

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

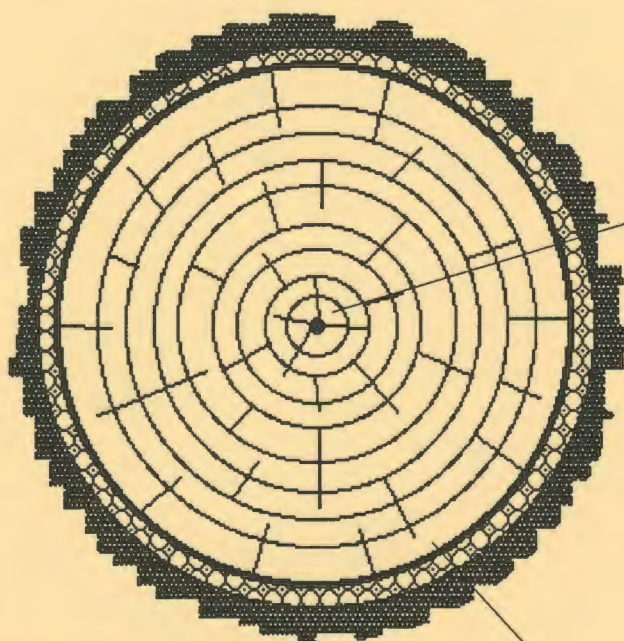


LOGO EXCHANGE

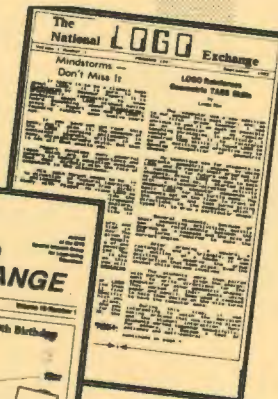
Fall 1991

Volume 10 Number 1

Logo Exchange Celebrates 10th Birthday



1982



Also in this issue:
A Time of Decadence?
Logs, Logo, and Arboreal Aging
Seymour Papert's Reflections

1991-1992

International Society for Technology in Education





LOGO EXCHANGE



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

Founding Editor

Tom Lough

Editor-In-Chief

Sharon Yoder

Associate Editor

Judy Kull

Assistant Editor

Ron Renschler

International Editor

Dennis Harper

Contributing Editors

Eadie Adamson

Gina Bull

Glen Bull

Doug Clements

Sandy Dawson

Dorothy Fitch

Judi Harris

Mark Horney

SIGLogo Board of Directors

Lora Friedman, President

Bev and Lee Cunningham,
Secretary/Treasurer

Publisher

**International Society for
Technology in Education**

Dave Moursund, Executive Director
C. Dianne Martin,

Associate Executive Director

Anita Best, Managing Editor

Talbot Bielefeldt,

Associate Editor

Lynda Ferguson,

Advertising Coordinator

Ian Byington, Production Assistance

ISTE BOARD OF DIRECTORS 1991-92

Executive Board Members

Bonnie Marks, President

Alameda County Office of Education (CA)

Sally Sloan, President-Elect

Winona State University (MN)

Gary Bitter, Past-President

Arizona State University

Barry Pitsch, Secretary/Treasurer

Heartland Area Education Agency (IA)

Don Knezek

Educational Services Center, Region 20 (TX)

Jenelle Leonard

Computer Literacy Training Laboratory (DC)

Board Members

Ruthie Blankenbaker

Park Tudor School (IN)

Cyndy Everest-Bouch

Christa McAuliffe Educator (HI)

Sheila Cory

Chapel Hill-Carrboro City Schools (NC)

Susan Friel

University of North Carolina (NC)

Margaret Kelly

California State University-San Marcos

California Technology Project (CA)

Marco Murray-Lasso

Sociedad Mexicana de Computacion en la
Educacion (Mexico)

Paul O'Driscoll

Lane Education Service District (OR)

David Walker

Apple Computer Europe (France)



LOGO EXCHANGE

Volume 10 Number 1

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

Fall 1991

Contents

From the Editor—Something Old and Something New	Sharon Yoder	2
Reflections	Seymour Papert	4
Quarterly Quantum—A Time of Decadence?	Tom Lough	6
LO-GOES Russian	Emily Myers	8
Beginner's Corner—A Is for Apple!11Or Control-A, or Maybe Even r-A!	Dorothy Fitch	11
Linking Logo and Literature	Teri McNee	16
Logo Ideas—Hypertext Revisited	Eadie Adamson	20
Why a Turtle?—Thoughts from Seymour	Seymour Papert	23
Logo LinX—Arboreal Aging	Judi Harris	24
Turtles Invade the Kindergarten Classroom!	Molly Kniffen	28
Math Worlds— ...the heart of mathematical research is tinkering...	A. J. (Sandy) Dawson	30
Windows on Logo—The First Decade of LX	Glen L. Bull and Gina L. Bull	33
Lunch and Logo	Chris Held	39
Extra for Experts—Finding a Place for Logo	Mark Horney	40
Logo: Search and Research—Turtle Talk	Douglas H. Clements	43
A Night in the White House	Janet Koske	48
Remember When....Take a stroll through LX yesteryear		52
Global Logo Comments—Volume 10, Number 1	Dennis Harper	55

Advertising space in *Logo Exchange* is limited. Please contact the Advertising Coordinator for space availability and details.

Logo Exchange (ISSN 0888-6970) is published monthly September through May except for a combined issue for December/January by the International Society for Technology in Education (ISTE), 1787 Agate Street, Eugene, OR 97403-1923, USA; 503/346-4414. Subscription rates: Membership dues for SIGLogo are \$16.00. \$10 of this amount is for the subscription to *Logo Exchange* for one year. This publication was produced using Aldus PageMaker®.

POSTMASTER: Send address changes to *Logo Exchange*, ISTE, 1787 Agate St., Eugene, OR 97403-1923. Second-class postage paid at Eugene OR. USPS #000-554.

ISTE is a nonprofit organization with its main offices housed at the University of Oregon.

Individual ISTE Membership: \$46.00

Dues support the development, coordination, and delivery of ISTE services, including 8 issues of the *ISTE Update* newsletter, either 8 issues of *The Computing Teacher* or 4 issues of the *Journal of Research on Computing in Education*, full voting privileges, and a 10% discount on ISTE books and courseware. Add \$10 for mailing outside the USA. Individual ISTE Members may join SIG Logo for \$16.00. Dues include a subscription to *Logo Exchange*. Add \$10 for mailing outside the USA.

Send membership dues to ISTE. Add \$2.50 for processing if payment does not accompany your dues. VISA, Mastercard, and Discover accepted.

© All papers and programs are copyrighted by ISTE unless otherwise specified. Permission for republication of programs or papers must first be gained from ISTE 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 ISTE.

Something Old and Something New

by Sharon Yoder

A New Format for a New Decade

This issue of the *Logo Exchange* marks the beginning of the 10th year of publication. During those 10 years LX has evolved from an eight-page newsletter produced on a dot matrix printer in Tom Lough's basement to a highly respected, professionally published journal. To celebrate this anniversary, we have made a number of changes in LX. No doubt you have already noticed the fresh, new look. In addition, you have probably noted that your issue of LX is thicker. Although this 10th anniversary issue is extra thick, you will find each future quarterly issue will have more pages than the monthly volumes you received in previous years."

In acknowledgement of the status of the *Logo Exchange* as a respected source of information about Logo for classroom teachers, computer coordinators, teacher trainers, and university faculty, we have decided to make LX a refereed journal. This change will add status to the journal and will increase the quality of the articles you read in future issues. As noted in the May 1991 issue of LX, Judy Kull will be managing the refereeing of articles—which should be sent directly to her. (See the address below.) We continue to invite all of you to share your experiences in working with Logo. I'm sure you will find working with Judy to bring an article to publication a pleasant and rewarding experience.

Where Have We Been?

I am often asked how I got started with Logo. It is quite embarrassing for me to admit that I really don't know. Our school acquired its first microcomputer—an Apple II—in 1979. I used that machine to teach BASIC to middle school students and to develop CAI for use in my math classes. By 1981 we were piloting Apple Logo in a gifted and talented program in one of our elementary schools. Somewhere between 1979 and 1981, a number of things happened:

- I read *Mindstorms*. I can't remember how I found out about it—perhaps someone in a graduate class I was taking mentioned it. But it clearly affected my thinking profoundly.
- A Texas Instruments representative loaned me a TI 99/4A and Logo to try in my classroom. I was fascinated by how quickly my students learned to work with Logo and how engaged they were.
- I purchased the very first version of *Terrapin Logo*. I recall walking into the local computer store with a copy of *Creative Computing* containing the classic Terrapin ad and asking them if they had this product. When they told me they didn't, I asked them if they wanted me to send my \$150 to Terrapin directly or if they wanted to order it for me. They ordered it.
- I struggled through Hal Abelson's book *Logo for the Apple II* recall being fascinated and overwhelmed by all that Logo could do.

All of these experiences came together to bring Logo to the center of the growing computer education program in our school system. Gradually over the next five or six years, the use of Logo grew from use in that first pilot project to use in almost all elementary classrooms. Logo even became a major component of the introductory high school computer course.

Like many of you, I evolved through a wide variety of versions of Logo: *Apple Logo*, *Atari Logo*, *IBM Logo*, *Apple Logo II*, several Macintosh Logos, *LogoWriter*, and more recently *Logo PLUS*, *Logo Express*, and the most recent version of *Object Logo*. Each new version has brought us more features and more power. Logo today is not at all like Logo 10 years ago.

The Old and the New

As you read this issue of LX, you will find a good deal of reflection about the past and glimpses into the future. Begin your exploration with Seymour Papert's wonderful article, which follows this editorial.

As I write this editorial, I have just returned from the National Educational Computing Conference (NECC). The exhibit floor was overwhelming. Every isle was filled with a plethora of powerful sights and sounds. The "hot topics" were clearly multimedia and hypermedia. Oh yes, the Logo vendors were there in force. But it was certainly easy for them to seem lost among the videodiscs, CD-ROMS, flashy graphics, and powerful new software products.

So, what does the future hold for Logo? Where will Logo be 10 years from now? Will it continue to grow and change? If the past 10 years is any indication, Logo will be dramatically different when we enter the next decade. But changes will only happen if you continue to

support the vendors of Logo software and text materials and if you continue to find a place for Logo in your classroom. To paraphrase Tom Lough, FORWARD NEXT.10!

- * Note that you will receive the number of issues that you have previously subscribed to get. That is, if you had a year-long subscription last year, you will get eight issues, but they will be spread over a longer period of time.

Send articles to:

Judy Kull
Department of Education
Morrill Hall
University of New Hampshire
Durham, NH 03824

INTERNATIONAL MEETING ON EDUCATION AND TELECOMMUNICATION TECHNOLOGY
San Jose, Costa Rica
November 7, 1991

V INTERNATIONAL LOGO CONFERENCE
San Jose, Costa Rica
November 4, 5, and 6, 1991

The V International Logo Conference seeks to gather teachers, researchers and Logo fans of all countries in a friendly academic forum for the discussion and exchange of ideas, research, latest developments, and experiences concerning this important learning environment and programming language. This Conference will be open to all levels of experience and research in Logo, ranging from pre-school to university level.

***Submission of papers and presentations—
new deadline: 15 September 1991***

Responding to several requests from potential participants in the Conferences, the Organizing Committee has extended the deadline for submission of papers or presentations to September 15, 1991.

Registration fees

- V International Logo Conference: US\$85:
- International Meeting on Education and Telecommunication Technology: US\$30
- Both conferences: US\$115

***Conference Headquarters:
Hotel San Jose Palacio***

For more information, please contact:

CONFERENCES 1991 ORGANIZING COMMITTEE
FUNDACION OMAR DENGÓ
APARTADO 1032—2050
SAN JOSE, COSTA RICA
CENTRAL AMERICA
FAX # (506) 22-16-54

EMAIL ADDRESSES:

Internet address is:

quesada@huracan.cr (recommended to Internet users)
huracan!quesada@uunet.uu.net (recommended to Bitnet users) (If one address does not work, please try the other one or consult your systems operator.)

Bitnet address is: aarayam@ucrvm2.bitnet



Reflections

by Seymour Papert

Congratulations to *Logo Exchange* on entering its second decade! And one could almost say the same for Logo itself; if you count its birth from the time it was carried into the big world by the advent of little computers, the nineties will be Logo's second decade. But for me it will be the fourth: With a little playfully upbeat projection (appropriate for an anniversary celebration) I see Logo as conceived in the sixties, gestated in the seventies, toddling through the eighties, coming of age in the nineties, and reaching real maturity in the tenties.¹

That's one way to look at it. But I long ago understood that unless you make three theories of everything, you start taking your ideas too seriously. So, in the hope of finding another, I cast around in my mind for metaphors that have served in related contexts, and after a while the following popped into my mind: The sixties is a "classical" period, the seventies a "romantic" one, the eighties is "reactionary," and the nineties will be the "pre-millennium." Hey!—I thought to myself—don't push it too hard (trying to make Logo decades and calendar decades line up exactly is bound to stretch historical literalness—perhaps the Logo decades lag by five years), but there's something right about that idea...it captures some of the "feel" of Logo history and maybe even relates it to other stuff going on at the same time in the larger computer culture and the still larger general culture.

I'll explain myself, but first let me warn you about the third theory: *you* will have to make it. I offer my two attempts in the spirit of Logo-like constructionism: they are not intended to be believed as such but to provoke you to invent your own.² The only Grand Truth I'd like you to take from me is the idea that making up predictions and strategies for the future of Logo is more fun (and more valid—the two do go together) if you look beyond the latest good idea for how to use Logo well (or the latest fad for using something else) and poke around for historical patterns.

The sense in which the early Logo work is "classical" is best brought out by contrast with the "romantic" period that followed. I see it in my own writing about Logo, which in the earliest period was focused on clearly structured issues. The very title of one of my early papers (written in 1969/70) on Logo supports the point: "Teaching Children to Be Mathematicians vs. Teaching Them About Mathematics." In the paper I present Logo as a classical formal system. I argue that Logo and turtle geometry allow children to do some-

thing more like traditional mathematics than school math. I describe some new ways to define a circle and to prove a theorem, but I stay within the framework of defining circles and proving theorems. None of this explicitly challenges the traditional concept either of Mathematics or of School.

Ten years later *Mindstorms* reflects my transition from a classical to a romantic period, and it begins to challenge both School and traditional philosophies of Mathematics (of the thing itself, not merely how it is taught). My use in the book of the Brazilian Samba School as a model learning environment is romantic in the obvious sense of the word, as is the pervasive image of children as epistemologists taking charge of their own intellectual lives. But there is also a deeper sense: Challenging the right of Mathematics to impose its "canonical" epistemology can be seen as a "romantic reaction" in the more sophisticated sense that historians of literature use when they classify Byron or Mary Shelley as participating in a romantic reaction against the neoclassical hegemony of Newtonian super-rationalism.³

The word "romantic" links *Mindstorms* also with the populist excitement about microcomputers, which manifested itself in the computer clubs and the proliferation of computer magazines of the late seventies. The same movement was expressing itself when visionary teachers brought the first microcomputers into their classrooms. Their intention was not a classical education goal of teaching math better (though they may have believed that would happen as well) but a more romantic goal of changing people's relationship to learning and to knowledge. The impression this made on me was dramatic. Even now I can close my eyes and see vividly a 1981 scene in a New York City public school. Two worlds seem to coexist in one room: at one end a teacher is giving a "lesson" at the blackboard; at the other end, a cluster of students are working on their own projects using a pair of TI 99/4s, the first small computers with Logo. The computer group gets into trouble and sends someone to "ask the teacher," who simply says "ask Bill" and continues her lesson without missing a beat, quite unperturbed by the fact that one more student (Bill) has joined those who aren't even pretending to listen to it. The image for me is a tiny foretaste of deep change in the relationships of learning.



By the middle eighties the typical picture was changing fast. For the "romantic" teacher the whole point was to make the computer part of the life of his or her class; to use it as a means to take one more step away from the limitations school imposes on the vocation of teaching. But when the centralized administration of a school system sets up computer labs on a city-wide scale, the motivation (surprise!, surprise!) is not to subvert the structure of school but to bolster it—and nine (or more) times out of ten this is what happens. Creating a separate place for the computer, with a special computer teacher and even a computer curriculum is calculated to thwart the goal of changing the mainstream life of the classroom. This new compartmentalization tends strongly (though, thank goodness, brave teachers, it doesn't always manage) to reaffirm School's balkanization of learning, by which I mean carving knowledge into "subjects," time into "periods," and the learning community into teachers who know and students who don't. Thus the computer, which promised to be the instrument of revolutionary change, is neutralized, and often even becomes a tool of the reactionary forces.

One's projection into the nineties depends on one's theory of the eighties. Were the earlier phases, the classical and the romantic, defeated or only driven underground? In a forthcoming book.⁴ I argue that the period of reaction was an inevitable stage in the development of an educational computer culture and that the old forces are still there. Here I mention only the very simplest corner of the argument. Think of the school sociologically and ask yourself what would be the "natural" way for it to distribute computers. One answer is that it depends on numbers. When there were few computers and few teachers who felt comfortable with such machines, it was "natural" that the teachers who felt comfortable with them could get the machines to integrate into their work. As soon as there were 15 or 20 computers, the pressure to isolate the machines became extremely high. I don't believe that the drive to use computers to change learning subsided. I believe that the conservative forces were temporarily in a position where their viewpoint was hard to resist. But the question is reopened when there are 40 computers and when young new teachers coming into the system have used computers routinely at home and at school. This time we have a better chance to win. And where we don't win this time, we have yet another chance the time after. History is on our side.

And so have a happy anniversary, *Logo Exchange*, and...

Forward Twenty.

Notes:

1. To Decades :number
output word
number - name :number
"ties
end
2. And remember to be recursive. When the fairy offers you three wishes you say: I'll have a white horse (or whatever) and a pile of gold...and three more wishes.
3. On the concept of romantic reaction in the history of computational thinking and on the social movements mentioned in the next paragraph, see Sherry Turkle, *The Second Self* (New York: Simon and Shuster, 1984). On the epistemological challenge, see Turkle and Papert, "Epistemological Pluralism," a paper that has been (or will be) published in somewhat different versions in *Signs: A Journal of Feminist Studies*, Winter, 1990; in *Constructionism*, Harel and Papert, eds. (New Jersey: Ablex, 1991); and in the *Journal of Mathematical Behavior*.
4. To be published by Basic Books in 1992.



A Time of Decadence?

by Tom Lough

Welcome to the new quarterly LX!

Several of my colleagues pointed out that the former title of this column, "Monthly Musing," just wouldn't cut it any more. So, welcome to the new QQ column! Each quarter, I plan to inflict upon you, dear LX readers, a quantum (that is, a fixed amount) of wordage. I hope you will find it helpful, interesting, useful, provoking, entertaining, or some of the above.

Logo Exchange is decadent! How about that for a lead sentence? Now, before you leap to the phones (there are no operators standing by, anyhow), allow me to explain what I mean.

LX is now entering its 10th year of continuous publication! For any periodical, especially one in the field of educational computing, this is a major accomplishment. I looked around for a suitable adjective to denote this ten-ness, but could find none. So, taking advantage of the malleability of the English language, I simply ascribed an additional meaning to "decadent," to wit, "adj., of, pertaining to, or having accomplished 10 years of existence or being."

I know this is not really proper, especially since the Latin Vulgar root for "decadent" is not clearly related to the Greek root for "decade." But the connection was so inviting that I took the liberty. At any rate, I hope it has focused your atTENTion on the TEN. Now let's focus on what to do about it.

AtTENTion to Action

Our collective continued professional activity helps Logo remain an option for students. Celebrate with LX by considering an action area for professional focus. Here are some suggestions.

Assessment

Dan and Molly Lynn Watt have emerged as pioneers in the area of Logo-related assessment. They have developed exciting projects under the widely respected sponsorship of the National Science Foundation, and have made presentations at several major conferences. They also wrote a series of columns on assessing Logo learning which appeared in Volume 7 (1988-1989) of LX. After you review their columns, you may wish to contact them for additional information. Write to Dan and

Molly Lynn Watt, Educational Alternatives, Gregg Lake Road, Antrim, NH 03440.

Mathematics

Stay current on Logo-related mathematics education research with Douglas Clements' excellent "Search and Research" column. If you are a member of the National Council of Teachers of Mathematics (NCTM), then you should also be a member of the Council for Logo In Mathematics Education (CLIME). In addition to publishing a newsletter and a series of disk-based mathematics microworlds, this organization is becoming more active in bringing Logo-related presentations to the attention of the NCTM membership. For more information, write to CLIME, 10 Bogert Avenue, White Plains, NY 10606.

Development

One of the reasons Logo continues to flourish is because of the dedicated support of the software companies that produce Logo versions. Moreover, they are continually upgrading their products and developing new ones. But this support must be two-way. Support Logo Computer Systems, Inc., Terrapin, Inc., and Paradigm Software by following the letter and the spirit of the license agreements on your software packages, by encouraging upgrades and purchasing them when available, and by communicating to the companies your comments, concerns, and suggestions for future products or Logo versions.

