For example, can the problem be identified as occurring within a REPEAT instruction? Is it in a certain subprocedure? For example, is the problem that a SQUARE procedure is not producing that shape? At worst, trace the entire program.
 - B. Check the location for the bug (e.g., for angle/rotation bugs, look for the faulty RT or LT).
4. Correct the bug. Examine the goal picture to determine the appropriate correction. Replace the buggy instruction with the correct one. Then go back to step 1 and try again.

Students might record the discrepancies they have found and the type of bugs that caused them for future reference.

The "tracing" mentioned above is an essential tool. Students should pretend they are the computer, executing each instruction that they read exactly as the computer would

Logo: Search and Research--continued

do. This is difficult; it's so easy to do what you meant to tell the computer to do! In all situations, help students guard against believing the computer has intelligent interpretive powers—that it “knows what they mean.”

Some versions of Logo offer debugging procedures that are useful in tracing. Most offer PAUSE and CONTINUE, commands that are placed in a program at strategic points. When the program reaches the point of a PAUSE command, it is suspended. Students can examine the output, type other Logo instructions to “see what happens at that point,” print the value of variable, and so on. They can then type CONTINUE, and the program will continue running from where it left off.

TRACE (or STEP in some versions) allows students to run a program one line at a time. After each instruction is executed, the program halts until a key is pressed. Such tools make tracing easier, but students still must be taught when, why, and how to perform such monitoring actions.

As teachers, our second task is to ensure that students generalize debugging skills as widely as possible. The following teaching strategies have been used successfully in research projects.

- Highlight the need for debugging, point it out when children evidence it, and specifically suggest it when it is needed. Promote a “learning from debugging” atmosphere. Expand this notion beyond a limited “fix a computer program” perspective to the use of debugging as an allegory for cognitive monitoring in myriad situations.
- Stress that bugs in programs and in other school work should be welcomed, not avoided. They lead us to reflect on their own thinking processes and reveal misconceptions.
- Model specific monitoring strategies and arrange for supportive, frequent adult-child interaction that encourages children to apply these processes.
- Use such questions as:
 - What have we done already?
 - What still needs to be done?
 - Why are we doing what we are doing?
 - How will it help us?
 - What's going to happen when you type that.
 - Do you know why that's there?
 - Are we solving the problem we set out to solve?

Something is wrong; what exactly didn't work out?
Does our solution really solve the problem?
Does it make sense?

1. The first two demonstrate “monitoring solution processes”; the third, “selecting a representation.”

References

- Carver, S. M., & Klahr, D. (1986). Assessing children's Logo debugging skills with a formal model. *The Journal of Educational Computing Research*, 2, 487-525.
- Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, 78, 309-318.
- Clements, D. H. (in press). Metacomponential development in a Logo programming environment. *Journal of Educational Psychology*.
- Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76, 1051-1058.
- Clements, D. H., & Nastasi, B. K. (1988). Social and cognitive interactions in educational computer environments. *American Educational Research Journal*, 25, 87-106.
- Davis, R. B. (1984). *Learning mathematics: The cognitive science approach to mathematics education*. Norwood, NJ: Ablex.
- Kull, J. A. (1986). Learning and Logo. In P. F. Campbell and G. G. Fein (Eds.), *Young children and microcomputers* (pp. 103-130). Englewood Cliffs, NJ: Prentice-Hall.
- Lehrer, R., & Randle, L. (1987). Problem solving, metacognition and composition: The effects of interactive software for first-grade children. *Journal of Educational Computing Research*, 3, 409-427.
- Miller, G. E., & Emihovich, C. (1986). The effects of mediated programming instruction on preschool children's self-monitoring. *Journal of Educational Computing Research*, 2(3), 283-297.
- Papert, S. (1980). *Mindstorms*. New York: Basic Books.
- Schoenfeld, 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.
- Silver, S. B., Lang, M. K., McCary, J. C., & Clements, D. H. (1987 April). *Logo, teaching strategies, and computer effects on metacognition*. Paper presented at the annual meeting of the American Educational Research Association, Washington, DC.