Conferences

Although conferences dedicated solely to Logo are few and far between these days, there are still plenty of opportunities for Logo users to make presentations at other conferences. Science, mathematics, music, language arts, and many other subject areas offer an inviting array of possibilities.

User Groups

Don't we have enough user groups already? Perhaps. But there are at least two areas that might be capable of supporting specialized Logo user groups, in addition to mathematics (see above). Science teachers, what about you? Who will found ALISE (Association for Logo In Science Educa-

tion)? Perhaps a group such as this could sponsor a special strand of Logo-related presentations at the conference of the National Science Teachers Association, for example. Educational robotics is a rapidly growing field, and certainly includes Logo (but not exclusively). An informal group of interested educational robotics professionals has gathered for discussions at the last three NECC meetings. When will the formation of the AER (Association for Educational Robotics) be announced?

As we close out the first decade of *LX*, let us resolve to seek for even more opportunities to help our teaching colleagues discover and learn the power and potential of Logo. Our nation's students deserve no less than this.

FD 100!

Tom Lough
Founding Editor
PO Box 394
Simsbury, CT 06070

NECC '92

Call for Participation

National Educational Computing Conference

June 15-17, 1992 Dallas, Texas

Looking through the windows of the world...

Those who would like to participate in NECC '92 are invited to submit original papers, and/or proposals for project reports or preconference workshops. Individuals representing all academic disciplines and all facets of education are welcome to submit.

To obtain a *Call for Participation* brochure (which describes the format in which proposals must be submitted), contact:

NECC '92

International Society for Technology in Education
1787 Agate Street

Eugene, OR 97403-1923

ph. 503/346-4414 FAX 503/346-5890

Submission deadline for all papers, project proposals, and preconference workshop proposals is October 1, 1991. NECC '92 will be held in conjunction with the Texas Computer Education Association's annual conference.

A First Course in Programming in Terrapin Logo, LogoWriter, and PC Logo

This is a complete curriculum for a semester course in programming. It includes student activity sheets, teacher lesson preparation sheets, tests, quizzes, assignments, and sample solutions for all student assignments (hard and softcopy!)

A First Course in Programming is a directed learning environment in structured programming. Its 450 pages emphasize problem solving strategies, critical thinking skills and solid principles of computer science.

Only \$150 for a building site license. Call us for further information!

Curriculum written BY teachers FOR teachers!

Logo Curriculum Publishers
4122 Edwinstowe Avenue
Colorado Springs, CO 80907
1-800-348-5646 (FIT LOGO)

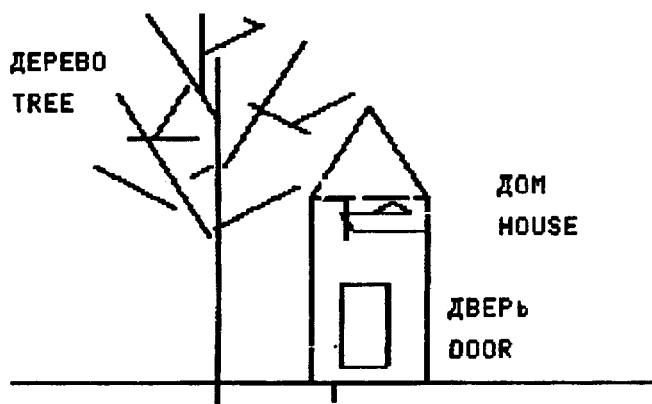


LO-GOES Russian

by Emily Myers

At Albert Einstein Academy in Wilmington Delaware, we are drawing, writing, and making music with Logo. Several of our students have recently emigrated from the Soviet Union. For writing in English, *Childrens' Writing and Publishing* works well, but for our bilingual requirement, Logo is wonderfully helpful.

Logo graphics are, of course, international. Two Russian children often work together on the same computer. Because I don't speak Russian, I resort to pointing and key words. Usually one face lights up and explosively explains to the other, in Russian, what needs to be done. I am left out of the ensuing conversation that goes back and forth. The picture by Margarita Golod show the results of one such exchange.



Writing in Russian

For writing, we needed a bilingual computer that could form both the English and Russian alphabet. Armed with the ability to create fonts in Logo PLUS, we were able to label pictures using both English and Russian characters. The list below shows a simple Russian/English correspondence list.

КОТ	CAT
ЩЕНОК	PUPPY
ДЕВОЧКА	GIRL
МАЛЬЧИК	BOY
ЛЮБЛЮ	LOVE
УЧИТЕЛЬНИЦА	TEACHER
ШКОЛА	SCHOOL
ГОРОД	CITY
КОМПЬЮТОР	COMPUTER

Students use the following chart to find the Cyrillic characters they need to use:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 АБВГДЕФГHIЯЕКЛМНОПЖУСТУВШХЮЗ
 абвгдефгHIяеклмнопжрстуvwшхюз

! @ # \$ % ^ & * () _ + [] { } ; ' : " , . / < > ?
 ! Ъ Ы Я & Щ () _ Э [Я { } Э ' Щ " , . / Ц Ц ?

Instant Logo

To make the process easier, we use a single-key program (F is FORWARD 10, R is RIGHT 30 degrees) adapted from Terrapin's Instant Logo. You can even make shapes by entering a single key. The numbers 7, 8, 9, and 0 each cause a circle of increasing size to be drawn. For triangles of increasing size, Q, W, and E are used. This helps children who may have used the MECC EZ Logo program at home or elsewhere feel more comfortable with our system.

To make use of the Cyrillic alphabet, a further modification of our Instant program was needed. We had exhausted meaningful mnemonic possibilities, so we selected a comma (,) as the key to choose the Cyrillic font used to write Russian words. After the font is loaded, the program automatically enters GWRITE mode, so text can be entered right away at any place on the screen.

Pressing Esc leaves the GWRITE mode and returns the student to single-key drawing. When a return trip to the English alphabet is required, a period (.) loads the standard ASCII font and puts the student in GWRITE mode. Drawing and text of either character set can be used in any order as many times as required. The picture of the house above was our first trial of the program.

How Does It Work?

The Instant program reads a character and then executes specified Logo commands as shown below. If the variable :COM equals a comma, then it reads the Cyrillic font and enters GWRITE mode. STOP ends the command and the INSTANT procedure is reentered recursively.

```
IF :COM = "," READFONT "CYRILLIC
    GWRITE STOP
IF :COM = "." READFONT "ASCII GWRITE
    STOP
```

Using "P" to start printing was a problem. When computers are sharing a printer with a switch and the switch is not set to print, the screen freezes until it thinks the screen has finished printing. If the student doesn't know what key was pushed, the event is puzzling. The print command was made an uppercase character ">" so that it wouldn't accidentally be pressed.

A form feed was added to the print routine to eliminate the nuisance of having to press the Select button, then the Form Feed button, and the Select button again to get the paper out of the printer. The paper is put in with the top edge at the edge of the cover

of the printer. For an Imagewriter, the print command line is as shown below:

```
IF :COM = ">" PRINTSCREEN 1 2 OUTDEV
1 CHAR 12 OUTDEV 0 STOP
```

More Modifications

The original Instant program processes the input character using a procedure named RUN.AND.RECORD, which keeps a running history of all commands entered. This history can then be used to repeat a figure or even to redraw a picture without the last step, effectively allowing an "undo" command. The history and undo are too slow for our students, so these features were eliminated. Undo was replaced with a BACKUP and an ERASE procedure that works more quickly.

Creating Music

The history is useful, however, when creating music. We have included a DRAW.SING procedure that turns the computer keyboard into a musical instrument. The keys A, S, D, F, and G correspond to the sounds DO, RE, ME, FA, and SO. Just as in the previously described procedure, the variable COM is read. Three actions result each time a key is read:

- A tone is played.
- History is recorded.
- A note is shown on the screen.

For example, if the letter A is entered, the note "middle C" plays. Next, the note is added last (LPUT) on a history list. A note shape is located on the screen, advancing 10 on the X-axis and moving up and down on the Y-axis varying with the pitch. The note stamped on the screen is rumored to be a certain well-known turtle in disguise. The "real" turtle must be hiding to produce the desired effect.

```
IF :COM = "A" TUNE [1] STOP
```

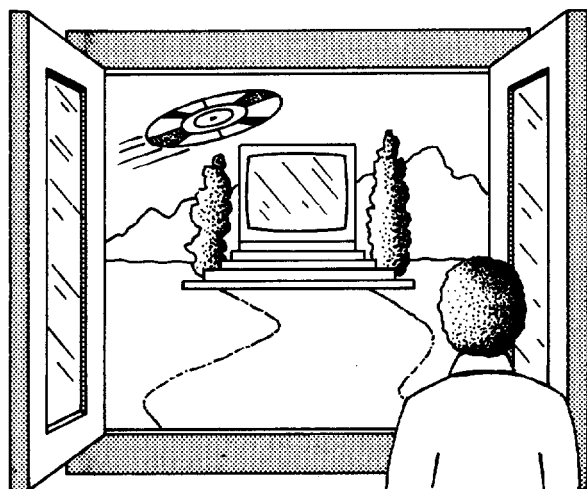
```
TO TUNE :NOTE
SING :NOTE
MAKE "HISTORY (LPUT FIRST :NOTE
:HISTORY)
MAKE "Y FIRST :NOTE
SETXY XCOR+10 :Y*8
STAMP
END
```

When the letter P is entered, the notes in the history list are played using the Terrapin procedure SING.

```
IF :COM = "P" SING :HISTORY STOP
```



A Window on the Past.



A Vision of the Future.

Imagine having the past four years of educational computing research at your fingertips. Plan your future based on the lessons of the past.

After four years of reviewing research for her column in *The Computing Teacher*, Betty Collis identifies key trends and issues of vital interest to all computer using educators. As an additional resource, she has compiled an extensive bibliography to aid readers in further investigation.

The Best of Research Windows: Trends and Issues in Educational Computing—a window of opportunity for educators at all levels.
\$12.95 plus \$3.25 shipping.

To order, contact:
ISTE, 1787 Agate St., Eugene, OR 97403-1923;
ph. 503/346-4414

For example, the :HISTORY of DO RE ME is [1 2 3]. When C is entered, the screen is cleared and the history list is emptied.

```
IF :COM = "C" MAKE "HISTORY []
CLEARSCREEN STOP
```

Although our emphasis is on teaching English, it was gratifying to be able to provide our new friends with their own home alphabet. Once again, that versatile turtle writes, draws, and sings.

Emily R. Myers teaches and provides consulting services for several schools in Delaware and Pennsylvania. She founded Decision Design Research, Inc., a firm involved in research on the ways computers can help people make decisions. Her first book, *Automated Financial Systems*, was published by McGraw Hill. She is working on her second book, which will describe six easy yet exotic Logo projects. She has written many articles about computers, this being the first in the field of education.

Emily R. Myers
Decision Design Research, Inc.
P.O. Box 113
Chadds Ford, PA 19317

A Is for Apple! Or Control-A, or Maybe Even r-A!

by Dorothy Fitch

Welcome (or welcome back) to Beginner's Corner, where we take a look at activities and projects for beginning Logo users. This time, we'll develop some programs that need just a press of a key to make something interesting happen.

Single keystroke programs have been around as long as LX! (Happy 10th anniversary to LX!). You may be familiar with the Instant program that comes with many Logo packages or similar implementations that use single keystrokes for commands. You can press "F" to move the turtle forward a short distance, "R" to turn it right, and so forth. These programs generally come with a limited set of single-key commands, but you can modify them quite easily.

Single-keystroke Logo programs are handy for use with young children, the physically handicapped, or the general learner in many situations. These programs require little or no reading ability, and with colored markers placed strategically on the "special" keys, children don't even need to be able to recognize and match letters.

This column will help you design a single keystroke program and customize it to do whatever you want. It will also show you how to access some of the non-alphanumeric keys (Esc, Arrow keys, Open-Apple, for example). The procedures are written using Terrapin's versions of Logo, although almost all of what is shown here can be done using any version.

The first part of this column can be understood and used by beginning Logo users. It assumes that you know how to define a Logo procedure (refer to your Logo documentation if you don't), and that you can use the programs here as models for your own programs. The second part tells how to use Control, Open-Apple, and other special keys in your programs. The concept of using these keys is the same as using letter keys, but you detect them in different ways. These ideas are included in the Beginner's Corner column to help you develop useful Logo programs for your beginners. Finally, there are ideas for some single keystroke programs you may want to explore.

Try typing these procedures as we go along to see firsthand how the program develops!

The Main Program

First we need a procedure that waits for a keystroke, then sends it off to be processed:

```
TO LOOP
COMMAND RC
LOOP
END
```

The command RC tells Logo to wait until you press a key. The key you press becomes the input to a procedure called COMMAND, which we'll write in a moment. Notice that the LOOP procedure is recursive; that is, its last instruction is a call to itself (or more precisely, to a copy of itself). This lets us keep pressing keys for different actions.

If you try to run the LOOP procedure, Logo immediately tells you that it doesn't know about the procedure COMMAND. So, let's define the following procedure (for your version of Logo, you may need to use the following IF syntax: IF :KEY = "F [FORWARD 10]):

```
TO COMMAND :KEY
IF :KEY = "F FORWARD 10
IF :KEY = "R RIGHT 30
END
```

Now, after typing LOOP, you can press F and R to move and turn the turtle. If you press any other key, nothing happens. (If nothing happens with F or R, press down the Caps Lock key.)

You can actually draw quite a nice design using just F and R, but you'll soon want to add more options! To stop the program, press Control-G or the equivalent "stop" command for your Logo.

You can add more keys using the same command format. The program will respond more quickly to your keystrokes if you add STOP to the end of each line (except for the last line, since the procedure will stop then anyway!). The STOP command keeps Logo from having to process lines for keys that haven't been pressed. Also, if commands for the most common keystrokes are at the beginning of the procedure, Logo will find them first and your program will react faster.

Here is a COMMAND procedure that includes many additional keystroke commands (the LX! picture below was drawn using this procedure):

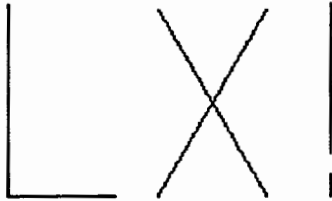
```
TO COMMAND :KEY
IF :KEY = "F FORWARD STOP
IF :KEY = "R RIGHT 30 STOP
```



```

IF :KEY = "L LEFT 30 STOP
IF :KEY = "B BACK 10 STOP
IF :KEY = "U PENUP STOP
IF :KEY = "D PENDOWN STOP
IF :KEY = "C CLEARSCREEN STOP
IF :KEY = "H HOME
END

```



You may want a procedure named **START** to set up the screen for you and begin the process:

```

TO START
SHOWTURTLE
HOME
CLEARSCREEN
LOOP
END

```

Adding Your Own Procedures

You can also include commands to run procedures you have written. Here are two lines that you could add to **COMMAND**, and the **SQUARE** and **JUMP** procedures that make them work:

```

IF :KEY = "S SQUARE STOP
IF :KEY = "J JUMP STOP

TO SQUARE
REPEAT 4 [FORWARD 30 RIGHT 90]
END

TO JUMP
PENUP
REPEAT 5 [FORWARD 10 WAIT 3 BACK 10
  WAIT 3]
PENDOWN
END

```

If your Logo doesn't have a built-in **WAIT** command, define the following procedure. You can change the amount of delay by making the number 50 larger or smaller.

```

TO WAIT :NUMBER
REPEAT :NUMBER * 50 [ ]
END

```

Looking for Special Keys

You may want to provide a more elegant way to stop your program. Many programs use the Esc key to end or quit. You can make Logo look for this key in the **COMMAND** procedure. But you just can't say:

```
IF :KEY = "ESC STOP
```

The way to look for the Esc key is to use its ASCII code. Every character you type has a number code assigned to it, called the American Standard Code for Information Interchange. These codes are the same from computer to computer (with a few exceptions, which we'll address later).

The ASCII or character code for the Esc key is 27. We can use this number with Logo's **CHAR** command to detect this key:

```
IF :KEY = CHAR 27 TOPLEVEL
```

Add this line to the **COMMAND** procedure. To stop the program, you can now press Esc. Rather than using **STOP** at the end of this line, we need to use **TOPLEVEL**. **STOP** would make the **COMMAND** procedure stop, but the **LOOP** procedure would continue, waiting for another keystroke. **TOPLEVEL** tells Logo to stop all procedures and give control of Logo back to us at **toplevel**, where we can type commands. (In LogoWriter, the equivalent command is **STOPALL**.)

The following procedure will tell you the ASCII code of any keystroke. (Type **ASC** to begin; stop it with **Control-G**.)

```

TO ASC
PRINT ASCII RC
ASC
END

```

You can find out the ASCII code for every letter (uppercase and lowercase letters have different ASCII codes), number, and punctuation mark. If you hold down the Control key and press a letter, you will find the ASCII codes for that combination keystroke. The Arrow keys, Tab, Delete, Return, and space bar each have a unique ASCII code. The only keys that don't have an ASCII code are Shift, Control, Caps Lock, the Apple keys and Option. There is another way to access some of these keys, described below.

Here are the ASCII codes for special keys you may want to use in your programs:

Keystroke	Apple	Macintosh
Left arrow	8	28
Right arrow	21	29
Up arrow	11	30
Down arrow	10	31
Control-A	1	1
Space bar	32	32
Return	13	13
Delete	127	8

Some keystrokes may be reserved for use by Logo. For example, Logo *PLUS* uses Control-G (CHAR 7) to stop a program. You can't give a special function to Control-G in a program you write; as soon as you press Control-G, your program will stop! Logo's use of this keystroke will override any function you give it.

Looking for Apple Keys on the Apple Computer

Although the Open and Closed Apple keys on the Apple computer do not have an ASCII code, there is a way to detect whether or not they are pressed. Your Apple computer keyboard has an Open-Apple key (r) and will have either a Closed Apple key (t) or an Option key. The Option key is exactly the same as the Closed Apple key; it just has a different name!

The Apple keys actually function as game paddle buttons. (If you have a game that requires that you press a game paddle button, you can use your Apple keys!) Logo has a command to look at a game paddle button and tell you if it is pressed.

Type the command below. Logo should respond FALSE:

```
PRINT PADDLEBUTTON 0
```

Now type the same command again, but this time, hold down the Open-Apple key as you press Return. Logo should respond TRUE.

Type this next command. Logo should respond FALSE.

```
PRINT PADDLEBUTTON 1
```

Type it again, but hold down the Closed-Apple key or Option key (whichever key your computer has) as you press the Return key. What did Logo respond this time?

How can you use Apple keys in your program? You could, for example, include a command to clear the screen when you press the Open-Apple key. It would look like this:

```
IF PADDLEBUTTON 0 CLEARSCREEN STOP
```

(You could type `IF PADDLEBUTTON 0 = "TRUE...` but it isn't necessary to type all that because `PADDLEBUTTON 0` automatically reports "TRUE" or "FALSE, which is what IF needs as an input.)

You can also use the Open-Apple key with a letter key. You may want to reserve this combination for special functions, such as printing or saving. To tell if the Open-Apple key is pressed *and* the letter P is also pressed, you need to use the Logo command `ALLOF` (or `AND`, in some versions of Logo), like this:

```
IF ALLOF PADDLEBUTTON 0 :KEY = "P
  PRINTSCREEN 1 2 STOP
```

With this command, when you hold down Open-Apple and press P, Logo *PLUS* will print your picture. It is wise to test this command first by printing text to the screen before you actually print to the printer. Why not first try:

```
IF ALLOF PADDLEBUTTON 0 :KEY = "P
  PRINT [Printing...] STOP
```

Likewise, you could include commands that look for the Closed-Apple (Option) key by using `PADDLEBUTTON 1`.

Special Keys on the Macintosh Computer

The Macintosh computer works somewhat differently from the Apple computer when it comes to the Open-Apple (Command) and Option keys. These keys are not read as paddle buttons.

In addition to ASCII codes for different characters, the Macintosh computer has an internal code assigned to each key. Since the code is linked to a specific physical key, not the character on the key, two symbols may have the same key code. For example, both "A" and "a" have the same Macintosh key code. The characters 1 and ! share the same key, so they have the same key code.

Terrapin Logo for the Macintosh has a primitive called `KEYSDOWN` that tells you the key code numbers of the keys currently being pressed. To find the key code for any key, use this procedure:

```
TO KD
  PRINT KEYSDOWN
  KD
END
```

Type KD to start, then press any key to see its key code number. You will find that the Command, Option, Shift, Control, and Caps Lock keys each have a key code. Since these keys do not have an ASCII code, the `KEYSDOWN` command gives you a way to detect



them. It is also interesting to note that the keys on the numeric keypad have different key codes from their regular keyboard counterparts (the * on the numeric keypad is key code 67; the * on the 8 key is key code 28).

Here is a list of some of the Macintosh key codes you may want to use.

Key code	Macintosh
Command ⌘	55
Shift	56
Caps Lock	57
Option	58
Control	59

To include the possibility of having either a Command-key keystroke or a simple letter keystroke, we must modify the LOOP procedure for the Macintosh in this way:

```
TO LOOP
IF MEMBER? 55 KEYSOWN [COMMAND.KEY
KEYSOWN]
IF RC? [COMMAND RC]
LOOP
END
```

If the Command key (key code 55) is pressed during the LOOP procedure, then the COMMAND.KEY procedure will be run. It takes the list of keys that are down as input. This version of COMMAND.KEY looks for the letter P (key code 35) in the list of keys that are pressed. If it is found, then Logo will print the active graphics window.

```
TO COMMAND.KEY :KEYS
IF MEMBER? 35 :KEYS [PRINTWINDOW
TWINDOW]
END
```

Note that the ASCII code for the uppercase letter P is 80, but its key code is 35. After checking the KEYSOWN list for the Command key, we can't then switch to looking for an ASCII code or the letter P; we must continue to use key codes to detect other keys. In other words, you can't combine a command to use a key code together with a letter key the way you can combine a paddle button reading and a letter key on the Apple computer.

Single Keystroke Program Ideas

Of course, you don't have to get fancy when you write single-keystroke programs. Some of the most useful single-keystroke programs (like the very first

example) simply use letter keys to move and turn the turtle. Here are some other ideas for single-keystroke programs.

Directions:

These procedures use N, S, E, and W to move the turtle in different directions on the screen. They can help students practice map skills. Can they move the turtle to a target on the screen using directional (N, S, E, and W) commands?

```
TO START
HOME
CLEARSCREEN
PENUP
LOOP
END
```

```
TO LOOP
COMMAND RC
LOOP
END
```

```
TO COMMAND :KEY
IF :KEY = "W SETX XCOR - 10
IF :KEY = "E SETX XCOR + 10
IF :KEY = "N SETY YCOR + 10
IF :KEY = "S SETY YCOR - 10
IF :KEY = CHAR 27 PENDOWN TOPLEVEL
END
```

Turtle moves:

This version of the COMMAND procedure above moves the turtle using the Arrow keys on the Apple computer. If you are using the Macintosh computer, be sure to use the correct ASCII codes (see previous chart).

```
TO COMMAND :KEY
IF :KEY = CHAR 21 SETX XCOR + 10
IF :KEY = CHAR 8 SETX XCOR - 10
IF :KEY = CHAR 10 SETY YCOR - 10
IF :KEY = CHAR 11 SETY YCOR + 10
END
```

A choice of turns:

In this procedure, the keystroke R turns the turtle a large amount, 30°. The keystroke Control-R (ASCII code 18) turns the turtle 15° at a time, allowing for a heading of 45°. An Open-Apple R keystroke turns the turtle just 1°! It is important to place the check for Open-Apple R first; otherwise, since :KEY would be "R, Logo would detect the R keystroke too soon and turn the turtle Right 30, which isn't what you had in mind. It is also critical to use STOP at the end of each line so that the turtle doesn't first turn 1° and then 30°.



```

TO COMMAND :KEY
IF ALLOF PADDLEBUTTON 0 :KEY = "R
  RIGHT 1 STOP
IF :KEY = "R RIGHT 30 STOP
IF :KEY = CHAR 18 RIGHT 15
END

```

Picture book:

An idea from the title of this column would be to have a series of procedures available to draw a picture for each letter of the alphabet. Each of these lines will cause Logo to draw a picture of an apple (if you have a procedure called APPLE that does this) by pressing A, Control-A, or Open-Apple A on the Apple computer. Can you tell which line is which and see how each one works?

```

IF :KEY = CHAR 1 APPLE
IF ALLOF PADDLEBUTTON 0 :KEY = "A
  APPLE
IF :KEY = "A APPLE

```

Happy Logo adventures!

Dorothy Fitch has been Director of Product Development at Terrapin since 1987. A former music educator, she has also directed a computer education classroom for teachers and students and provided inservice training and curriculum development for schools. She is the author of *Logo Data Toolkit* and coauthor of *Kinderlogo*, a single keystroke Logo curriculum for young learners. At Terrapin, she coordinates software development, edits curriculum materials, writes documentation, and presents sessions at regional and national conferences.

Dorothy Fitch
Terrapin Software, Inc.
400 Riverside Street
Portland, ME 04103

CompuServe: 71760,366
Internet: 71760,366@COMPUSERVE.COM
(207) 878-8200

Introducing...

Educational IRM

Quarterly, a new publication for information resource managers. ISTE's newest journal recognizes that educational information resource managers stand in a unique position between school administrators, teachers, and students.

If you are involved in making technology-related decisions, you'll find the current and unbiased information you need in ***Educational IRM Quarterly***.

***Guaranteed to inform,
entertain, and inspire.***



*Call or write ISTE for additional details
and a free sample copy.*

ISTE, 1787 Agate St.
Eugene, OR 97403
503/346-4414

educational
IRM
QUARTERLY

premieres October 1991



Linking Logo and Literature

by Teri McNee

Integrating technology into the curriculum is not always an easy task. At the end of each year, as a means of formative evaluation, I ask my students to select the one piece of software they couldn't do without. Almost unanimously the first and second graders respond, "the turtle." Being able to program the Logo turtle to carry out a simple sequence of commands gives children of this age a tremendous sense of power and creativity. However, despite rave reviews from the children, educators are constantly called upon to justify Logo in terms of "curriculum relevancy." Logo's lifespan in the classroom often depends on innovative applications.

Literature offers a rather unique forum for Logo explorations. Why not use Logo REPEAT commands to create "webs" while reading E.B. White's *Charlotte's Web*? Or make Logo patchwork quilts to display at a

classroom County Fair attended by Wilbur and Charlotte! Sylvester, in William Steig's *Sylvester and the Magic Pebble*, was interested in pebbles and stones of many shapes. Use your Logo commands to draw pebbles of various shapes and sizes. Let your students discover how many pebbles can fit across the screen. *Frog and Toad Are Friends*, by Arnold Lobel, spins a wonderful tale of friendship between a frog and a toad. Use turtle graphics to draw a diamond from two triangles. Add a tail, and you have a kite for Frog and Toad to fly at the park! While studying Beatrix Potter's *The Tale of Peter Rabbit* you can have your students take the turtle on an imaginary trip through Mr. McGregor's garden by using a transparent overlay maze on the computer monitor. Perhaps the following lesson will act as a springboard to other Logo/literature connections!

Address for *Easy Art*

Easy Art, Teacher Created Materials, P. O. Box 301,
Sunset Beach, CA 90742.

The "Amazing Gardens" lesson described in this article is part of a larger unit, "Cotton Capers: Adventures with Peter Rabbit." This lesson plan won second place in the 1989-90 Computer Learning Month Contest, and was reproduced with the permission of Computer Learning Month.

Teri McNee
Shetland Place
Westland Village, CA 91362



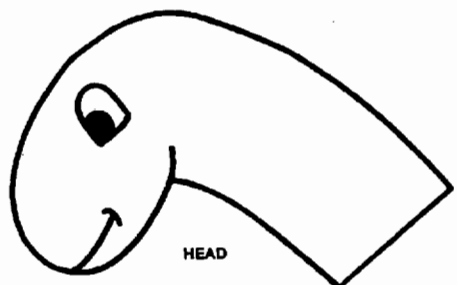
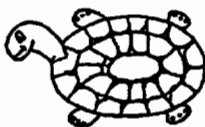
Amazing Gardens

Anticipatory Set

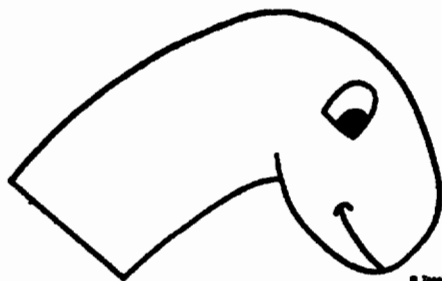
Read *The Tale of Peter Rabbit* by Beatrix Potter. The class will make Logo turtle hats using paper plates, green paint, and a ditto of the turtle head, tail, and legs. (Full-sized sheets for making turtle hats are available in *Easy Art*. See the end of this article for the address.)

PAPER PLATE TURTLE

1. Color paper plate.
2. Reproduce all body parts on green construction paper.
3. Cut out heads and glue together. Make a slit in the paper plate and attach head with glue as shown in diagram.
4. Attach feet and tail with glue as shown in diagram.

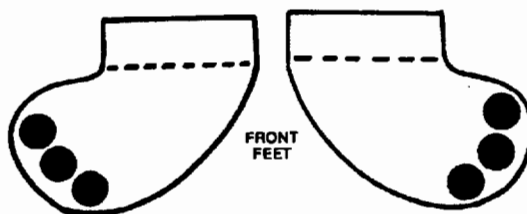


Run on green construction paper.

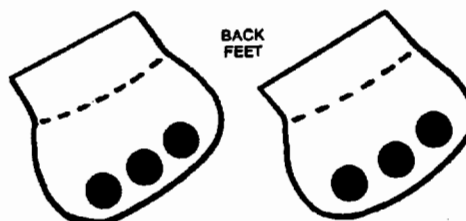


© Teacher Created Materials

PAPER PLATE TURTLE



FRONT FEET



BACK FEET



TAIL

18

Set up a replica of the transparent maze overlay in the classroom using chairs for "veggies" and the teacher's desk for Hilltop Farm. (See the picture showing the content of the overhead at the end of this article.)

Instructional Objectives

The student will guide the turtle through a Logo maze by changing compass directions into Logo commands.

Input

The teacher will place the maze of Mr. McGregor's garden over the computer monitors. Laminated task cards with directions should be placed at each computer station. A large compass should be prominently displayed.

Model

Teacher and students wear their Logo turtle hats and walk through the classroom maze of Mr. McGregor's garden. Students will take turns giving the commands (FORWARD 5 turtle steps, etc.). Students will quickly review DRAW, FORWARD, RIGHT 90, LEFT 90, BACK, HIDE TURTLE, and SHOW TURTLE.



Check for Understanding

The teacher will observe and question. The teacher can ask the students the following questions: Which way is west (RIGHT 90 or LEFT 90)? How many turtle steps do you think it will take to get to the end of the first row of veggies? Which way will you turn to go up to the top row? How far do you turn to make nice square corners?

Guided Practice

Using a large screen monitor, the teacher will then call on individual students to begin to direct the turtle around the vegetable patch. Begin with LEFT 90 to head the turtle west. Remember to use the compass for help. Continue going FORWARD until the turtle is at the end of the row of cabbages. Read the directions out loud together. Point to the watering can and to Hilltop Farm as you read. Assign the student "turtles" to computers in pairs.

Independent Practice

The student will direct the turtle:

```
LEFT 90
FORWARD <to the end of the row of cabbages>
RIGHT 90
FORWARD <to the top row of veggies>
RIGHT 90
FORWARD <past the veggies and Mr. McGregor>
RIGHT 90
FORWARD <to Peter in the watering can>
LEFT 90
FORWARD 30
LEFT 90
FORWARD <until you reach Hilltop Farm>
```

Exact inputs are not given for FORWARD commands because students will discover that there isn't only one correct way to follow the maze. Allow time for exploration. Which students used the fewest commands? Which students took lots of little turtle steps? Which students took just a few giant turtle steps? Can you add up all the FORWARD inputs to see how many steps the turtle took in all?

Extension

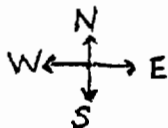
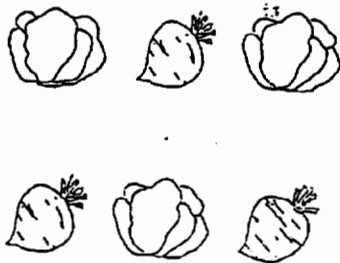
Using a ditto copy of the transparent maze, the students can record the path the turtle took to go through the maze, write the commands to reflect the path they followed, and then color the picture. Students might also like to draw a new "home" for Peter Rabbit by using basic primitive commands.



AMAZING GARDENS TASK CARD

CHANGE THESE DIRECTIONS INTO LOGO COMMANDS

1. Turn west.
2. Go around the outside of Mr. McGregor's vegetable patch.
3. Don't step on the vegetables! Remember what happened to Peter's father!
4. Find Peter in the watering can.
5. Turn east.
6. Go north to Hilltop Farm.



Hypertext Revisited

by Eadie Adamson

If you use any form of writing tool in your daily existence, you're familiar—consciously or unconsciously—with the concept of hypertext. If you scribble notes on a napkin at lunch, you use hypertext. If you draw diagrams to represent ideas or make outlines for presentations, you'll be on familiar ground with hypertext. If you see underlying patterns and connections in the work you do, hypermedia could prove to be a very empowering tool for you. If you find yourself, from time to time, daydreaming or thinking in a nonlinear manner, worry not. You have the earmark of a fine hypermediacian.

Michael Fraese

Macintosh Hypermedia, Volume 1

Unrestricted by sequence, in hypertext we may create new forms of writing which better reflect the structure of what we are writing about, and readers, choosing a pathway, may follow their interests or current line of thought in a way heretofore considered impossible.

Ted Nelson

Literary Machines

Over the past 10 years, versions of Logo have changed at a steady rate. Who would have dreamed in 1981 that LEGO objects would be controlled by a version of Logo? Who would have thought of using scanned images in Logo programs? Of controlling videodiscs—did we even have them then?—with Logo? Of using a version of Logo for telecommunications? Or of creating hypermedia documents with *LogoWriter*? Who would have thought of these things, indeed! Over the last 10 years, many of us Logophiles have become more and more sophisticated in our vision of what a Logo environment might be as well as our in vision of what we might do with Logo.

In this column, I want to reflect on a Logo project I began working on a little over three years ago: hypertext with *LogoWriter*. Even in that short space of time, my own vision of Logo hypertext and its uses has changed dramatically. I first began working on *LogoWriter* hypertext as an interesting context in which to explore *LogoWriter*'s word-processing primitives. My original idea was to create a hypertext program for students who were nonprogrammers. The Logophiles among them could add graphics and animation if they wished.

Today's *LogoWriter Hypertext* has come a long way from that first vision. It can include digitized color images, recorded sound (on the new Macintosh computers), and videodisc sequences. In the future I would expect Logo hypertext might operate LEGO TC logo machines and other external sensing and recording devices as well.

Hypertext is a nonlinear, or nonsequential, method of delivering information. Words become triggers that lead the reader off to other pieces of information. Clicking on or selecting a word accesses information related to that word. The reader becomes the writer, developing his or her own sequence, moving about and backtracking at will. Hypermedia is hypertext that includes sound, pictures, or video delivered in a sequence determined by the reader. Multimedia is a term often misused these days. Multimedia is not usually hypertext because multimedia presentations are usually just that: linear in nature, with one point following upon the other in a straight sequence from beginning to end and allowing no deviation from the single path.

Here's some of what I said in May 1989 about the potential for Logo hypertext:

Hypertext in *LogoWriter*

Hypertext in its most recent application, *HyperCard*, the cause of the excitement behind the articles you may have read, runs only on a Macintosh computer. For many schools a Macintosh is only a dream, not a reality. CD-ROM and videodisc players are still further removed, yet there are many demonstrations using computers and hypertext to drive these systems. The idea of hypertext doesn't need to remain a dream, however. There are ways in which you too can show your students a little about hypertext—by using *LogoWriter*!!

A Bit of History

The idea of hypertext is not new. In 1945 in an *Atlantic Monthly* article, "As We May Think," Vannevar Bush (1986) 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 1960s.) Bush even spoke of buttons and pointers, envisioning a kind of workstation that would contain all this information on microfilm and that could be organized accord-

ing 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.

A recent novel, *The Dictionary of the Khazars*, by Milorad Pavic (1988), uses an approach to storytelling that is remarkably close to the idea of hypertext. There are three sections to the novel, which 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 you just left. You invent your own continuity as you read along.

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. I wanted my students to be able to experience this radically different way (to some) of thinking about text, in particular. 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 databases, all of which are familiar to us and to our students. *LogoWriter's* page metaphor puts us quite naturally into this manner of thinking about the pages (files) stored on our disks. A *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 and write a procedure that 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.

Back to the Present

By 1989 I had already been playing around with my own version of hypertext with *LogoWriter*. Michael Tempel had also developed a hypertext program, using a slightly different approach. We both had collected samples of student work that we shared at numerous conferences in joint presentations.

While my students and I had added Logo animation to our text, in 1989 accessing a videodisc player or scanned images seemed as far removed as having

Macintoshes in the classroom. In the past year or two, however, that has begun to change. Scanners are available for the older Apple computers, the II's and the IIGS, as well as for IBM computers. Videodisc players are being produced that can be interfaced with Macintoshes. The same connectors used with a videodisc player and a Macintosh may be used on an Apple IIGS with the same cable, while a special cable is needed for an Apple IIe. People are beginning to develop ways to use these remarkable machines with Logo (see, for example, the numerous articles by Glen Bull and Gina Bull (1990-1991) in last year's *Logo Exchange*). The changes in the technology available began to change what I wanted students to be able to do with *LogoWriter* hypertext too.

Hypermedia with LogoWriter

Last winter Michael Tempel and I decided to combine our two versions of hypertext into a single program incorporating the best ideas of each. To a new set of *LogoWriter* Hypertools we added Vidtools, procedures to access a videodisc player.

Experimenting with the videodisc player and *LogoWriter* produced some surprises. First, it turns out to be very easy to use a videodisc player with *LogoWriter*, once you have some tools to help you. Using a videodisc player also adds a new dimension to programming, not unlike the way in which thinking in terms of time on and off works with *LEGO TC logo*. Frames can be turned on and off; sequences of still frames can be programmed; motion sequences can be played in their entirety at varying speeds. The biggest surprise was that the computer sometimes processes faster than the videodisc player. The end of a procedure can be reached before the poor player is up and running! We had to begin to think of slowing things down to let the "toddler" catch up. The most effective way to do this was to have the procedure wait for some kind of input from the user, usually by adding a line of instructions for the user and then waiting for a key to be pressed:

```
type [\ \ \ \ Press a key to see the
      video.]
name readchar "key
cc
```

This is a nice way to pause a procedure. Type takes a single input (a quoted word or a list) and puts that text in the *LogoWriter* Command Center. In the example, it simply asks the user to press a key. The four backslashes simply force Logo to treat the spaces following them as spaces (technically, the backslashes "reserve" the space), thereby indenting the text neatly. Newer versions of *LogoWriter* allow you to surround with vertical bars all text (including spaces) that is to be treated as a unit.



The next line of code, name `readchar "key"`, causes the procedure to pause until it can read a key typed at the keyboard. Once the user presses a key, the Command Center is cleared and the procedure can continue. With the video display, this allows the videodisc player a chance to catch up with the computer.

You might also play with using the `label` command to put instructions on the computer screen. Repeating the same command at the end of the procedure will remove the instructions:

```
label [Press a key to see the  
      video.]
```

Move the turtle to one side of the screen before using a command like this so you can center the text neatly on the screen. `Label` puts text on the screen, but the text is actually graphics. To remove the text, you must `label` the same text on top of it. It's rather like turning a switch on and off. The `label` line needs to be added at the end of the procedure to turn the text off.

A similar process can be used with the `Vidtool` procedures to put text on the video screen. This makes it possible to direct the user's attention from screen to screen and still be able to identify images or give necessary instructions or information on either screen.

Who Might Use Hypertext?

In one of my *Logo Exchange* articles (Adamson, 1988), I wrote about the problem of deciding when to teach students something:

As with most new Logo ideas, I believe they are ready [to use it] *when they need it*. ... we as teachers need to look carefully at our decisions about when and what to teach. Are we holding back on teaching something a child can really use because of our own misconceptions about what that child can understand? (p.5)

I thought about this last spring when I tried out hypertext with a group of willing fourth-grade students. The intellectual idea of working with hypertext seems quite sophisticated. I had originally developed it for use with seventh-grade students. I wondered if fourth-grade students would understand how it worked. In true Logo style, I decided to find out what would happen.

Seeing Is Believing

I began by showing an example that used the Visual Almanac (1). It was, appropriately enough, a document about turtles. To my surprise my students were instantly ready to begin on their own hypertext projects! They saw its validity. For their science class

they had each been assigned a planet to learn about. Then they were to teach their class about that planet. Since there were many images of the planets on the Visual Almanac, they immediately perceived the usefulness of hypertext with video. In fact, they envisioned a multidimensional presentation: the computer, the videodisc, and themselves. Some students grasped more clearly than others the idea of words being references to other words or video images. They could hardly wait to get started!

Time passed, and what I thought might be a brief project grew into a considerable piece of work for each small group of students. The Visual Almanac comes with *HyperCard* stacks about each section of the disk. Each card contains information about the frame or animation. On the Macintosh on my desk I placed the files from the planets section of the Visual Almanac. The students had access to books and information from the school library, but they also browsed through the planet stacks on my Macintosh when they wanted more information, especially specific information about the frames they had chosen to display. In the end, not only were there some impressive "live" presentations for the class, but the students had certainly done more work researching and writing than either their science teacher or I expected.