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

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		

This month's column gives *LX* readers a look at an inspiring Colombian project using Logo with deaf and blind children. The following report was written by Pedro Fernando Cuadros Marin of the Institute for the Blind and Deaf in Cali, Colombia.

The Institute for the Blind and Deaf Children is a non profit private institution created in 1940 by Luisita Sánchez de Hurtado. Its objective is to help blind and deaf children develop through early diagnosis of auditory and visual alterations, and through rehabilitation and integration of these children in the society as independent and useful individuals. The institute has a complete medical facility and offers several rehabilitation programs such as early stimulation for deaf and blind children, educational programs for these children, community integration programs, education of parents of deaf and blind children, outreach rehabilitation programs, research programs, and recently, a computer education program for deaf and blind children.

The computer was introduced in the institution as an educational tool. The basic interest was to help deaf and blind children to develop procedures and strategies for production of logical reasoning. In order to achieve these objectives the language and methodology chosen was Logo. This choice was based on the results already produced by the use of Logo with handicapped children at MIT and the work of Antonio Battro with deaf children in Argentina.

Before disseminating Logo throughout the institution, the researchers tried to understand how Logo could be used by the deaf and blind children and what would be the benefits of its use. This investigation was developed with a group of nine deaf and three blind children ranging in age from 8 to 11 years who were in the process of acquiring reading and writing skills. The blind group included one male and two females with vision capabilities inferior to 20/200.

The objective of this work was to start an educational program for these children and to investigate several issues such as whether these children would be able to use Logo, how Logo could help these children's rehabilitation process and whether Logo would help their visuo-motor coordination, visual memory, perceptual development and spatial orientation. Initially, the deaf children were evaluated with visuo-motor, memory, perception, spatial orientation and reasoning tasks. Then they used Logo for 18 sessions and were evaluated at the end of their Logo usage period.

In order to develop this work three NEC computers were installed in the institution. The deaf children were divided into three groups of three children and the blind children constituted another group. Each child worked individually on his own computer for the 18 sessions and received assistance from a monitor. The Logo version used was for the NEC computers with commands in English.

The results showed that the deaf children were able to learn the turtle drawing commands and used them to produce several drawings which were very well elaborated and creative. However, they did not reach the stage of defining procedures. They were able to work independently, showed a great interest in their activities, and were very motivated to work with the computer. In terms of their motor and perceptual evaluation, the tests showed a great improvement in their visuo-motor coordination. There was no difference in other skills. However, the computer work made it possible to diagnose other aspects of their cognitive abilities that were not evident prior to the computer experience, for example, to communicate between themselves and to exchange ideas. Also, the children whose tests showed memory problems, deficiency in their fine motor coordination, and difficulties with laterality and with spatial concepts, demonstrated behaviors at the computer consistent with this evaluation. It was possible to observe that their work was slow, they had diffi-

Global News--continued

culty in typing, were very impatient, frustrated, forgot commands and were very disorganized and confused about their working plan and manipulation of the computer.

The work with blind children showed their great difficulty with laterality and fine motor coordination. However, the computer enabled them to exercise their debugging skills, be more patient with their work, exercise motor coordination, and play with colors and geometric concepts. It also enabled them to understand that the cause of an error is not only a problem of vision but thinking abilities as well.

For additional information about this work write to:

Pedro Fernando Cuadros Marin
 Instituto para Niños Ciegos y Sordos de Cali
 Carrera 38 D-29-39
 Cali, COLOMBIA

Dynamic Classroom Press

10 Bogert Avenue, White Plains, NY 10606 (914) 946-5143

announces

two new publications which will support your teaching
 and learning with *LogoWriter*™ and *LEGO TC Logo*™

A
BOOK



Creating Dynamic Stories with LogoWriter



by Ihor Charischak

\$19.95
+ \$2.50 shipping

Creating Dynamic Stories shows you how to put on a dynamic story or play. You learn to build a set, choose your actors (turtles) and costumes, choreograph the action, present the script dynamically, and put on an opening night performance.