Other students are now curious and somewhat interested in working with hypertext. One student looked at the frames on hurricanes and began a document on hurricanes in Florida. She was so excited that she went with her mom to the library that day after school in order to get more information about hurricanes. Another student, who likes dinosaurs, decided to use the dinosaur images on the Visual Almanac for his project. *Hypertext LogoWriter* adds an extra dimension to Logo work, which is especially interesting to students who like to write. *LogoWriter Hypertext* is such an easy authoring tool that very little instruction is needed to use it (2).

Just recently at a conference I heard a talk about another computer language, Boxer. One reason for its development was to create a language able to do some of the things you cannot do with Logo. What things? Such things as ease of editing, use of text, and hypertext! Now it seems all there, and yet it seems, in more ways than one, there is certainly no ceiling. I think we should begin dreaming about what Logo will be like 10 years from now. Are any of us as prescient as Vannevar Bush about the future of technology, particularly where Logo is concerned? What *will* be next?

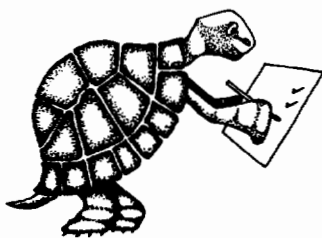
References

- Adamson, E. (1988). Logo Ideas. *Logo Exchange*, 7.
Bull, G., & Bull, G. (1990-1991). Logo connections. *Logo Exchange*, 9(1-8).

- Bush, V. (1986). As we may think. In S. Lambert & S. Ropiequet (Eds.), *CD Rom: The New Papyrus*. (pp. 3-20). Redmond, WA: Microsoft Press (Reprinted from *Atlantic Monthly*, July, 1945)
- Pavic, M. (1988). *The dictionary of the Khazars*. New York: Alfred A. Knopf.

Notes

- 1 The Visual Almanac is available from Optical Data Corporation, 30 Technology Drive, Warren, NJ 07059. It comes with the videodisc, HyperCard stacks on disk and on a CD-ROM, a manual, and a cable to connect the player to the computer.
2. *LogoWriter Hypertext*, by Michael Tempel and Eadie Adamson, can be obtained from Logo Computer Systems (1-800-321-5646).



The Turtle was chosen for a convergence of several reasons:

1. We needed something with which one could identify, so an animal seemed the best choice, although we considered all manner of vehicles
2. We thought animals that were slow and deliberate would be best—no hares, grasshoppers, birds, etc.
3. We wanted something pleasant and not scary

There aren't many animals that qualify up to this point, but there is another factor.

4. In early work on cybernetics, a British neurologist named Grey Walter had made an artificial

Eadie Adamson is a member of The New Laboratory for Teaching and Learning at The Dalton School. The New Lab is dedicated to developing innovative uses of technology in education. Eadie works primarily with students in grades four through eight, using *LogoWriter*, *LogoExpress*, and *LEGO TC logo*. She also works with math and language classes using *LogoWriter*.

Eadie Adamson
The Dalton School
108 East 89th Street
New York, NY 10128
(212) 722-5160

E-mail: LCSi LogoExpress BBS, New York and
Montreal: EadieA
CompuServe: 73330,3266

Why a Turtle?

Thoughts from Seymour

animal that caused a stir in the 1950s by finding its way around, plugging itself into an electric socket when its batteries were low, etc. He called his construction a Tortoise.

So the Turtle honored and continued Grey Walter's contribution. It was a step forward in that his was not programmable—it just did what it did by instinct (so to speak)—and of course his was not made with an educational intent.

One German translator of Logo maintained that children there hated turtles and decided to make it a hedgehog. I found it hard to believe. But I think that children's liking for turtles is confirmed by experience, including the Ninjas.

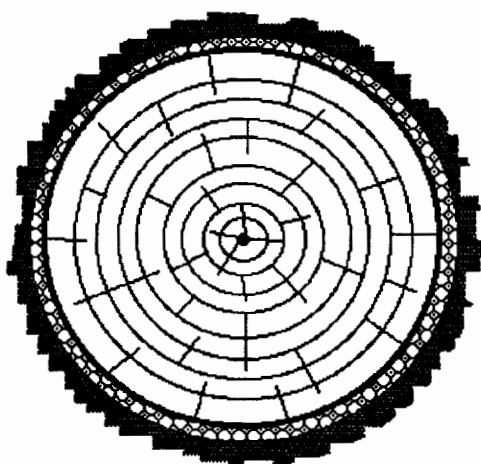
Thanks, Seymour!



Arboreal Aging

by Judi Harris

Age can be represented in many ways. Years accumulate to a running total, resale values depreciate, grey hairs multiply, wrinkles trace tributaries of skin folds, and, in the plant kingdom, arboreal perennials get taller and thicker. One way to represent *Logo Exchange's* age looks like this:

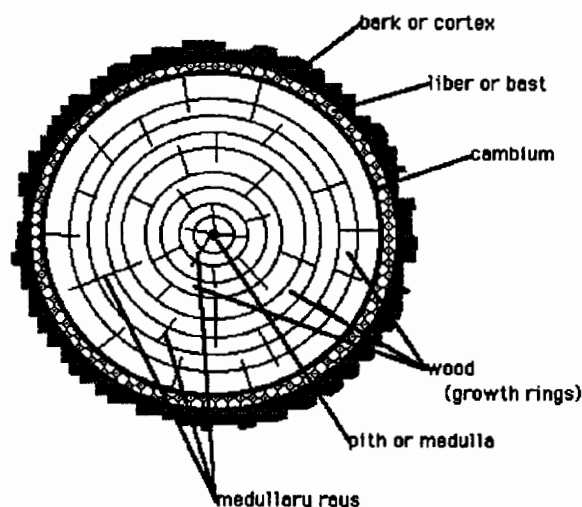


Since the 2000+ copies of *LX* that are mailed for each issue are primarily made from trees, it is fitting that this 10th anniversary edition of *Logo LinX* link Logo exploration to arboreal study. Certainly the *LX* exists because of the creativity, dedication, labor, and interest of its authors and readers, but without the paper on which it is printed, the exchange of ideas, experience, and support that has characterized *LX's* 10-year history would have been impossible.

A Timely Topic

Autumnal curricula often include the study of trees. Students make leaf collections, take hikes with their field guides to identify tree types, explore the chemical processes that cause leaves to change color, diagram and analyze venation and branching structures, and make rubbings of bark and leaf textures. This fall, why not help your students to encode and decode arboreal age? (Lanzara & Lanzara, 1978)

The trunk of a tree grows larger each year as it performs its job of sustaining leaves, flowers and fruit. A cross-section of a tree trunk reveals several distinct types of cells, each with its own purpose.



- The *bark* or *cortex* protects the tree from atmospheric changes and, to a certain extent, from damage caused by animals and people.
- The *liber* or *bast* allows the sap to descend and redistribute after it has been photosynthetically processed in the leaves. It also fuels the growth of the *cambium*, the next cell layer toward the center of the tree.
- The *cambium* is primarily responsible for the diametric growth of the tree. It produces woody cells toward the inside of the trunk, and liber cells toward the outside of the trunk.
- The *wood* is comprised of a series of concentric rings, one for each year that the tree has been alive. Woody fibers and vessels in the outer rings (*sapwood*) transport raw sap (a watery solution of mineral salts taken from the soil) to the tree's leaves, where photosynthesis transforms these inorganic elements to a solution of organic compounds. Older growth rings (*heartwood*) are dead, performing the functions of support and storage. This is the wood that is cut and used for building.

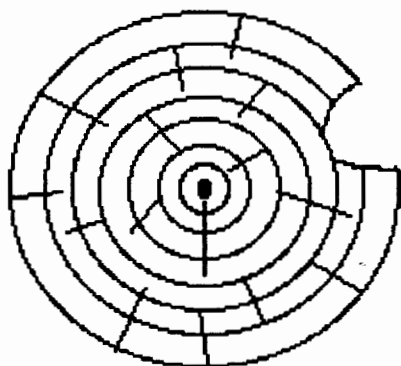
- The *pith* or *medulla* is the central part of a trunk in its first year, but as the tree ages, its size reduces, and it can even disappear, leaving an empty medullary cylinder.
- The *medullary rays* are fine lines of cells that stretch along the trunk's radius from the pith to the bark and distribute nutrients to all parts of the tree.

Ringed Maps of Time

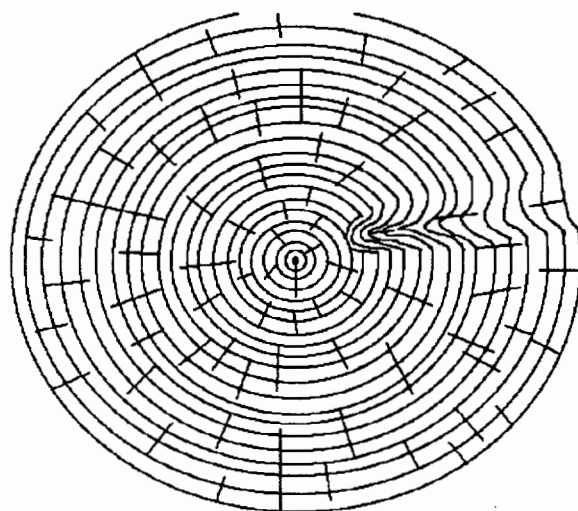
Scientists in ancient Greece and Rome recorded observations of tree trunk cross-sections, but it was Leonardo da Vinci who realized that a tree's age could be discovered by counting its rings, and that yearly climatic conditions could be deduced from differences in ring width. A.E. Douglass, an American astronomer living at the beginning of the 20th century, established the foundation of the science of *dendrochronology*, or the systematic study of the interrelationship of tree ring appearance, environmental variation and time. These scientists have shown us how to reconstruct a tree's life history from one cross-section of its trunk.

Each annual tree ring has two parts: a section light in color, comprised of large vessels, and formed during the spring, and a section darker in color, comprised of narrower, more numerous vessels, formed during the summer, when less moisture must be transported to the tree's already-formed foliage crown. Thus, counting a tree's annual rings reveals its age in years. But closer examination of the rings' width, form, and color can allow us to infer important events in the life of the tree.

The diameter of each tree ring depends directly upon the climatic conditions to which the tree was subjected during that year; wider rings indicate more favorable growing conditions than narrower rings. Ring color is determined, in part, by the chemical composition of the environment in which the tree exists. Marbling, or deviations from ring concentricity, is caused by a change in the growth pattern of the tree, such as the formation of a branch or the sustaining of an injury.



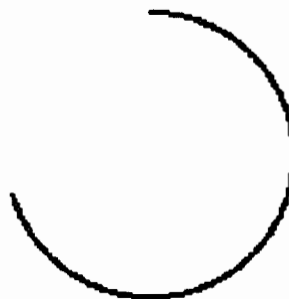
This injury, occurring in this tree's 7th year, is gradually accommodated until it is externally quite imperceptible in the tree's 20th year. Such intrusions upon ring patterns are what cause aesthetic elegance in the shape, color, and variety of cut wood grain.



The "Logo Link" should be obvious by now. Why not challenge your students to construct a tree trunk cross-section, ring by ring?

Arboreal Arithmetic

Successive tree rings differ primarily by diameter. Drawing these concentric circles can be quite a programming challenge in itself for novice Logo users. Discovering that a REPEAT 360 [FORWARD 1 RIGHT 1] circle, for example, can also be drawn with the turtle travelling along the circle's radius can represent an exciting change of cognitive context for a beginning Logo programmer.:



```
TO CIRCLE1
REPEAT 360 [PU FORWARD 57 MARK BACK
57 RIGHT 1]
END
```



```

TO MARK
PD
FORWARD 1
BACK 1
PU
END

```

Concentric circles can then be drawn by changing the inputs to FORWARD and BACK in CIRCLE1. This also can provide a perfect introduction to the use of local variables, since the length of the radius that the turtle travels is all that varies between tree rings:

```

TO CIRCLE1 :RADIUS
REPEAT 360 [PU FORWARD :RADIUS MARK
BACK :RADIUS RIGHT 1]
END

```

More experienced Logo programmers may enjoy writing procedures that draw concentric circles with the turtle travelling around the circle's circumference rather than along its radius. In this interpretation of the challenge, radius size is used to make the computer calculate the size of each step the turtle takes. Since the circumference of a circle is equivalent to $2 \times \pi \times$ the radius, each step the turtle takes is equivalent to that product divided by 360:

```

TO STEP :RADIUS
OUTPUT 2 * 3.1416 * :RADIUS / 360
END

TO CIRCLE2 :RADIUS
REPEAT 360 [FORWARD STEP :RADIUS
RIGHT 1]
END

```

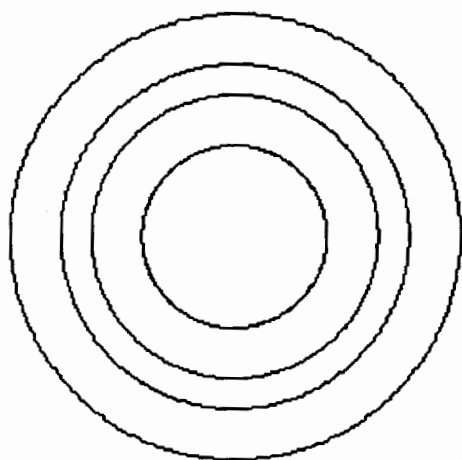
Concentric circles drawn with the above procedures can be positioned with MOVE.OUT and MOVE.BACK procedures, which are invoked before and after each call of CIRCLE2, respectively:

```

TO MOVE.OUT :RADIUS
SETH 0
PU
FORWARD :RADIUS
PD
SETH 90
END

TO MOVE.BACK :RADIUS
SETH 0
PU
BACK :RADIUS
PD
END

```



```

MOVE.OUT 45 CIRCLE2 45 MOVE.BACK 45
MOVE.OUT 70 CIRCLE2 70 MOVE.BACK 70
MOVE.OUT 85 CIRCLE2 85 MOVE.BACK 85
MOVE.OUT 110 CIRCLE2 110 MOVE.BACK 110

```

Arborescent Autobiographies

If the appearance of tree trunk cross-sections symbolically tell the tree's life story, why not use these woody models to help your students construct their own arboreal autobiographies? The color, thickness, and ring pattern of each year in their lives could be drawn with LogoWriter, for example, with text on the screen explaining the events that caused ring modulation, color change, and diameter fluctuations. These diagrams could then be shared and compared with those of classmates to see if any "across-trunk" patterns during specific years in students' lives could be detected. Similar tree-ring histories for schools, historical figures, literary characters, or countries could also be created.

Some Final Thoughts

Marshall McLuhan once wrote:

For tribal man space was the uncontrollable mystery. For technological man it is time that occupies the same role.

The Mechanical Bride, 1951

Where will LX be 10 more years into the future? Will we still be reading it on printed pages or will it be distributed electronically to its subscribers? Will students still be using turtle graphics to explore *Logo LinX* like the mathematics of tree growth and personal histories expressed in arboreal symbols? Or will our classrooms be sufficiently "Logo-like" so that student-centered, interactive, interdisciplinary explorations, using a wide va-

riety of learner-based technological tools, will be the rule rather than the exception? Only time, and the collective vision and action of readers like you, will tell.

Reference

Lanzara, P. & Lanzara, M. (1978). *Simon & Schuster's guide to trees*. New York: Simon & Schuster Inc.

Judi Harris works in the Department of Teacher Education at the University of Nebraska at Omaha as an assistant professor of educational technology. Her teaching, research, and service interests include Logo (of course), developmental sequencing in interactive hypermedia materials design, computer-

mediated educational telecommunications, the expressive qualities of children's computer-assisted artwork, and the restructuring of teacher education paradigms.

Judi Harris
522H Kayser Hall
Department of Teacher Education
University of Nebraska at Omaha
Omaha, NE 68182

BITNET: JHarris@unoma1
Internet: JHarris@Zeus.unomaha.edu








LogoWriter Activities for Readers



**Build reading and writing
skills in grades 1 through 6**

- **Twenty-five scrapbook disks** with easy-to-follow student directions and teacher notes
- Supports **whole language, writing process,** and **traditional** approaches to instruction
- Works in conjunction with LogoWriter 2.0 or higher
- Apple IIe, MSDOS, and Macintosh versions
- Single classroom version - **\$99.50**
- Building and district licenses based on population

Call for sample disks: 201 • 703-9676

My name is Fred and I live in a big  at the end of my street.
We have lots of   in the yard and a very large oak  that
I love to climb. My father takes a  to work each day and my
mother has the  so she can drive me to school. I have a 
named Sparky.

Fluffy is a happy cat who loves to play in the flower garden. His favorite game is hiding from Wolf, the big dog next door. Sometimes when I walk by, he jumps out and grabs my foot. That scares me for a second. I love fluffy.



Copyright 1990 Compuco, Inc.
LogoWriter is a registered trademark of Logo Computer Systems, Inc.

Compuco, Inc. • 26-05 D Fair Lawn Avenue • Fair Lawn, N.J. 07410 • 201 703-9676



Turtles Invade the Kindergarten Classroom!

by Molly Kniffen

Computers, computers, everywhere—even in a kindergarten classroom! What will they think of next? What do I do with a room full of five-year-olds just itching to get their little hands on that expensive machine? Five-year-olds can't program a computer! Why, they can't even read!

These concerns are common to many teachers. The problem isn't whether or not five-year-olds are capable of programming a computer. The real problem is that most kindergarten teachers have a fear of computers and of programming.

Most anyone can program a computer to some extent. You don't necessarily have to be able to read to program. Take, for example, what happened the day the turtles invaded the classroom.

The Fantasy Begins

Everything was running along smoothly in our classroom when suddenly we received word that our school was being invaded by enemy turtles. The children became very excited and agreed to help keep a watch out for these unwelcome visitors.

After several group discussions on what an enemy was, the children reached the conclusion that they did not want this type of person in our classroom. We decided that enemy turtles would surely want to fight in order to take control of our classroom. This was terrible because, after all, fighting was definitely against the school rules! As the day progressed, the children kept a very close watch for the invaders, but to our relief, no turtles appeared.

When the children arrived the next morning, we were told that one turtle had been seen in our school building earlier that morning. We were also told that these turtles were not the type of animals we were used to seeing and catching at the pond or lake. These turtles had changed their shape to have three straight sides that connected and three angles or corners where their sides joined. Since the children were studying shapes, it didn't take long for them to design a turtle in the shape of a triangle.

Since all the corners looked the same, we decided it was best to just color one corner of the triangle so we could locate its head and we could tell which way it was headed.

We still felt fairly safe, however, because we usually kept our door closed and we felt that no turtle could get into our classroom through a closed door. But, just

to be certain, we decided to build a cage. After all, we might have to help some other teacher capture an enemy invader even if the turtle didn't wander into our classroom.

As the children returned from P.E. that day, they found our room in complete chaos. The turtle had invaded! We had to have a plan immediately. Everyone agreed that since the turtle was an enemy, if we should find one in our class, only the teacher should try to pick the turtle up. After all, what if the turtle was really mean? The turtle might try to bite us!

After a short search, the turtle was spotted and was soon in captivity. We were safe again. What a relief!

Now we were faced with a new problem. What do you do with an enemy turtle? Even though the turtle looked friendly enough, how could we be sure? We finally decided to keep the turtle as a pet and try to teach him proper school behavior.

Since the children were studying shapes in mathematics, we decided to start by teaching the turtle about shapes. The children already had a basic understanding of the properties of the square, the rectangle, the triangle, and the circle.

We took the turtle out of the cage and placed it on the floor. It would not move. It just lay there. We tried everything to make the turtle move. We stomped our feet, we clapped our hands, we called the turtle, but nothing worked. The turtle simply would not move. We finally decided to choose a student to hold the turtle and make the turtle walk. This had to be a very brave person. Being very careful to point the turtle's head away from our bodies (just in case the turtle got mad and decided to bite), we would walk as directed to help the turtle form a square on the floor. For example, we would instruct the turtle to go forward three steps, turn left, go forward three steps, turn left, go forward three steps, turn left, and go forward three steps. The children again reviewed the fact that squares had four corners and four sides. We were reminded that all of the sides were exactly the same length.

After several minutes of practice helping the turtle make different sized squares on the floor, we noticed that the turtle looked a little tired and maybe even a little sad. We decided this was a good time to return the turtle to the cage for a rest. We then discussed the fact that we could train the turtle if we remembered that it could only walk in the direction its head was pointed,



and that we had to tell it every direction to move and how many steps to take. (These skills would come in handy when the children found the "turtle" living in their own computer a little later).

The next morning when the children came into the classroom, we had a new machine in our class. A computer! The children could hardly wait to get their hands on the computer. After a discussion on the care and "feeding" of the new equipment, we loaded the computer with the EZ Logo software package by MECC. (This package is designed for use in the lower elementary classrooms.)

Suddenly a strange thing happened. There in the middle of our computer screen was a creature that looked a lot like the turtle we had captured in our classroom. Upon checking the turtle cage, we discovered that our turtle had escaped! The turtle had gotten into our computer!

What now? How were we to get the turtle out of the new computer? The turtle certainly looked happy enough. We even thought the turtle was smiling. We decided to let the turtle live in our computer. The turtle seemed to really like his new home.

The children then discussed how they could make the turtle move inside the computer. After all, the turtle would need some exercise. After a short review of the skills we had learned from the previous day, the children discovered that by pressing certain keys, they could make the turtle walk. To make the turtle walk forward, they pressed F. To make the turtle walk backward, they pressed B. To make the turtle turn left, they pressed L. To make the turtle turn right, they pressed R.

The children were then challenged to see if they could make the turtle in the computer draw a square on the computer screen. They were reminded that all of the sides had to be the same length and that the square had to have four corners or angles.

And Look What Happened

It was amazing! With a basic understanding of shapes and a limited understanding of computers, the children were able to draw shapes. With a little extra help and guidance, the children could make rectangles, triangles, and even circles. With a little additional instruction and a lot of experimentation, the children were able to combine shapes to make original pictures.

The five-year-olds had suddenly become computer programmers! The activities built one upon another. First the children discovered what a turtle was and what it looked like. Next, they discovered that it moved in the direction its head was pointed (the orientation of the turtle on the computer screen). They then discovered how to make the turtle follow their commands to move and turn. The children still didn't know how to read, but they had experienced success on the com-

puter. All they had needed was excitement and a willingness to try.

Regardless of their individual learning abilities, all the children had experienced some degree of success from the invasion of the turtles. The gifted students were challenged, the at-risk students felt success, and everyone had fun.

Teachers Take Note

If a five-year-old can program a computer, so can we. Regardless of our educational background and past experiences, we can learn to work with computers too. All we really need is a willingness to try.

So come on, kindergarten teachers, get with the program! Give it a try. Who knows? Maybe we are really all computer programmers in disguise.

This article was produced under a grant from the Texas Higher Education Coordinating Board and the U. S. Education Department under the auspices of the Eisenhower Mathematics and Science Grants Program, Title II.

Molly Kniffen
312 Bascom Road
Whitehouse, TX 75791



...the heart of mathematical research is tinkering...

by A. J. (Sandy) Dawson

It was during the fall of 1983, I think, that a graduate student in Simon Fraser University's brand new Masters program for computers in education came to me and said, "You know, we really should do something with Logo. It is the only thing I have seen that holds any real promise for computers in education at this point." I had to admit I didn't know what Logo was. When I relayed this student's suggestion and my own ignorance about Logo to the remainder of the class, another student piped up with a further suggestion. "Why not," he said, "invite Dan and Molly Watt to Vancouver to give some courses on Logo?" And so we did. During the summer of 1984, Dan and Molly came to Vancouver for five weeks. They taught a couple of Logo courses and anchored a wonderful retreat/conference held at Whistler, Canada's renowned world-class ski area just two hours north of Vancouver. There wasn't any snow on the mountains at that time, but the participants did take the chairlift up the mountain and there formed the BC Logo Users' Group.

The following year, 1985, Gerri Sinclair, David Bell, and I organized the World Logo Conference, the first simultaneous on-line and on-site telecommunications-based conference, from a remote centre in British Columbia called Paradise Valley. The call went out to join us in being On-Line from Paradise, and for two days and two nights in mid-October we were connected around the globe with people in Great Britain, Israel, Australia, all across North America, and into parts of South American and continental Europe. Dan and Molly came back to join us in Paradise, and they were joined by a number of researchers from Canadian universities. On-Line guests and participants included Uri Leron, Seymour Papert, Richard Noss, Celia Hoyles, Tom Lough, Bill Higginson, and many more.

It was not till 1986 that I met Tom Lough in person and he invited me to edit a column for Logo Exchange. My column MathWorlds has appeared regularly since that time. Though it is LX's 10th anniversary, MathWorlds has only been around for half that time.

Ah, yes, those were exciting times, but now LX embarks on a new path. Are the times still exciting? Are Logo and the Logo-like environments that have developed in the past five years still at the forefront of the computers-in-education world as they were when my graduate students suggested that SFU should get involved? Have the teaching and learning of mathematics been impacted by the advent of computers, and in

particular by Logo and Logo-like settings? For this inaugural MathWorlds column for LX in its new format, I would like to address these questions, put forth some ideas, and relate some experiences that might provide some insights about these questions.

On the one hand, I think it is fair to say that if computers have had an impact on the teaching and learning of mathematics at the secondary school, it is not due to Logo. On the other hand, the recent appearance of three books, Al Cuoco's (1990) *Investigations in Algebra*, Phil Lewis' (1990) *Approaching Precalculus Mathematics Discretely*, and Paul Goldenberg and W. Feurzeig's (1987) *Exploring Language With Logo*, bodes very well for the potential of Logo and Logo-like environments influencing the teaching of secondary school mathematics.

If anything in the realm of computer software has made its way into secondary school mathematics classes, it is spreadsheet software. Now interestingly enough, spreadsheets have the potential of being very Logo-like in their use. Indeed, Chris Bigum (1987) of Deakin University in Geelong, Australia, has written a delightful monograph entitled *Convivial Spreadsheets*, in which he argues persuasively for a reconstructed view of what mathematics is and how it should be taught. He further suggests that the one kind of software currently available that might prove attractive to traditional secondary mathematics teachers is the spreadsheet.

However, before any software, Logo or otherwise, will have an impact on the teaching of mathematics, it will be, in my view, necessary for curriculum designers, textbook authors, department chairs, and mathematics teachers to restructure their view of what mathematics is all about, what it means to know and do mathematics, and what secondary school students are capable of when they address mathematics.

In the introduction to his book, Al Cuoco (1990) makes the bold claim that "the heart of mathematical research is tinkering" (p. 7). The first thing to be noted from this statement is that secondary students should actually be doing mathematical *research*. Second, students should be *tinkering* when they do mathematics. Two questions arise from these almost heretical conclusions. What is it that students should research? What does it mean to let students *tinker* with mathematics, what tools should they use, and how long should this *tinkering* be allowed to continue?

The answers one gives to these questions depend

very much on one's view mathematics, learners, and the the interaction between learners and mathematics. Let's begin with the learners.

It is not trivial to say that all students must ultimately learn for and by themselves, that the construction of knowledge is a private affair. Students must make sense of the material presented to them, whether it is by a teacher, a textbook, a classmate, or a computer. Students must put their own meaning on the symbols, sentences, diagrams, pictures, graphs and so on. There is no choice in this matter. Someone else cannot learn for you. Someone else cannot put knowledge into your head, your heart, or your body. You must do all of those things yourself. All learners, young or old, have to create and re-create their own knowledge of the world. This does not imply a licence for teachers not to teach, by the way. Rather, it means that teachers have to interact with students in ways very different from the ways they may have interacted in the past, and they have to change their expectations both of themselves and of their students. More on this later.

There is, however, a further aspect to this individual knowledge-creation process. It is not the case that anyone's knowledge is as good as anyone else's knowledge. Individual knowledge has to be tested against the world outside of oneself to ascertain if the idiosyncratic view one has of the world is viable; that is, whether one's knowledge can stand up to the scrutiny of one's peers and feedback from the physical world. In the classroom situation, this means that students' ideas should be honoured and tested. The classroom becomes comparable to the mathematicians' academic community, where ideas are put forth at conferences and in papers, and the community of scholars reacts to these ideas. In that sense, the classroom is also a community of neophyte scholars, and the ideas, conjectures, and hypotheses produced by these scholars should be subjected to the test of viability. This is the Lakatosean view of the growth of mathematical knowledge (Lakatos, 1976). It is a view that Lambert (1990) has amplified and demonstrated in her work with fifth-grade children. Hence, though I have been referring to secondary school students, the case I am making applies equally well to elementary school children. Indeed, it applies to learners of all ages.

The classroom becomes, therefore, the world of mathematics for the student, the environment in which mathematical conjectures can be tested, refuted, and perhaps then modified or rejected outright. In short, the classroom should be the place where students tinker with mathematical ideas. In so doing, they will develop their *reflective* capabilities, a goal most educators would certainly embrace.

And what are teachers doing while their students are tinkering? Clearly, teachers are important and cru-

cial people in the mathematical community created in the classroom. For one thing, teachers by virtue of their training and expertise in the field of mathematics and mathematics education are, or should be, representatives of the larger mathematical community. This is not to say, however, that teachers simply pass *along* their expertise. As noted above, students have to make their own meaning and sense of things, and teachers cannot, figuratively speaking, pour their knowledge into the heads of their students. Sadly, too often in the past this is exactly what was attempted in mathematics classrooms. Teachers have to subordinate this kind of teaching to the learning of their students because it is the latter that is most important. Nonetheless, teachers should function as questioners and challengers of ideas, as sources of information when that seems appropriate, as suggesters of possible approaches to tackling of problems, and as supporters and encouragers of students when they are tinkering with mathematical ideas.

But with which mathematical ideas and concepts, you might well be asking, should students be invited to tinker? Here, knowledgeable mathematics teachers play vital roles. As representatives of that larger mathematics community, it is teachers who select from the current body of mathematical knowledge that which in their professional judgement would best match the needs, interests, and aspirations of their students. The body of mathematical knowledge from which they draw is—parallel to the growth of any learner's knowledge—built by individual mathematicians subjecting their conjectures to the scrutiny of the mathematical community. In this way, mathematics is a growing, vital body of knowledge, fallible and open to question. (See Paul Ernest's [1991] recent book, *The Philosophy of Mathematics Education*, for a description of the growth of mathematical knowledge as seen through a synthesis of Lakatosean, constructivist, and conventionalist perspectives.) Since mathematics is growing and changing, teachers must keep abreast of new developments in mathematics in order to maintain the freshness of their own knowledge. The advent of Chaos Theory and the creation of Fractal Geometry is a case in point where mathematics teachers need to update their knowledge so as to be able to draw on these areas in their teaching of secondary school mathematics. Several writers in a recent publication of the National Council of Teachers of Mathematics have begun to address the question of the requisite mathematical knowledge for teaching secondary mathematics, and, indeed, what the curriculum for secondary school mathematics might look like (Davis, Maher, & Noddings, 1990).

Thus far I have mentioned nothing about computer software or even Logo, but perhaps the connections are obvious. Papert proposed Logo as a computer environment where students could *tinker*. It is clear, moreover,

that Logo manifests powerful mathematical ideas. But not only Logo is capable of doing this. As mentioned previously, Chris Bigum (1987) argues convincingly that spreadsheets also provide the kind of *what if* environments so conducive to playing with mathematical ideas. Indeed, at their best, computers and the appropriate software are ideal tools for the exploration and creation of mathematical ideas, for the making of conjectures, and for attempts at refutation.

How long should students be allowed to tinker in these ways? Forever! That is not meant to be a glib answer. I am quite serious, for I think that students' experiences of mathematics should be seen as open-ended, and that any results that are produced, tested, and found viable by the mathematics community, locally in a classroom or globally in academia, have to be seen as *developments along the way* and not as end products. Students' experiences of mathematics in schools should have the flavour of a grand experiment with ideas of the mind, and the fruits of that experiment should provide the seeds for further explorations.

And yes, Logo and Logo-like software environments have a very central role to play in assisting teachers to fashion a learning community in their classrooms, where, as Al Cuoco (1990) said, "the heart of mathematical research is tinkering."

References

- Bigum, C. (1987). *The convivial spreadsheet*. Geelong, Australia: Deakin University.
- Cuoco, A. (1990). *Investigations in algebra*. Cambridge, MA: MIT Press.
- Davis, R. B., Maher, C. A., & Noddings, N. (Eds.). (1990). *Constructivist views on the teaching and learn-*

ing of mathematics. Journal for Research in Mathematics Education Monograph #4. Reston, VA: National Council of Teachers of Mathematics.

Ernest, P. (1991). *The philosophy of mathematics education*. London: Falmer Press.

Goldenberg, E. P. & Feurzeig, W. (1987). *Exploring language with Logo*. Cambridge, MA: MIT Press.

Lakatos, I. (1976) *Proofs and refutations: The logic of mathematical discovery*. Worrall, J. & Zahar, E. (Eds.). Cambridge Press: Boston.

Lambert, M. (1990). When the problem is not the question and the solution is not the answer: Mathematical knowing and teaching. *American Educational Research Journal*, 27(1), 29-63.

Lewis, P. G. (1990). *Approaching precalculus mathematics discretely*. Cambridge, MA: MIT Press.

Sandy Dawson is on leave as Director of the Professional Development Program at Simon Fraser University in Vancouver, Canada. At the time of the preparation of this column, he was a guest of the Mathematics Department of Concordia University in Montreal, Canada. Sandy wishes to thank the Department, and in particular Dr. Joel Hillel, for the hospitality shown him during his stay in Montreal.

A. J. (Sandy) Dawson
Faculty of Education
Simon Fraser University
Vancouver, B. C.
Canada V5A 1S6

Email addresses: Userdaws@sfu.bitnet

The First Decade of *LX*

by Glen L. Bull and Gina L. Bull

Academicians believe that books have the power to produce change, although it is rare that one does. Each year tens of thousands of books are printed, and by the end of the year the majority sink from sight without a trace. Yet one book changed our lives in remarkable, unforeseen ways.

Microcomputers

We first began to experiment with microcomputers in 1976. A perceptual experiment on which we were working involved presentation of sequences of tones. A colleague suggested that a new device known as a "microprocessor" might allow us to control the sequences more efficiently than the hardwired electronic equipment we were using at that time. Since it was not possible to purchase a preconstructed microcomputer, an engineering friend helped us to construct a wire-wrapped system. It had an Intel 8080 microprocessor, four kilobytes of Random Access Memory (RAM), and four kilobytes of Read Only Memory (ROM). Programs were written in assembly language. Since floppy disks did not exist, data were stored in the form of holes punched into a paper tape. That system is sitting downstairs in the basement as we write.

Over time a few students in the School of Education asked to work with us, and we began offering small seminars on uses of microcomputers. The initial courses did not address educational uses of microcomputers because it was difficult to create significant educational applications in 4 kilobytes of memory. Reviewing our lectures notes from that time, it appears that we taught students about things such as Boolean algebra, AND gates and OR gates, TTL logic, and how to create a software monitor. (A monitor in that context refers to a primitive predecessor to an operating system.) We taught those things not because they were particularly useful to education students but because they were the things we understood about microcomputers.

Improvements gradually arrived over time. We doubled our memory, and then doubled it again, to a total of 16 kilobytes. We acquired an 8 kilobyte BASIC on paper tape. It took about 20 minutes to load, but we were able to program in a higher level language. Later we acquired a Tarbell cassette interface, so we were able to save our programs on audio cassettes. A minor catastrophe occurred one Christmas when we inadvertently taped carols over a week's work of programs. Since then we have lost data in a number of ways, but

that was one of the most unusual. We soon acquired an S-100 computer that we built (from a kit this time) with 32 kilobytes of memory, and we acquired our first word processor.

Mindstorms

In 1980 a friend who knew of our interest in computers lent us a book that he said might interest us. We read *Mindstorms* (subtitled "Children, Computers, and Powerful Ideas") by Seymour Papert in one sitting that afternoon. We then sat up the rest of the evening discussing it. We passed it on to Steve Tipps, who was equally interested. Shortly thereafter we found that Dan and Molly Watt were going to offer a workshop on Logo in Washington, D.C. We quickly enrolled.

We arrived about 45 minutes before the start of the workshop. The computers, which were being shipped from Texas to Washington, D.C., had not arrived. Dan was considering how the workshop might be taught without computers if they failed to appear. The shippers began to bring the computers in during the midst of this discussion. They had arrived 30 minutes before the workshop.

The scene as boxes were opened and computers were unpacked was one of controlled chaos. It was a sight that would become familiar over the years. The first Logo workshop which we took was taught using TI 99/4 computers. At that time this was the only commercially available version of Logo. We also were able to see a demonstration version of Logo on an Apple II computer, although it had not been released for general use at the time of the workshop.

The workshop convinced us of the educational potential of Logo. On the drive back home that evening, we stopped for dinner to discuss how we might proceed. The most logical step seemed to be to hold an experimental seminar that would allow us to learn more about the educational philosophy of Logo with some of our students. The only difficulty was that we did not have any computers with Logo. The following day we called Texas Instruments, and the corporation agreed to donate a half-dozen TI 99/4 computers with Logo.

As the beginning of the semester drew closer, the computers still had not arrived. Finally, on the morning of the day of the first class, boxes from Texas Instruments began to appear. We were unpacking computers with Steve Tipps to prepare for the class when an



inquisitive physics major appeared. (The discerning reader will begin to detect a recursive pattern.) Tom Lough said that although he was not an education major, he had read about Logo and would like to enroll in the seminar. As it developed, this was not to be the last education course which Tom took. *Mindstorms* had just reached out and changed the course of another person's life.

That semester was one of the most exciting of our academic careers, as we discovered the potential of Logo. We jointly taught the course with Steve Tipps. As the semester progressed, it became clear that there was a potentially useful role for technology in education. (It was not at all clear to us that this was the case prior to that time, despite some extravagant claims which had been made.) We did not even have a Logo manual for TI Logo, which had not yet been printed. Instead Texas Instruments sent photocopies of the notes for a book on Logo which Hal Abelson was writing (later published as *Logo for the Apple II*). In the midst of this exploration the most exciting aspect of the semester was not the technology itself but the educational philosophy Logo embodied.

It was clear that the technology would change over time. The TI 99/4 did not have a floppy disk drive at first. The Logo program was on a ROM cartridge, and students programs were saved to cassette tapes. The cassette tapes were slow and sometimes unreliable. When the first floppy disk drives for the TI 99/4 were made available, they were expensive (about \$1,000 per drive) and only held about 100 kilobytes of data. A decade later we are writing this on a computer with a hard disk drive that cost less than \$500 and holds more than 100,000 kilobytes (or 100 megabytes) of data. In other words, the storage device is less than half the price, but has more than a thousand times the capacity. As the technology changed, the capacity of the programming language increased, taking advantage of the increased computing power. However, the educational philosophy remained constant.

Logo Exchange

During the middle of the semester, Tom Lough told Steve Tipps and us that he would like to create a journal that would allow teachers to exchange ideas about Logo and Logo-like approaches to teaching. Although an academic institution is a place of intellectual ferment, and students often develop exciting ideas, this discussion must have made a particular impression because we recorded the notes from that conversation, and we still have them. In short order Tom completed his master's degree in physics, established the Logo Exchange, and entered the doctoral program in education. (We observed with some amazement that Tom is a very unusual and energetic individual.)

Tom initially used his basement as a production facility. As the magazine matured, it was clear that it was outgrowing Tom's basement. There was a short period of time when it was distributed by a commercial publisher, but it was soon clear that the journal needed academic oversight. *Logo Exchange* then found a home with the International Society for Technology in Education (ISTE), where Sharon Yoder has served as editor and ably guided its growth as Logo has matured.

During the past decade we have written a column for each issue of *Logo Exchange*, for a total of approximately 90 columns. (During one year we also wrote a series of articles with Paula Cochran about uses of Logo in special education, in addition to the regular column.) Gina's original degree was in art history, and when *Logo Exchange* was founded she was working as a librarian in the Fiske Kimball Fine Arts Library. As a result of her interest in Logo, she returned to school to acquire a graduate degree in computer science (taking all the courses in mathematics she had avoided in her undergraduate years), and joined the staff of the department of computer science. Consequently, her interest in Logo now encompasses both the dimensions of art and computer science.

Logo did not begin with the age of microcomputers. More than a decade of research on larger computers preceded its implementation on personal computers, and some of the most important writing on the uses of Logo is found in working papers from the M.I.T. Artificial Intelligence Laboratory from that era. However, the availability of Logo on personal computers has made it possible to observe uses of the language outside a controlled environment. This latter period roughly coincides with the decade of publication of *Logo Exchange*, which has recorded many of the uses teachers have discovered and invented during this time.

Logo Tools

Against this background of a decade of use in the public schools, it may be worth considering what the most salient educational characteristic of Logo may be. The first impulse of the casual observer might be to suggest that the most important characteristic of the language is a programming feature, such as the Logo turtle or recursion. However, our conclusion after a decade of watching teachers and students use Logo is that its most important characteristic may be summarized with the word "tools."

Humans are tool-using animals. At one time it was said that humans were the only tool-using animals. Three decades of research by Jane Goodall have demonstrated that this is not the case. Chimpanzees shape twigs to extract termites, and other uses of tools by animals have also been observed. However, even if humans are not the only tool-using animals, the use of

tools is certainly central to the human experience.

In a different era our environment fostered use and experimentation with tools. The agrarian economy of the last century rewarded tinkering and ingenuity; the phrase "held together with baling wire" had its roots in practical necessity. In the first half of the 20th century, kitchen table experimentation with crystal radios and electronic circuitry evolved. Even the term "bread-board" (electrical circuits were frequently constructed on a breadboard) betrays the home origins of this exploration. At first, tinkering was essential if a car owner wished to keep a vehicle running, and even after cars became more reliable, a generation of shade-tree mechanics continued to work on automobiles in the back yard.