A NEWS
LETTER



Scenes from the Dynamic Classroom

featuring
LogoWriter
LEGO TC Logo

\$35.00
+ \$2.50 shipping
(4 issues)

Scenes will bring you creative lesson plans, microworlds, sources of learning materials, opportunities for networking with educators, and a **bonus disk** full of useful examples.

AppleWorks for Educators, by Linda Rathje, really shines. The new edition has been expanded to include sections for :

- mail merge
- integration activities
- a glossary, and
- software applications.

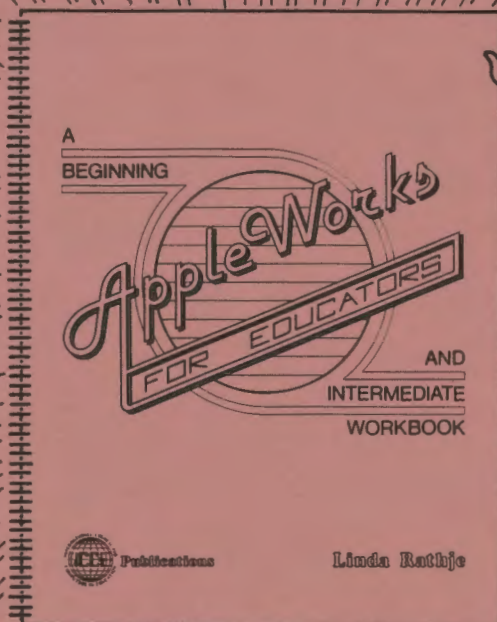
Each section provides step-by-step instructions. Beginning and intermediate *AppleWorks*® users learn word processing, database and spreadsheet management, and printer options.

Your copy is laser printed, spiral bound, and includes a data disk of working examples. Add *AppleWorks for Educators* to your harvest this year and watch them shine.

Single order price: \$22.95

To order from ICCE see page 56.

We've polished up a proven favorite!



Give your students a mind of their own

Teaching Thinking Skills with Databases



Expand their minds and your teaching capabilities on any subject with *Teaching Thinking Skills with Databases*. There are no limits on what you can do with this step-by-step guide for Grades 3-8.

Fifteen steps progress from lower order to higher order thinking skills, and each step is illustrated with scripted lesson plans on the 50 United States. (The states unit serves as a model for any subject area.)

Teaching Thinking Skills with Databases contains 14 data files on disk and 46 worksheet and transparency masters.

Teach with databases in any subject. *Teaching Thinking Skills with Databases* is available for *AppleWorks*®, and a site license is included. By Jim Watson. Single order price: \$30.00

For more information, see Jim Watson's article in *The Computing Teacher*, Aug./Sept. 1988. To order from ICCE, see page 56.

LONG DISTANCE LOGO

Educators—You don't have to go to 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.

SAVE \$\$\$

If your district supports group training, *you* save money! Districts enrolling six or more educators in this course will receive a reduction in the fees of each person enrolled. To qualify, your district must provide lab facilities and a resource person with experience in computing to help answer questions.

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 the University of Oregon and carries graduate credit from the Oregon State System of Higher Education.

ON LOGO VIDEOTAPES

School districts may acquire a license for use of the ON LOGO package of eight half-hour videotapes and 240 pages of supporting print for \$599. For a one-time fee of \$1,295, the package may be obtained with both tape and print duplicating rights, enabling the district to build libraries at multiple sites.

Group Enrollment. A tuition of \$260 per participant is available to institutions that enroll a group of six or more educators. This special price does not include the ON LOGO videotapes. Your group must acquire the tapes or have access to them. Once acquired, the library of tapes and materials may be used with new groups enrolling for the same reduced fee.

Individual Enrollment. Educators with access to the tapes may enroll individually for \$290. Tuition including tape rental is \$320. A materials fee of \$60 per enrollee is charged for texts and a packet of articles. Enrollees who already have the texts do not need to order them.

Additional information and order blanks can be obtained from: LONG DISTANCE LEARNING, ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905. Ph: 503/686-4414.