In the 1960s and 1970s the widespread use of the integrated circuit led to great advances in electronics. This trend has played an important role in the decreasing cost and increasing power of personal computers. However, at the same time it has made technology more opaque. The child who opens a radio today sees printed circuits that provide no clue to betray their functions. They do not encourage experimentation or tinkering.

Logo provides a modern environment for experimentation and tinkering. It is literally possible to take apart a procedure to see how it works. In a well-constructed, modular Logo program each of the subcomponents can be run independently. The term "software construction kit" has been used to describe programs such as *HyperCard* for the Macintosh and *Toolbook* for IBM computers. Logo was the first such software construction kit designed for practical use in a public school environment.

A software tool does not have to be complex to be useful. One of our favorite Logo tools consists of two lines:

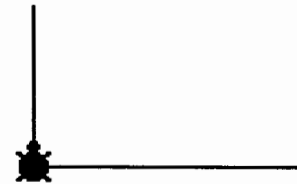
```
To Line :Length
FORWARD :Length
BACK :Length
End
```

Consider the following procedure to create the axis for a bar graph. Without the use of the line tool, it looks like this:

```
To Axis
FORWARD 100
BACK 100
RIGHT 90
FORWARD 100
BACK 100
LEFT 90
End
```

With the addition of **Line** to the tool repertoire, the **Axis** procedure can be written in the following way:

```
To Axis
Line 100
RIGHT 90
Line 100
LEFT 90
End
```



Creating the Axis for a Bar Graph

Attributes of Good Tools

A good Logo tool has several attributes:

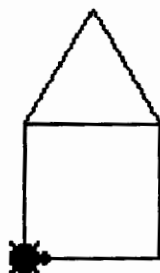
- The name of the tool should suggest its function.
- A good tool should be general purpose in nature so that it can be employed in a variety of situations.
- A Logo tool should usually leave the Logo environment in the same condition as it found it.
- It is desirable for the code that comprises a tool to be comprehensible so that anyone with a general background in Logo can understand the procedure.

We will explore these characteristics of a good tool by examining another favorite tool we often use, the **Over** procedure. It is often the case that you may want to move the turtle over by a certain amount. For example, if you have just made a house, you may wish to move the turtle over to make a second house.



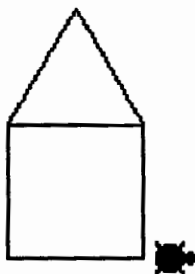
Moving the Turtle Over

This process consists of three steps. First you must flip the turtle on its side with the command RIGHT 90. If you do not want the turtle to draw a line as it moves over, you may also want to pick up the pen before moving.



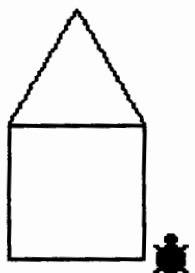
Step 1: Turning the Turtle on Its Side

The second step involves moving over the desired distance. In this instance, the distance traveled will be the width of the house (50) plus a few steps more.



Step 2: Moving Forward

The third step consists of flipping the turtle upright again by using the command LEFT 90. In order to restore the turtle to its original state before drawing a second house, it will also be necessary to put the pen back down.



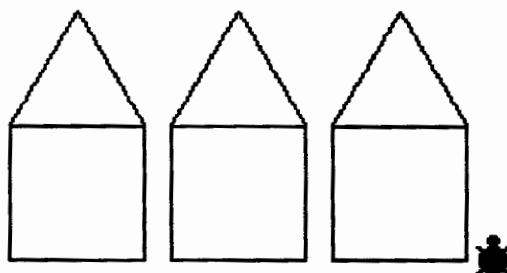
Step 3: Flipping the Turtle Upright

When this sequence of steps is put together into a procedure, it may look like this:

```
To Over :Distance
PU
RIGHT 90
FD :Distance
PD
LEFT 90
End
```

This greatly simplifies the process of drawing a row of houses.

```
To RowOfHouses
Repeat 3 [House Over 60]
End
```



A Row of Houses

State Transparency

The *Over* procedure as written above fulfills three of the four conditions that we have outlined for a good tool: (1) it has a meaningful name, (2) it is general purpose, and (3) the code is understandable. It violates one of the rules; it does not always leave the turtle in the same state as it was originally. Even if the pen was up before the turtle was moved, *Over* always puts the pen down at the end of the procedure.

In some versions of Logo it is possible to check the state of the turtle's pen and put it back in the same condition as it was found before the move. For example, in Terrapin Logo the *TURTLESTATE* command can be used to check the status of the turtle's pen:

```
To Over :Distance
MAKE "PEN.STATE FIRST TURTLESTATE
PU
RIGHT 90
FD :Distance
LEFT 90
IF :PEN.STATE = "TRUE PD
End
```

Unfortunately, the command to check the status of the turtle's pen is different in various versions of Logo. For example, in LCSI Logo II for the Apple II, the command to check the status of the turtle's pen is *PEN* rather than *TURTLESTATE*:

```

To Over :Distance
MAKE "PEN.STATE PEN
PU
RIGHT 90
FD :Distance
LEFT 90
IF :PEN.STATE = "PENDOWN [PD]
End

```

In those particular dialects, checking the state of the pen at the beginning of the procedure makes the the procedure more powerful, but less understandable. It would be desirable if there were a consistent command to check the status of the pen in all dialects of Logo. For example, the command `PENDOWN?` might be used to check whether the pen was down. (As far as we know, this command does not exist in any current version of Logo.) This would make it possible to write the procedure in this way:

```

To Over :Distance
IF PENDOWN? [MAKE "PEN.STATE "True
  PU]
RIGHT 90
FD :Distance
LEFT 90
IF :PEN.STATE = "TRUE [PD]
End

```

Another possible way to write the `Over` procedure would be in the following manner:

```

To Over :Distance
SETX XCOR + :Distance
End

```

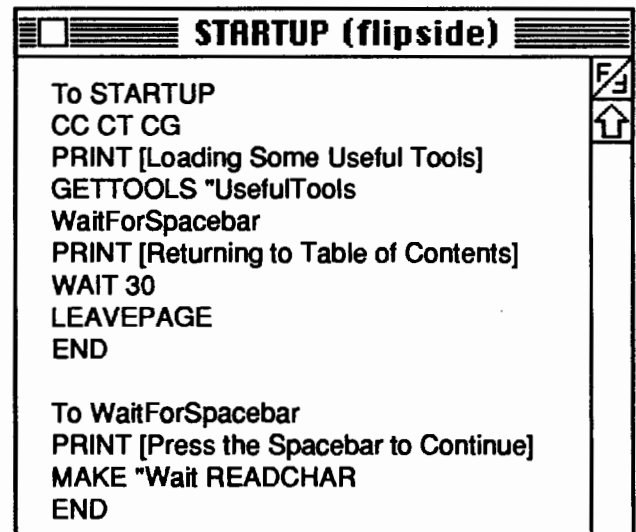
This variant might appear as though it is better because it is shorter. However, it is also more difficult to understand for the same reasons that Cartesian commands (`SETX`, `SETY`, `SETPOS`) in general are more difficult to follow than the intrinsic geometry of the turtle (`FORWARD`, `BACK`, `LEFT`, `RIGHT`, etc.). In cases such as this, readability is generally more important than brevity.

Extensibility

LogoWriter has a particularly useful feature that allows a page of tool procedures to be loaded so that they can be accessed from any Logo page. The `GETTOOLS` command can be used to load a page of useful tools (such as `Line` and `Over`) in the following manner:

```
GETTOOLS "USEFUL.TOOLS
```

LogoWriter also allows a special page called `STARTUP` to be created, which automatically is loaded when *LogoWriter* is first started. A special procedure called `STARTUP` may be created on the `STARTUP` page. When *LogoWriter* is started, the `STARTUP` page is automatically loaded, which in turn automatically runs the `STARTUP` procedure. The figure below illustrates a procedure we use to automatically load a page of useful tools when *LogoWriter* is started. (Note: the `LEAVEPAGE` command is only available in *LogoWriter* 2.0.)



A STARTUP Procedure on a STARTUP Page

The ability to create tools that behave as though they were built-in commands is one of the most important attributes of Logo. When a procedure such as `Line` is loaded with the `GETTOOLS` command, it is almost indistinguishable from a built-in command. This characteristic of Logo is referred to as *extensibility*. Prior to Logo, only a few arcane languages, such as *Forth* (a language used in astronomy to control the movement of telescopes), were extensible. Now, many software construction toolkits, such as *HyperCard* and *Toolbook*, also offer extensible environments.

LX: The Next Decade

When Logo was first developed for microcomputers, most Apple II computers had 48 kilobytes of memory, and it was necessary to upgrade them with an additional 16 kilobytes of memory to 64 kilobytes before they could run Logo. Today, the standard computers being purchased for schools, such as the Macintosh LC and the IBM Model 25, seldom have less than 1024 kilobytes of memory, and many have several megabytes of memory. Under these conditions it is possible



to enhance Logo in ways that would not have been possible with 64 kilobytes of memory.

To a certain extent, the computers used in departments of computer science provide a window on the future. A visitor to the M.I.T. Artificial Intelligence Laboratory in 1970s would have been able to see Logo controlling robot turtles that would appear in the public schools in the 1980s. In a sense, a visit to a computer science department provides a type of time machine that allows us to see what may be in the future. Computer science faculty are generally not using personal computers but instead are using larger workstations, such as Sun SparcStations and IBM RS-6000 computers with RISC architectures. The displays of such machines are generally at least the size of a page (8 1/2 by 11 inches) and often larger. An examination of these displays reveals that the users seldom run only one application at a time. Instead, they generally have several windows with different programs open simultaneously. One window may contain a telecommunications program, another window may contain a word processor, and a third window may contain a graphics program.

With several windows available at one time, users can readily move information and graphics from one application to another. We are curious as to what the potential uses of Logo might be in such an environment and hope to explore such applications in future col-

umns which will be titled "Windows on Logo." This title for future columns has a double meaning; it simultaneously refers to a window on the future and to the windows found on the screen of a graphics user interface (GUI). Already two versions of Logo are available that can be run in such an environment (*LogoWriter* for the Macintosh and *Object Logo*), and we expect that more will follow in future years. We are not sure what the future may bring, but we are certain that if it is as eventful as the last decade has been, it will certainly be interesting!

Glen Bull is an associate professor in the Instructional Technology Program of the Curry School of Education at the University of Virginia. Gina Bull is a system administrator in the Department of Computer Science at the University of Virginia. By day she works in a Unix environment, by night in a Logo environment.

Internet Addresses: GBull@Virginia.edu,
Gina@Virginia.edu
BITNET Addresses: GBull@Virginia,
Gina@Virginia


NEW Terrapin Logo for Mac UPGRADE!

Now you can upgrade to new Version 1.2 of Terrapin Logo for Macintosh.
With it, you'll get:

- **Cutting & Pasting** of Logo graphics to clipboard, scrapbook or windows
- **Automatic window refresh** when windows overlap
- **Print more easily** using menu options
- **Runs under MultiFinder**
- **20 useful Logo tools**
- **Runs on an AppleShare or MacJanet network** with Site License version

For complete information on the new features and tools in Terrapin's Logo for Macintosh upgrade, write or call. You'll get our usual helpful, friendly support.

How to upgrade: Send your old Terrapin Logo/Mac disk with \$25 for the first disk and \$7.50 for each additional disk returned to be upgraded. Site licenses can be upgraded by sending us a P.O. for \$100 noting your site license number.

 Terrapin Software, Inc. ♦ 400 Riverside Street ♦ Portland, ME 04103
207-878-8200 ♦ Fax: 207-797-9235

Lunch and Logo

by Chris Held

Dear Sharon,

I've been meaning to send you things our room has been doing in Logo and LEGO/Logo for years, but I keep forgetting.

It took a fairly direct, corresponding event to finally remind me.

Our class was working with probability in math. The kids had been throwing dice and recording their findings but were never getting a large enough sample to make the point. Enter Logo!

I wrote up a little program over lunch hour to share with the kids. As I walked to the office to duplicate it, I picked up my copy of *Logo Exchange* and it finally clicked! I could share this with other teachers.

In this program the turtle would roll the die and keep a running count of how many 1s, 2s, 3s, 4s, 5s, and 6s that it had rolled. We asked the turtle to roll the die many times. The final test was to ask it to roll it 6,000 times. Guess what the results were!

You guessed it! We finally began to approach a number large enough to show randomness (very close to 1,000 for each of the six faces of the die!)

I thought I'd share this with other teachers who might like to see another of the many ways our little turtle friend serves as a tool to help us learn more about numbers.

The following program is written in *LogoWriter*:

```
to play :number.of.rolls
  get.started
  throw :number.of.rolls
  give.results
end
```

```
to get.started
  ct
  make "ones 0
  make "twos 0
  make "threes 0
  make "fours 0
  make "fives 0
  make "sixes 0
end
```

```
to throw :number.of.rolls
  repeat :number.of.rolls [make "dice
    (1 + random 6) insert :dice
    keep.track]
end
```

```
to keep.track
  if :dice = 1 [make "ones (:ones +
    1)]
  if :dice = 2 [make "twos (:twos +
    1)]
  if :dice = 3 [make "threes (:threes
    + 1)]
  if :dice = 4 [make "fours (:fours +
    1)]
  if :dice = 5 [make "fives (:fives +
    1)]
  if :dice = 6 [make "sixes (:sixes +
    1)]
end
```

```
to give.results
  print []
  print [one]
  print :ones
  print []
  print [two]
  print :twos
  print []
  print [three]
  print :threes
  print []
  print [four]
  print :fours
  print []
  print [five]
  print :fives
  print []
  print [six]
  insert :sixes
end
```

Chris Held
Phantom Lake Elementary
1050 160th Avenue SE
Bellevue, WA 98008



		Computers			
		One (1)	Cluster (3-5)	Half Set (10-15)	Class Set (>20)
A C C E S S	Rare	?	?	?	?
	Occasional	?	?	?	?
	Frequent	?	?	?	?
	On Demand	?	?	?	?

Finding a Place for

Logo

by Mark Horney

I have noticed a disturbing trend as I work with the schools in my area. On the surface they seem to be making strong progress integrating computers into their curricula and the daily lives of teachers and students. Most of these schools have one or two labs of Apple II or Macintosh computers, and many individual classrooms have more computers permanently on hand. Each school has a full-time computer teacher, the school district has a plan for the use of technology, the faculty has received a reasonable amount of inservice training over the years, and whenever I stop by for a visit, the computer labs are always bustling with activity. While it can't be claimed that technology has revolutionized these schools, they certainly have passed the stage of technophobia where computers gather dust in the closet.

The problem? As I talk with students I find that few actually use computers very often and almost no one has ever heard of Logo.

I think the causes of this problem are straightforward. The school district decided children should learn about computers, so they provided the equipment and staff to put every student through a computer literacy class each year. In these classes, students learn such things as keyboarding, word processing, databases, robotics, desktop publishing, hypermedia, electronic studying, and, once in a while, even a little Logo. To

deliver this curriculum, the computer teachers use five of the seven hours that the computer lab is available in the school day. Every other computer application, every other teacher, and every other student project must fit within the remaining two hours of lab access time.

It would seem that the solution to this part of the problem is equally straightforward: cancel the computer literacy classes. There is certainly no point in giving students computer skills if they never have a chance to put the skills to use. Making this change, however, is easier said than done. Some of those skills—keyboarding, for example—are very useful if students are to be effective computer users, and it takes time for such skills to develop. Even if such skills were to appear magically in every student, there wouldn't be much change in the access students and teachers have to the building hardware. As the situation stands, the two open hours per day allow only three or four teachers access to incorporate computers into their curriculum in some significant way. If all seven periods were available, this number would increase to 10 or 15. In a medium-sized school, this means students might use the computers in one or two classes once or twice a week. There simply aren't enough computers around for them to have more impact than the little they already have.

Computers are, and I think will remain for the time being, a scarce resource. Teachers intent on using computers in a serious fashion for Logo based activities must compete for the use of those scarce computing resources. This implies that Logo itself must also compete against all the other computer activities from which teachers may now choose. In the past, when the competition was dull electronic workbooks or whiz-bang arithmetic shoot-em-ups, the argument for Logo was easy to make. Today, Logo must contend with more compelling uses for technology, such as telecommunications, hypertext, interactive multimedia, word processing, and a broad range of information organizers. Logo no longer faces the drill-and-practice straw man, and it must struggle for its place in the curriculum. This struggle may well be one for Logo's very existence.

To meet this challenge, Logo teachers must expand their repertory of Logo-based activities. They must find ways to incorporate Logo into every nook and cranny of the school and exploit available computer resources to the fullest extent possible. One tool for achieving this deep penetration of the curriculum is the Computing Resource Chart shown in the figure above. It displays two dimensions of computing resources—how many computers are available and how often teachers can make use of them. Logo teachers must find Logo activities for *every square* on the chart.

Some parts of the chart, notably the edges, are easy to fill. Teachers with only rare access to computers can still use them effectively for introductions and demonstrations. A physics teacher, for instance, can give his students a head start in understanding friction with just a demonstration of the classic Logo microworld DynaTrack. Obviously it would be better for students to explore this program themselves, but even a one-time demonstration would be useful.

The squares at the bottom of the chart are also easy to fill. Teachers with access to computers all the time—even if it's only a single computer—can generate many compelling uses for Logo. The traditional vision of student exploration of the Logo environment fits here. For this part of the chart, students have the time, possibly all day, all year, to learn with and about Logo. Whether they work alone or in groups, every day or just on occasion, Logo is always around, so it can be woven into the fabric of instruction.

A much harder task is finding appropriate activities for the central parts of the resource chart. Here, there are enough computing resources to allow teachers to do more than just introduce or demonstrate but not enough for computing to be a fundamental basis of the whole curriculum. Unfortunately, the middle of the chart is just where most teachers find themselves situated; therefore, these activities must be carefully con-

ceived and constructed. I believe activities must have four basic characteristics: they must be *integrated* with other parts of the curriculum, they must be *justifiable*, they must be (dare I use the word?) *efficient*, and they must be *adaptable* for use beyond their initial applications.

Logo activities must *integrate* well into other parts of the curriculum because it is likely they will be knitted together with other noncomputer activities. Students will work on alternative assignments while waiting their turn to rotate onto the computers, or while the teacher waits for a shot at the computer lab. Sometimes the on-computer and off-computer activities can be very closely related, such as when students compose and analyze procedures at their desk and only use the computers for entering and executing code. In general, the more similar and related the on-computer and off-computer activities are, the better. What should be avoided are the jarring transitions students face when moving quickly between widely divergent activities. Valuable computer time is wasted as students cope with the cognitive load of switching from one unrelated activity to another, not to mention the difficulties faced by harried teachers flitting from one station to another trying to render assistance.

Logo activities must also be *justifiable*. That is, these activities must have a justifiable place in the regular curriculum. Until curricula are restructured in some way that overtly includes the kind of procedural learning for which Logo is particularly appropriate, the time allocated to Logo activities must come from more traditional subjects. Students, parents, administrators, and colleagues must all see how and why Logo fits into the instructional plan. Without such justification, each of these groups can, and should, challenge how scarce computing resources are being consumed.

Logo activities must be *efficient*, and I mean this in a very practical sense. Consider, for example, all of the skills students need just to write a procedure executing the command: FD 100. They must know how to turn the computer on and off, how to move in and out of the editor, how to save and retrieve files from a disk, how to format a Logo procedure, and how to give execution commands. Students lacking any one of these skills are helpless. But having these process skills doesn't contribute anything to a student's understanding of FD 100, or Logo, or computer programming, or English, or math, or anything much at all. Thus, it's vitally important for skills such as these to be taught as efficiently as possible, particularly when computer resources are scarce.

Finally, Logo activities must be *adaptable* to different circumstances. This only makes sense: The more applications teachers can find for various student skills, the more widely these costs in time and resources can

be amortized. Furthermore, Logo concepts and skills can serve to foster an interdisciplinary connection among the separate parts of a student's education.

An example of a Logo activity that meets the requirements of being integrated, justifiable, efficient, and adaptable would be a Vector Construction Set. This set of tools would consist of procedures for doing four different vector manipulations:

1. Plot vectors, either individually or in a sequence.
2. Add or subtract vectors.
3. Convert between Polar to Cartesian coordinates.
4. Calculate angles between vectors.

Such a set of procedures could be used for exploration, as a problem-solving tool, or as a study aid in geometry, trigonometry, physics, or computer science. Students could write the procedures themselves, modify and extend a set of procedures begun by the teacher, or just use what the teacher provides as a tool.

Once each level of the resource chart given above has been completed with multiple activities targeted to all parts of the curriculum, teachers can use it to plan for the incorporation of Logo throughout the school day. The chart may also highlight the advantages of differ-

ent ways to make use of equipment within the building. Having more access to fewer computers may be preferable to concentrating computers in labs where they can only be used on occasion.

The chart will also be useful for long-range planners. With it they can see how making changes in computing resources affects what teachers do and how teachers interact with their students. The hope of the Logo community must be that this will compel planners to increase the importance of technology in the future of education and that it will demonstrate that Logo can have a vibrant, central role to play for a long time to come.

Mark Horney is an educational researcher and consultant at the University of Oregon. His work there involves all forms of educational computing, but particularly the areas of electronic studying and hypertext. Prior to his university work, Mark was a middle school math, science, and reading teacher and a high school computer science instructor.

Mark A. Horney, Ph.D.
1787 Agate Street
Eugene, OR 97403 USA
BITNET: mhorney@oregon

Introduction to Programming in Logo Using LogoWriter

Introduction to Programming in Logo Using Logo PLUS.

Training for the race is easier with ISTE's Logo books by Sharon (Burrowes) Yoder. Both are designed for teacher training, introductory computer science classes at the secondary level, and helping you and your students increase your skills with Logo.

You are provided with carefully sequenced, success-oriented activities for learning either *LogoWriter* or *Logo PLUS*. New Logo primitives are detailed in each section and open-ended activities for practice conclude each chapter. Both books sell for \$18.95 each plus \$4.50 shipping.

Keep your turtles in racing condition.

To order, contact:

ISTE, 1787 Agate St., Eugene, OR 97403-1923; ph. 503/346-4414

The turtle moves ahead.



Turtle Talk

by Douglas H. Clements

Logo Exchange: Thank you for agreeing to talk with us for our 10th anniversary issue.

Turtle: It's about time we set the research record straight. There's been a lot of misunderstanding.

Logo Exchange: Well, give us the big picture first: Are computers educationally effective?

Turtle: Meta-analyses—reviews of research that statistically combine the results of numerous studies—reveal that students make significant learning gains using computers (Roblyer, Castine, & King, 1988). Some reports suggest that the greatest gains have been in mathematics skills. These gains are consistent across schools, grades, and testing instruments and have been judged to be both educationally significant and cost feasible.

Logo Exchange: So, computers have been proven to be effective?

Turtle: No, this doesn't mean that the use of any software under any conditions leads to such gains. Their effectiveness depends on the quality of the software, the amount of time the software is used, and the way in which it is used.

Logo Exchange: When you say "quality of the software," do you mean different types of software, like Logo versus traditional approaches, such as computer-assisted instruction or CAI?

Turtle: Not really. Interestingly, several different types of approaches have been shown to be effective, and both lower order and higher order abilities can be enhanced. Within each approach, there are effective techniques. For example, there are effective ways of providing practice that few CAI packages use.

Logo Exchange: Such as...?

Turtle: Adaptive instruction—in the traditional sense. One example is the technique of "increasing ratio review." In simple practice, a missed item often is placed at the end of the list for additional practice. This is characteristic of both traditional computer drills and flashcard drills. Unfortunately, if the list is

long, the student forgets the answer. If the list is short, the student is not assisted in remembering the item over longer time periods. Instead, using an adaptive technique called increasing ratio review, the computer inserts the missed item into the list at several places, such as the 3rd, 7th, and 13th item. This technique has been shown to increase students' achievement without increasing the total time they work on the task (Siegel & Misselt, 1984). Hey! Wake up!

Logo Exchange: Oh, sorry, I dozed off. Are you saying that it doesn't matter if you use CAI or Logo, as long as it's good quality?

Turtle: No! I'm saying that research indicates that both approaches, done right, can raise achievement scores. But that's just the beginning. You've got to consider what *your goals are*. Recent recommendations for school reform, such as the National Council of Teachers of Mathematics' (1989) *Curriculum and Evaluation Standards for School Mathematics* demand that other approaches be emphasized.

Logo Exchange: Why is that?

Turtle: Well, CAI as a main approach to teaching just isn't consistent with this view of education. In the words of another such report:

In reality, no one can teach mathematics. Effective teachers are those who can stimulate students to learn mathematics. Educational research offers compelling evidence that students learn mathematics well only when they construct their own mathematical understanding. (National Research Council, 1989, p. 58)

Logo Exchange: It seems clear how Logo's vision fits that constructivist perspective. But shouldn't we be looking at newer software developments?

Turtle: Call me biased, but this is one issue that really gets under my shell. First, Logo is not "old" software; it is evolving. I've heard people—many who should know better—say that newer software is easier to use. The epitome of that misunderstanding was a consultant who said, "I don't suggest Logo



anymore, now that we have Dazzle Draw." We know that the benefit of Logo is that it encourages students to think about what they are doing in creating geometric figures. It helps them translate their intuitions into mathematical concepts and link these two (Clements & Battista, in press). I've seen packages that make drawing a lot easier, but *the point is not the drawing, it's the reflection on doing the drawing*. A lot more thought has to go into deciding what should be "easy" and what should remain a struggle, in the positive sense of the word. You don't always win the race by going faster, you know.

Logo Exchange: Yes, but you must at least admit that interfaces have improved.

Turtle: Listen, I'm not green due to envy. First, as I've already said, menus are just as applicable to Logo as they are to any other computer program. Second and more important, you've got to consider the role of language in education. Once again, making things too "easy" is not the point—using only "point and click" can lead to only point and click thinking. The Bulgarians, who are developing a series of Logo microworlds for secondary mathematics (Clements, 1991), put it well:

At the heart of this system is the philosophy that in order to do mathematics students must have LANGUAGE to express their mathematical ideas and that the notion of DEFINITION is so central to mathematics that it cannot be ignored in mathematical education....The most important and fundamental mathematical activity is dealing with notions—mainly composing and decomposing of notions—which definitely needs a language. (Filimonov & Sendov, 1990, p. 1)

Logo Exchange: But surely you agree that students should use a lot of different types of programs.

Turtle: No problem...except that we need to be concerned that whatever programs they use become tools for thinking. To engage students in substantive intellectual enterprises, we need to develop their mastery of a flexible, extensible tool. Extensions to this tool are welcome, but we need to be wary of providing a potpourri of applications with no internal coherence. Research comparing Logo to a set of utility and problem-solving programs demonstrated that stronger feelings of control and mastery emerged with Logo. Logo alone provided a coherent intellectual tool (Nastasi, Clements, & Battista, 1990).

Logo Exchange: But couldn't another flexible tool replace Logo? For example, a previous column here indicated little difference between research on different computer languages (Clements, 1990a).

Turtle: That's not quite true. Remember that students programming in Logo rather than in BASIC initiated more of their own explorations and interacted more with the teacher. These are the type of activities the reform movements are calling for. More important, the author actually forgot to say that one meta-analysis showed that Logo studies overall have stronger effects on problem solving than studies using other computer languages (Liao & Bright, 1989)! Of course, he's only human. And these studies weren't using new versions of Logo—just watch when the mutant turtles start emerging.

Logo Exchange: What else has changed in Logo research? Take an historical perspective.

Turtle: Too many early Logo researchers believed that Logo alone would lead to increased achievement, problem solving, and so on. Listen, I never said I could—or even *wanted*—to work alone with students. Logo was supposed to be a tool for building new mental frameworks...for manipulating representations of mathematical ideas.

Logo Exchange: So these are two different ways of using Logo?

Turtle: Yes. The author of this column has called the first way the "exposure approach" and the second way the "conceptual framework approach" (Clements, 1990b). But remember that he's at a university and needs to make up terms like that to stay employed.

Logo Exchange: Does the research on the two approaches also differ?

Turtle: Research on the "conceptual framework approach" is much more positive. For example, appropriate use of Logo helps students analyze and understand the properties of geometric figures (Clements & Battista, 1989; Clements & Battista, in press). Still, misconceptions persist for many students, and what is learned in Logo is not always transferred to other situations (Clements, in press).

Logo Exchange: Why so?

Turtle: I talk mathematics. But students don't always listen (Leron, 1985). For instance, they often rely too heavily on visual clues. You hear them saying, "It looks like about 100."

Logo Exchange: So Logo has limitations?

Turtle: Sure, considered alone. But I just told you, I never intended that! When I told Seymour to use the samba schools thing, I thought I made that clear (Papert, 1980, p. 178). Hey, I don't mean to snap your head off, but this has constantly been misconstrued.

Logo Exchange: So what is this approach?

Turtle: Well, research is consistent on this point. In successful Logo projects, teachers don't get Logo loaded and then withdraw into their shells. They think about the Logo tasks. They help students abandon visual strategies by presenting problems that require analytic solutions. They talk to students about their work and encourage them to talk to each other. They mediate their students' experiences with Logo. As a result, the students reflect on their work and link what they know in and out of the Logo context (Clements & Battista, 1989).

Logo Exchange: Does this approach have a name?

Turtle: Trust me, it gets worse. This he called the "mediated conceptual framework approach."

Logo Exchange: Tell me more about the links.

Turtle: These can be forged in two ways. Logo can be used to help deliver the traditional curriculum. Or, Logo can be a catalyst for educational reform.

Logo Exchange: We're back to the constructivist position again, then?

Turtle: Yes. Logo is mentioned repeatedly in such reform documents as the *Standards* that I mentioned previously (National Council of Teachers of Mathematics, 1989). But what I find more interesting is that there is an even more fundamental consistency when Logo is *not* mentioned by name than when it is.

Logo Exchange: I'm sorry....

Turtle: Consider this quote:

Computer microworlds such as Logo turtle graphics and the topics of constructions and loci provide opportunities for a great deal of student involvement. In particular, the first two contexts serve as excellent vehicles for students to develop, compare, and apply algorithms. (National Council of Teachers of Mathematics, 1989, p. 159)

This is fine. But the full power of Logo is exemplified better in other quotes that do not mention Logo specially:

In learning geometry, children need to investigate, experiment, and explore. (p. 48)

When mathematics evolves naturally from problem situations that have meaning to children and are regularly related their environment, it becomes relevant and helps children link their knowledge to many kinds of situations. (p. 23)

A major goal of mathematics instruction is to help children develop the belief that they have the power to do mathematics and that they have control over their own success or failure. (p. 29)

Logo Exchange: I see your point. These seem to express the Logo philosophy.

Turtle: Right! The Logo and constructivist philosophies have the same two major goals. First, children should develop mental structures that are more complex, abstract, and powerful so that they are increasingly capable of solving a wide variety of meaningful problems. Second, students should become autonomous and self-motivated. Such students believe that they do not "get" knowledge from their teacher so much as from their own explorations, thinking, and participation in discussions. They see their responsibility in the classroom not so much as completing assigned tasks but as making sense of and communicating about the subject matter (Clements & Battista, 1990).

Logo Exchange: How about the teacher?

Turtle: Accepting the "mediated conceptual framework approach," the role of teachers is also the same. They guide and support students' invention of viable ideas rather than transmit "correct" adult knowledge. Not everyone recognized the importance of this mediation right away. The usual author of this column fell all over his feet in his rush to describe the role of the teacher in his studies of Logo and higher order thinking—after he figured out how important it was.

Logo Exchange: So, using this approach, Logo is consistent with recent philosophies of education and recommendations for reform.

Turtle: Yep.

Logo Exchange: And research substantiates that such use of Logo does indeed increase children's mathematical and problem-solving power?

Turtle: Not only that. One of the surprising research results is that the largest, most consistent benefits are in the social and emotional domain (Clements, 1986a; 1990b). Students cooperate more in Logo environments and they cooperate on *learning*. They also disagree more—but they disagree about *ideas*, and they are more likely to successfully resolve these disagreements (Clements & Nastasi, in press; Nastasi, Clements, & Battista, 1990). They learn to listen, criticize in a constructive fashion, and appreciate the work of others (Carmichael, Burnett, Higginson, Moore, & Pollard, 1985).

Logo Exchange: How about the emotional side?

Turtle: They also increase their self-confidence. Special needs students gain prestige and respect from their peers. Most important, they learn to judge situations for themselves and accept responsibility for their actions (Clements, 1986b).

Logo Exchange: Thank you. I know you have a screen appearance that you must get to. Can you summarize for us?

Turtle: I don't mean to be hard shelled about this, but the research does suggest the following. If you want a safe and relatively easy path, choose appropriate CAI software. You'll probably increase achievement. If instead you want to effect substantive change in the quality of students' educational experiences, use Logo with the mediated conceptual framework approach...but be ready to work hard at it.

Logo Exchange: Is this the final word on research?

Turtle: There can be no final word. Research and education are dynamic, growing. It took a long time to learn what we know about Logo, and we still have a lot to learn. But as my grandfather told me, slow but steady wins the race. A gradual accumulation of evidence tells us that Logo can help students achieve new vistas. Seymour's right on this one—other educational movements have attempted project-based approaches before, but usually the activities were unable to handle the more formal aspects such as mathematics. Logo and I provide a context in which mathematics, science, and powerful ideas can be

used to achieve personal goals. In this way, we also serve as a catalyst for educational reform.

References

- Carmichael, H. W., Burnett, J. D., Higginson, W. C., Moore, B. G., & Pollard, P. J. (1985). *Computers, children and classrooms: A multisite evaluation of the creative use of microcomputers by elementary school children*. Toronto, Ontario, Canada: Ministry of Education.
- Clements, D. H. (in press). Research on Logo: Effects and efficacy. *Education*.
- Clements, D. H. (1991). A new vision of Logo in the secondary school. *Logo Exchange*, 9(6), 24-29.
- Clements, D. H. (1990a). Logo or "Logo-like." *Logo Exchange*, 9(1), 33-34.
- Clements, D. H. (1990b). Turtle soup: A beginning look at Logo research. *Logo Exchange*, 9(1), 32-35.
- Clements, D. H. (1986a, 1987). Logo: Search and research. *Logo Exchange*, 5(3,5).
- Clements, D. H. (1986b). Logo: Search and research. *Logo Exchange*, 5(4).
- Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, 20, 450-467.
- Clements, D. H., & Battista, M. T. (1990). Constructivist learning and teaching. *Arithmetic Teacher*, 38(1), 34-35.
- Clements, D. H., & Battista, M. T. (in press). Geometry and spatial reasoning. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching*. Reston, VA: National Council of Teachers of Mathematics/Macmillan.
- Clements, D. H., & Nastasi, B. K. (in press). Computers and early childhood education. In T. Kratochwill, S. Elliott, & M. Gettinger (Eds.), *Advances in school psychology: Preschool and early childhood treatment directions*. Hillsdale, NJ: Lawrence Erlbaum.
- Filimonov, R., & Sendov, B. (1990). *A brief design rationale for the plane geometry system*. Sofia University, Bulgaria.
- Leron, U. (1985). Logo today: Vision and reality. *The Computing Teacher*, 12(5), 26-32.
- Liao, Y.-K., & Bright, G. W. (1989). Computer programming and problem solving abilities: A meta-analysis. In W. C. Ryan (Ed.), *Proceedings of the National Educational Computing Conference* (pp. 10-17). Eugene, OR: International Council on Computers for Education.
- Nastasi, B. K., Clements, D. H., & Battista, M. T. (1990). Social-cognitive interactions, motivation, and cognitive growth in Logo programming and CAI problem-solving environments. *Journal of Educational Psychology*, 82, 150-158.



- National Council of Teachers of Mathematics. (1989). *Curriculum and evaluation standards for school mathematics*. Reston, VA: Author.
- National Research Council. (1989). *Everybody counts*. Washington, DC: National Academy Press.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Roblyer, M. D., Castine, W. H., & King, F. J. (1988). *Assessing the impact of computer-based instruction: A review of recent research*. New York: Haworth Press.
- Siegel, M. A., & Misselt, A. L. (1984). Adaptive feedback and review paradigm for computer-based drills. *Journal of Educational Psychology*, 76, 310-317.

Douglas H. Clements, associate professor at the State University of New York at Buffalo, has studied the use of Logo environments in developing children's creative, mathematics,

metacognitive, problem-solving, and social abilities. His book *Computers in Elementary Mathematics Education* (published by Prentice-Hall, 1989) emphasized Logo. Through a National Science Foundation (NSF) grant, he codeveloped a K-6 elementary geometry curriculum, *Logo Geometry* (published by Silver Burdett, & Ginn, 1991). He is currently working on a second NSF-funded project to develop a full K-6 mathematics curriculum featuring Logo.

Douglas H. Clements
State University of New York at Buffalo
Department of Learning and Instruction
593 Baldy Hall
Buffalo, NY 14260

CIS: 76136,2027 BITNET: INSDHC@UBVMS

Getting from here...

with ISTE's new School Restructuring Tool Kit.

A research-based collection designed to help you and your school district implement the process of restructuring. The Tool Kit consists of 15 reports, programs, journals, books, and videotapes that provide:

- Examples and studies of successfully restructured schools, classrooms, and media centers
- Classroom-tested inservice programs
- Long-range proactive planning for integrating technology into classroom curriculum
- Step-by-step procedures and materials for program evaluation
- Site reports, research summaries, new teaching techniques

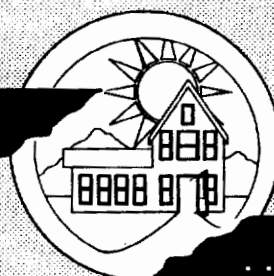
Order today and create your school's vision with ISTE's research-based School Restructuring Tool Kit.



Price: \$310.00 plus 19.25 UPS shipping (First Class mail \$33.00) (member price \$279.00 plus shipping)
Note: All 15 components of the Restructuring Tool Kit are available individually.

To order contact: ISTE, 1787 Agate St., Eugene, OR 97403-1923; ph. 503/346-4414

Book prices and shipping charges are subject to change without notice. Any discrepancy in an order will be billed or credited to customer.



...to there



A Night in the White House

by Janet Koske

To: Dr. Judi Harris
From: Janet Koske
Subject: Secret White House Visitor

President George Bush has said that he wants to be remembered as the "education president." Two weeks ago, President Bush announced that he was going to learn how to use a computer. He said that he is not computer literate, but feels it is an important technology for today.

(Psst...)

I happen to know what happened in the White House the night before, and how it led to this announcement. Please read on.

Listen, Americans!
Lend me your ear:

There's an education program
of which you should hear.

I was working in my study
late one night,
While my advisors were discussing
America's plight.

Our schools were in need
of major reform.
We MUST do something to
take them by storm.

I leaned back in my chair
for a moment's rest,
While the committee droned on
and on...about tests.

I let my eyes close
for just a quick nap,
When suddenly I awoke
with a resounding SNAP!

I opened my eyes
and felt very strange,
I strolled to the window
and thought about change.

I undid the latch
and stared at the trees,

They were swaying and rustling
back and forth in the breeze.

Then,
without feeling a care,
I opened the window,
needing fresh air.

The breeze rustled the curtains,
and a few papers did blow.
It filled my head with the
faint sound of...Logo.

My advisors were still talking
as I turned around.
It was obvious that they
had not heard the sound.

They picked up their studies,
then started for the door.
"Good night, Mr. President. We'll see you
tomorrow
with statistics and more."

I sank into my chair,
being somewhat weary.
I looked at my desk,
and felt overwhelmingly dreary.

The computer on my desk
in front of me
Was simply there
for people to see.

This new machine,
I had yet to learn,
Yet, this is the technology
to which our schools will turn.

It seemed so foreign, so complicated,
so...hard,
Yet entire programs could run
from this five-inch card.

My eyes were drooping;
I would soon be asleep.
Computer technology would just
have to keep.

Just as my world
fell silent and dark,
I saw something glisten.
It looked like a spark.

I opened my eyes
only to see
A kindly old gentleman
standing by me.

He reached in his pocket
for a pinch of wonder dust,
Told me to "Relax!"
Said that it was a MUST.

His face was weathered,
but his eyes were quite kind;
Even if I called him "Santa,"
he probably wouldn't mind.

"Santa?" I said,
knowing that wasn't right.
"Seymour!" he laughed,
flashing a smile so bright.

"Yes, Seymour's my name,
and Logo's my game.
I'm really immersed,
as some folks say,
And, Mr. President...sir,
I'm gonna show you how to play!"

He motioned me over
to the computer screen,
"But, Seymour," I said,
"I can't work this machine!"

"I'm not even sure I can
do this at all.
It's like basketball for me;
I'm just NOT that tall."

The man didn't shrug;
he didn't even blink.
He put a turtle on the screen,
then asked for a drink.

As he took his glass
and sat in my chair,
I sat and pondered
the turtle that was there.

"What does it do?
What do I do with it?"
I tried to be brave,
But I almost had a fit.

Yet I sat there and listened
as the kind man explained,
"There's commands to be used,
but they're easy and plain."

"There's RIGHT and LEFT,
to make the turtle turn,
There's REPEAT for squares,
And circles to learn."

"There's also FD and BK,
and don't be surprised
If you can code powerful ideas
with no special knack."

My squares were beautiful;
my circles terrific!
WOW! This was fun!
I'll bet Logo could be learned by...everyone!

Seymour began to tell me how he adapted Logo
for learning and play.
His hope was that teachers and students
would find a better way.

He believes that each of us
can be a great thinker,
If we give ourselves enough time
to...tinker.

Children are both great thinkers
and great tinkerers.
They have so much patience, you see;
That is usually lacking in you and me.

Adults? Well, we hurry,
And often we run.
We have so many things
we simply must get done.

As Seymour talked,
I was having a ball.
My turtle was drawing shapes
both large and small.

Usually, the teacher has the text,
and you, the student, had better pay attention.
If you're doodling or thinking (like I was),
you might get detention.

A teacher usually wants the answer that is right,
Not a guess...or a try.
So shape up and get the answer right,
or you might want to cry.

A Logo-like class
is really much different,
Although activity is facilitated,
productive, and efficient.

The students are invited to try,
to reason, to guess.
They actually THINK their way
out of each mess.

Logo-like teaching
is not so hard to grasp.
Many progressive educators
have tried it in the past.

"What else can I do?"
I decided to ask,
"I think I have really mastered
this task."

"I want to do something personal,
all for me; MINE...that no one can take.
Something to which
I can personally relate."

Seymour raised an eyebrow
then smiled as he winked.
"That's the whole idea, Georgie!
Now then...let's think."

He gave me a hint
and I was off and away.
I now had four turtles
with which to play.

Just look at the math
the kids would learn!
And all this was done
with a turtle turn!

"Seymour, this is great...
really great stuff!
But, changing our teachers
may be pretty rough stuff."

"A Logo-like classroom
may not be accepted.
The way schools do things
is pretty protected."

"Administrators may not think
Logo's so good.
They'll say the kids aren't learning
all that they should."

"New ideas are hard to implement.
The school curriculum is like..."
"In cement?" replied Seymour,
obviously having heard all this before.

"Tell me more about you, Papert;
Were you considered a crackpot?
Just how did you get
the info you got?"

"I studied children
over the years.
I watched them laugh,
and saw their fears."

"Child-centered learning
is really the key.
I've recorded the results of my studies
at M.I.T."

"Logo-like learning
and teaching, too,
Is always unfinished;
always more to do."

"There's room for something different,
more ideas you can add;
Maybe make the program better
by just...a tad."

"Kids have powerful ideas
that really bring results.
When all that they are thinking
Is not structured by adults."

"Logo-like teachers
can surely take the lead.
Their students learn to problem-solve
with skill and with speed."

"Well, George, it's getting late
and I really must go.
Tomorrow I begin a new venture
with...Nintendo."

"I hope that I've shown you
something you haven't thought about before.
And there are still many powerful ideas
left to explore."

"Adopting a Logo philosophy
is not really taking such a chance.
It need not replace, at first,
but will creatively enhance."

My turtles were whizzing
around and around.
I even played music
to give them a sound.
I yelled, "Hey Seymour!"
as I turned in my chair,
And noticed that Father Turtle
was no longer there.

I heard knocking and calling
coming from beyond my door.
"Mr. President, Mr. President...
Are you ready for more?"

My advisors entered my office
one by one,
And I went to the window
to let in the sun.

It was a bright, warm day,
and I told my committee
That I had something
to say.

"Sir," they said,
"Are you all right?"
"You look as if
You've been up all night!"

"I'm great, you guys.
I've never been better.
I've just had a mindstorm,
And you can quote me to the letter."

"Our education system really WILL change
if we're brave enough to do it.
It will take courage and fortitude,
and this administration is certainly up to it."

"We'll make a right turn
of 180 degrees,
Then FORWARD in action,
as far as we please."

"Now let's get some breakfast
as I tell you about it.
Though my sanity may seem questionable,
please do not doubt it."

As we made our way slowly
to the room's door,
I felt exhilarated
from the night before.

Logo was no longer a whisper;
it was a clear shout,
And I couldn't wait to tell my committee
what it was all about.

Janet Koske is a graduate student at the University
of Nebraska at Omaha who recently became
"hooked" on Logo. This poem constituted one of
her final projects for TED 8580, "Logo and the K-12
Curriculum."

Janet Koske
812 Elmhurst Drive
Papillion, NE 68128



Remember When....

Take a stroll through LX yesteryear with these pages reprinted from our first year of publication.

The National



Exchange

Volume 1 Number 1

FORWARD 100!

September 1982

Mindstorms — Don't Miss It

If there is to be a classic book about LOGO, it is sure to be Mindstorms by Seymour Papert. It is practically impossible to know anything about LOGO without having heard something about Mindstorms, since Dr. Papert was one of LOGO's founding fathers some twelve years ago.

If you have not yet read this book, don't miss it. The essence of the LOGO learning philosophy is contained therein, written in an easy, straightforward style. But that is not all that awaits you.

Papert develops many powerful ideas which are of importance and concern in today's education systems. Most deal directly with the role of computers in the classrooms.

His main thrust seems to be in the direction of "natural learning," a philosophy he gained from direct study with Piaget. He views LOGO as an ideal vehicle to provide the children with a learning environment in which they can explore, experiment, and play, using the computer and its capabilities in self-directed ways.

He also looks to the future, and sees the need for a revolution in education as we know it today! The QWERTY phenomenon is his name given to any process which is so deeply established that, regardless of any suggested improvements or developments, no changes are permitted. "We have always done it this way," is the motto of the QWERTY-ite. Does this idea sound familiar? Perhaps you will find this part of Mindstorms extremely stimulating!

If you are looking for LOGO lesson plans, you will not find them in this book. But, you will find several concrete ideas and specific philosophical concepts upon which you can easily base your LOGO plans, activities, and curricula. And you will gain an appreciation for the incredible potential of LOGO, the next giant step in education!

Mindstorms: Children, Computers, and Powerful Ideas, Seymour Papert, Basic Books, New York, 1980.

LOGO Reinforces Geometric TABS Skills

by
Linda Nix

The computer was a new addition to our school this past year, and, as the fifth grade teacher, I was anxious to use it as much as possible as an extension of my regular teaching strategies. In particular, I wanted to use it to reinforce required TABS skills. I found LOGO to be an exciting and effective way to develop recognition of geometric shapes.

My technique was for students to recreate geometric shapes by using LOGO and to record their programs in a notebook for future use by other students. Since LOGO was new to both my students and me, I started slowly with a teacher-directed activity demonstrating computer commands in basic LOGO for the turtle mode. The students sat on the floor in front of the computer and took turns giving directions to the computer, which I typed in, to draw a particular shape.

Several students, because of their scouting activities, quickly saw the relationship between the degrees on a compass and the size of the turns required to make the correct angles.

After drawing a square, a rectangle, and a triangle, we drew the outline of a house. It was exciting to observe the group interaction that took place. Even the shyest child became involved.

The students were then paired with partners and given the assignment to write directions for the computer for other geometric shapes. They worked at their desks using paper and pencil. Some used a compass to help them decide on the size turns to have the computer make.

During this time, it was interesting to watch the decision making process that was taking place and the student interaction. Many students worked together beautifully. For others, it became a test of patience and self-control.



The National LOGO Exchange, copyright 1982 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, \$25 per year, \$45 for two years, mailed FIRST CLASS from Charlottesville, VA. Add \$5 per year for addresses outside the United States. Single copy price \$3. Opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Address all editorial correspondence to the editor at The National LOGO Exchange, Post Office Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.....Tom Lough

WELCOME!

Welcome to The National LOGO Exchange, a monthly newsletter for LOGO teachers and parents! We hope that you will be able to make use of the tips and techniques which are presented to you here.

In each issue, we will have columns to help you develop LOGO plans and activities. Book reviews and critiques will be included, as well as comments on LOGO products we have reviewed. Important features will be articles written by classroom teachers and parents, reporting on their LOGO activities and discoveries.

We also hope that you will be willing to share your experiences and lessons in LOGO with your fellow readers. The future for LOGO is so exciting! Here is an innovation which is capable of changing education as we know it today. How its use develops will determine its effectiveness, to a large extent. You are in a position to influence that development by reporting your LOGO successes to others so their children can also benefit. Why not write your LOGO article today?

Here at The National LOGO Exchange, our motto is FORWARD 100! This reflects our enthusiasm toward LOGO and its role in education. We are very excited about the potential represented by LOGO. We want to press on in a FORWARD direction with our efforts to bring LOGO to a position from which it can influence our country's children. We want to give 100% support to you, the LOGO teachers and parents, as you work with LOGO in the classrooms and homes. Please feel free to adopt this motto as your own, for LOGO, and for life!

FORWARD 100!

Tom Lough
Editor

You're Invited!

The avalanche is beginning! LOGO is on the way ... and it has the potential to revolutionize education as we know it!

But, the teachers across the country need your help. They are beginning to experience challenges which you already have met and fears which you already have faced down. Would you like an opportunity to be of service to untold numbers of fellow LOGO teachers and their students? If so, then consider writing a short article about an incident from your LOGO experience for The National LOGO Exchange.

This monthly newsletter presents articles written by LOGO teachers for LOGO teachers. We are looking for short (500 words or less), snappy, practical pieces about specific LOGO incidents from which you learned about yourself and how you were teaching. Do you have a neat technique or method for encouraging children to explore a certain concept? What was your very first day with LOGO and children like? "If I had it to do all over again, I would be sure to ... " What are your favorite programs developed by your students? How did you help the discover the difference between "variable and "variable?

You could double the number of such questions in less than 15 seconds, I'm sure. Why not take a few minutes to write about one or two, or more?

And please do not limit yourself necessarily to those things which worked. The beginning LOGO teachers need to know what things might not work as well!

Parents, we need to hear from you, too! What are you experiencing with LOGO? What are your concerns? When you observe how your child is thinking when working with LOGO, what are your thoughts? Do you have any ideas or suggestions which teachers might find helpful for their LOGO work?

Mail your articles to The National LOGO Exchange, Box 5341, Charlottesville, VA 22905.

We thank you and look forward to hearing from you. Your fellow LOGO teachers thank you and look forward to reading about you. The children thank you and look forward to benefitting from you.

FORWARD 100!

Meet Logo's Best Friend.

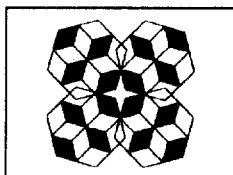
NEW!

Delta Drawing® Today Software for Exploration and Discovery

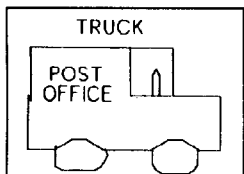
With Delta Drawing Today, students of all levels can create their own beautiful pictures. For example:



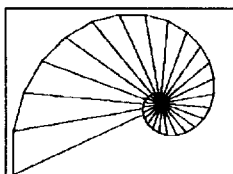
Lettering & Numbering



Four-Way Symmetry
(kaleidoscope)



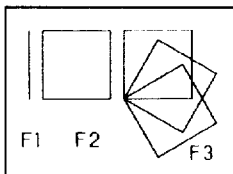
Drawn by 7-year-old David.



Log Spiral



Single Keystroke Scaling



Single Keystroke Macro
Generation

It's Delta Drawing Today, the perfect pre-Logo program and supplement to Logo, used by thousands of educators worldwide. It even has turtle graphics! Delta Drawing is a curriculum integrator which helps build basic skills in art, math, letters, computer literacy and more.

The original classic dramatically enhanced. Since 1982, the original Delta Drawing has been student tested and teacher approved. And now we are introducing this new, award-winning software for exploration and discovery in the 1990's.

Drawing to program versus programming to draw. That's the difference between our software and Logo. Delta Drawing records a student's keystrokes while he or she draws. By pressing T for Text screen, a student views the recorded commands which make up a program. One can then either edit this program, alter the drawing, or switch back and forth.

Students find Delta Drawing's phonetic keystroke commands intuitive and easy to master. Drawing occurs with no syntax, no numeric arguments, no carriage returns, no error messages – and, therefore, no frustration.

Indeed, Delta Drawing is the ideal learning tool for early education, slower learners and special education. It helps build kids' writing skills and self-esteem. At the same time, advanced students will love its randomizer, animation, right-left symmetry and accumulator variable capabilities.

Delta Drawing is amazingly easy to use. With single keystroke commands, students can draw in 15 colors and print in color, b/w or shades of grey.

You'll use Delta Drawing Today from September to June, year to year, grade to grade. And our practical multi-boot program diskette (classroom license) makes it an even greater value. Plus...

- ▶ Delta Drawing runs on all Apple II's, Apple IIGS's, IBM's, Tandy's, compatibles – and most popular printers.
- ▶ Comes with a brand-new teacher's manual and on-line HELP system that make it simple to learn and use.
- ▶ Includes a deluxe three-ring binder with 10 laminated activity cards, black-line summary masters and a toll-free support line.

Where learning becomes play...Delta Drawing Today. Go ahead. Introduce your students to Logo's best friend. Try the software program Apple Computer recently listed on the Honor Roll of Apple II classics.

Try Delta Drawing at no risk whatsoever with our 30-day money-back guarantee.

To order, return the coupon below or call us today toll-free at

800-395-5009.

Delta Drawing®
Software for Exploration and Discovery Today

Rated ★★★★★ by InCider/A+

YES!

ACTION REQUEST FORM

Send me the new, dramatically improved version of Delta Drawing Today in the quantity indicated below. I understand that if I'm dissatisfied for any reason whatsoever, I may return my software package within 30 days for a full refund.

Name _____ Title _____
School/Organization _____
Address _____
City _____ State _____ Zip _____
Phone # (day) _____ (eve) _____

Hardware and Media Selection

- ☐ Apple II or Apple II GS ☐ 5-1/4" Disk ☐ 3-1/2" Disk
☐ IBM, Tandy, 100% Compatible ☐ 5-1/4" Disk ☐ 3-1/2" Disk

Quantity

_____ Delta Drawing Today Packages for only \$69.95 ea. \$ _____
Add \$4 per copy for shipping and handling. \$ _____
MA residents add 5% sales tax. \$ _____
Total your order here: \$ _____
☐ Enclosed is my check payable to Power Industries LP.
☐ Please charge my: ☐ MasterCard ☐ Visa
Card # _____
Exp. Date _____ Signature _____

Call, return by mail or FAX today to: Power Industries LP, 37 Walnut Street, Wellesley Hills, MA 02181 • 800-395-5009 • FAX 617-235-0084

Volume 10, Number 1

by Dennis Harper

Logo Exchange Continental Editors

Africa

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

Asia

Marie Tada
St. Mary's Int. School
6-19, Seta 1-chome
Setagaya-ku
Tokyo 158, Japan

Australia

Jeff Richardson
School of Education
GIAE
Switchback Road
Churchill 3842
Australia

Europe

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

Latin America

Jose Valente
NIED
UNICAMP
13082 Campinas
Sao Paulo, Brazil

Volume 10, Number 1—this simple designation has a special ring to me (as it probably does to other long time *LX* columnists). By my count, there have been 104 different issues of either *Logo Exchange* or *International Logo Exchange* sent to enthusiasts around the world. So many memories come flooding through my mind: midnight calls to Kyoto, Buenos Aires, and Amsterdam gathering articles to meet deadlines, for two years walking through the hot humid afternoons to get columns into the mail in Singapore and Malaysia, trudging through snow and freezing weather to drop articles into the post office in Helsinki, Finland, and plugging my computer into a neighbor's generator in order to meet an *LX* deadline after Hurricane Hugo devastated the Virgin Islands.

During the past year I traveled to Sydney, Australia, for the World Conference on Computers in Education as well as to Europe to visit universities. These have certainly been exciting times for Logo internationally, but what about the next ten years? Unfortunately, I see some disturbing trends taking place.

When international Logo news first began to be reported in *LX* in January, 1983, excitement abounded in many schools throughout the world. Now most reports that I get seem to be from university professors doing small research projects with a small number of students. Many ministries of education have not included Logo or Logo-like activities in their plans. Countries promising massive Logo infusion into their curricula in the mid-1980s are no longer doing so. For example, the Netherlands spent much effort establishing the Logo Centrum Nederland (LCN) in 1982. Today only a few schools in that country regularly have students using Logo, and LCN spends less than five percent of its time and resources on developing or promoting Logo.

Bulgaria's exemplary efforts in introducing Logo into its schools has come to a near halt. The Logo project supported by Seymour Papert in Senegal has shut

down, with very little Logo use in the schools of that African nation. Indeed, you would be hard pressed to find any school child using Logo on a given day on the entire continents of Africa or Asia. I suppose much of this lack of Logo activity stems from lack of money and political instability. Much, but not all!

There still are some exemplary efforts and successes happening outside the United States. Volume 10 will continue to report on Logo activities throughout the world. One continent that has shown continued Logo growth is Australia. *Logo Exchange* would like to welcome back as our Australian editor, Dr. Anne McDougall, of Monash University. Anne was our original Australian editor in 1983 but left us in 1986 when she went to England on sabbatical. Below, Anne files her first report since returning.

Logo in Australia

by Anne McDougall

The Fifth International Logo and Mathematics Education Conference (LME5) was held at Lake Tinaroo, near Cairns in Queensland, Australia, in April of this year. It was a residential conference, including participants from the United States, Canada, the United Kingdom, New Zealand, and Australia.

LME5 was structured as a working conference, comprising a mix of fairly informal paper presentations with discussion and task-group activities. Topics tackled in the paper sessions included aspects of teaching and learning about recursion, the development and use of Logo tools and microworlds for mathematics, classroom environments and collaborative learning, and questions concerning curriculum in mathematics.

Group discussions focused on four themes: what we mean by "pedagogy" in Logo contexts, relationships between Logo programming activity and doing mathematics, the implications of predominantly psy-



chological perspectives in Logo and mathematics education research, and the extent to which there has been real change in mathematics education in the 10-odd years since the publication of Papert's *Mindstorms*.

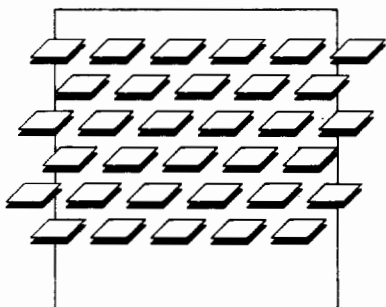
Task groups worked on delineating important issues and directions for teacher professional development, preparing specifications for a probability microworld exploring queueing theory with turtle "shoppers" in a screen-based supermarket, and writing programs to explore number patterns using a variety of representations—points around a circle on the screen, heights on a bar graph, and musical notes—concurrently.

The conference center at Lake Tinaroo provided an idyllic setting for our work, with beautiful local birds and butterflies, the lake, and nearby rain forest. Some of us visitors to northern Queensland found the local insect life rather disconcerting and were not altogether relieved by the locals diverting our attention with warnings about closing doors to keep out the poisonous snakes found in the area. But probably the most dangerous thing we did was to participate in a rather chaotic but most enjoyable cricket match, Australia versus the Rest of the World (sort of), after the conference closing session.

Dennis Harper is currently an associate professor at the University of the Virgin Islands. He was previously Educational Computing Coordinator at the Institute of Education in Singapore. He has been a lecturer at the National University of Malaysia and the University of Helsinki, Finland. In addition, he has been a teacher of mathematics, science, and computer science in Spain, Liberia, West Germany, Australia, and the United States. He holds a Ph.D. in International Education from the University of California, Santa Barbara. He is the author of *Logo Theory and Practice* and *Run: Computer Education*.

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

Learning With Computers: Effective Teaching Strategies



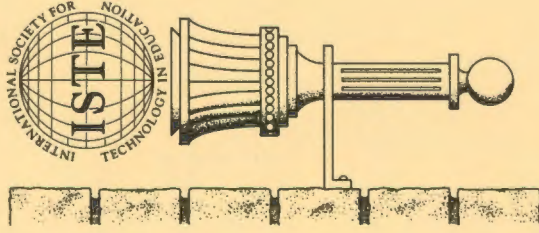
by Ken Ryba and Bill Anderson

Learning With Computers: Effective Teaching Strategies is filled with suggestions to help students become independent thinkers and productive members of learning teams. Excellent for university classes and as a resource for K-12 teachers. Drawing on cognitive theory and research, Ken Ryba and Bill Anderson outline specific methods that teachers can use with a variety of computer applications (Logo, word processing, adventure games, databases, spreadsheets, telecommunications) to enhance social and cognitive development. Includes strategies for cooperative learning.

Learning With Computers: Effective Teaching Strategies is \$20.00 plus \$4.50 shipping.

To order contact:
ISTE, 1787 Agate St., Eugene, OR 97403-1923; ph. 503/346-4414.

The *International Society for Technology in Education* touches all corners of the world. As the largest international non-profit professional organization serving computer using educators, we are dedicated to the improvement of education through the use and integration of technology.



Drawing from the resources of committed professionals worldwide, ISTE provides information that is always up-to-date, compelling, and relevant to your educational responsibilities. Periodicals, books and courseware, *Special Interest Groups*, *Independent Study* courses, professional committees, and the Private Sector Council all strive to help enhance the quality of information you receive.

It's a big world, but with the joint efforts of educators like yourself, ISTE brings it closer. Be a part of the international sharing of educational ideas and technology. Join ISTE.

**Join today, and discover how ISTE puts
you in touch with the world.**

ISTE

1787 Agate St., Eugene, OR 97403-1923.
ph. 503/346-4414.

Logo Exchange

ISTE

1787 Agate Street

Eugene, OR 97403-1923