

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



LOGO EXCHANGE

Winter 1991

Volume 10 Number 2

A Blizzard of Logo Ideas



Also in this issue:

Pattern Microworlds & Microworld Patterns

Logo in Latin America, Israel, & Japan

Beyond Piaget...

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,

Director of Advertising Services

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 2

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

Winter 1991

Contents

From the Editor—Logo First...and then HyperCard	Sharon Yoder 2
Quarterly Quantum—Do You Know a Logo Person?	Tom Lough 3
Fourth Irish National Logo Contest 1990	Michael D. Moynihan 4
Beginner's Corner—A Blizzard of Logo Ideas	Dorothy Fitch 8
Logo LinX—Pattern Microworlds and Microworld Patterns	Judi Harris 13
Logo Ideas—Snow Thoughts	Eadie Adamson 16
Windows on Logo—Pictograms	Glen L. Bull and Gina L. Bull 20
Math Worlds—...beyond Piaget: Logo learning under a new light...	A. J. (Sandy) Dawson 23
Sanity and the Single Computer (Part 1)	Judi Menken 26
Extra for Experts—The John Wayne Theory of Learning Lists	Mark Horney 30
Logo: Search and Research—Higher-Level Math Thinking: Part I	Douglas H. Clements 33
Starting With Scribbles	Eleonora Badilla-Saxe 37
Global Logo Comments—Reports From Costa Rica, Venezuela, Brazil, Japan and Israel	Dennis Harper 43

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.

Logo First...and then HyperCard

by Sharon Yoder

Another term is over, and with it another Logo class. Although I have taught Logo classes for nearly ten years now, this class—as always—had refreshing and delightful moments. This small but lively group was distinguished by the willingness of many class members to ask all sorts of questions. These delightful students were glad to share their confusion, their questions, and their discoveries. It seemed we never finished *my* plans for the day, but instead went off on wonderful tangents that resulted in a great deal of learning for us all. We were a real Logo community by the end of the term.

I have taught Logo so often that I recognize a fairly typical pattern in student learning. Initially, there is the thrill of making those first designs with the turtle. All of the easy-to-use features of the new Logos extend this excitement far beyond creating a square. Labeling drawings, stamping shapes, filling areas—all of these contribute to the initial thrill of success.

The next stage is learning to write procedures. That comes quickly and easily to most students. It's a natural move from writing code in the immediate mode or in the Command Center to adding to and ending in an editor. Successful procedure writing gives novices the excitement of writing *real* programs and lets those with programming experience make connections with other programming languages.

At that point in my course, I begin to emphasize structure and style. We talk about modularity. We discuss program design. We draw procedure trees. We develop guidelines for the final, published product students will turn in. We discuss the technicalities of Logo grammar. While I consider program design to be quite important, I am actually giving my students time to digest the powerful new ideas they have experienced in a short time span.

The next step is to begin dealing with more traditional programming ideas. We create loops using tail recursion. We learn about conditionals to stop recursive procedures. We learn about interactive reporters. We talk about procedure inputs...

...and about that point in time, my students' eyes begin to glaze over. Each new idea seems clear when explained by itself, but there comes a moment when most students are overwhelmed. While I intersperse these new programming ideas with such "fun" topics as

animation, multiple turtles, and music, I continue to move forward with new programming ideas—passing parameters, global variables, incrementing, using OUTPUT, predicates, list processing.

That overwhelming moment of feeling lost is something I have come to expect. It often happens at about the same time to the whole class. I constantly reassure my students that it will all come together for them. I emphasize which ideas are really important and which ones they can take their time digesting. I model new concepts over and over and recommend that they use such models to build the programs for their assignments.

While not everyone gets to the end of the class grasping all of these important programming ideas, I am confident that all of the students have experienced most of the ideas at a level that will allow them to come back to later to add depth to their understanding.

This most recent class was no different than most, except that it included three teachers who were also taking a Hypermedia course in which they were learning *HyperCard* and the associated programming language, HyperTalk. One of the teachers was a complete novice to programming. A second had worked with Logo in her classroom. The third was a self-taught BASIC programmer who taught high school computer science classes. These three were perhaps the most vocal in the class. They brought a myriad of classroom and personal experiences to their Logo learning. In addition, they shared the experience of taking two classes together, and thus made unexpected connections between the two classes. I often found myself answering HyperTalk questions before or after class—and the Hypermedia instructor often found himself answering Logo questions! These particular teachers were quite vocal when they reached the "overwhelmed" stage—in part because they were overwhelmed in Hypermedia as well. Their willingness to share made it easier for me to address the problem directly for the entire class.

Toward the end of the class, these three teachers came to me with a suggestion. They said that they thought that the Logo class should be required before the Hypermedia class. They felt certain that they would have experienced much less frustration in learning to program in HyperTalk if they had a solid grounding in

Logo first. Only toward the end of the two classes did they feel that they had the knowledge level that allowed them to make really useful connections.

In fact, the Logo class is required before the Hypermedia class in our Master's program. Somehow the course announcement these students had received omitted the information about prerequisites, so they unknowingly took both classes. I was certainly sorry that they had experienced so much frustration, but I was delighted by their observations. They had clearly confirmed my long-held belief that Logo is a *great* first programming language for anyone.

So before your school throws out Logo in favor of *HyperCard* (or its equivalent), think carefully. Logo is still the best way I know to learn important programming ideas, no matter what the hypermedia enthusiasts say. Just ask my students!

Sharon Yoder
ISTE
1787 Agate Street
Eugene, OR 97403

Do You Know a Logo Person?

by Tom Lough

Welcome to the QQ. For new readers, this column (which may never be sufficiently brief!) offers a quantum (a fixed quantity or amount) of information, news, commentary, or what have you each quarter.

Recently, I was struggling up a confusing maze of ramps with a heavy load of instructional materials at the entrance of a university building. Another person, wanting to be helpful, came up from behind to offer assistance just about the time I discovered the right path. He exclaimed, "Ah, you found it. You must be a Logo person."

I've been called lots of things during my life (some of them not very pretty!). But it was a satisfying feeling to be called a "Logo person." I liked it; however, I must admit that I felt a little uncertain. After all, I have never seen a list of the qualifications one must meet to be a "Logo person."

What does it take to be a "Logo person," or to be perceived as one? What does a "Logo person" do? How does a "Logo person" think and act? I started to write down some possibilities, but ran into difficulty. I quickly realized that I needed some big-time help on this one.

How about it, LX readers? You are the ultimate resource for a project such as this. Would you take a moment to describe what you feel are the qualities of a "Logo person" and mail them to me at the address below? Or, better yet, send me the name and a description of someone you know personally who exemplifies the qualities of a "Logo person." I'll select and publish several descriptions in future columns.

Let's agree to a couple of ground rules. A "Logo person" does not have to possess a lifetime of Logo experience. Nor should he or she necessarily be a famous person. Moreover, we should not do an exercise of this sort in an exclusionary spirit. The purpose is

not to set up an elite group of people but, rather, to identify some characteristics and qualities to which we all can aspire. Most of all, I would prefer descriptions of living, breathing people who possess some of these characteristics and qualities. Naturally, I would be delighted to receive descriptions and stories of any of your students who became "Logo people."

I look

FORWARD 100

to hearing from you. Many thanks!

Tom Lough
Founding Editor
PO Box 394
Simsbury, CT 06070

PS: A sure way to beat the winter blahs is to play with the REPEAT command. Have you ever tried putting REPEATs inside REPEATs? If not, you are in for a treat! Use the example below to get started. Can you predict what will happen? You take it from here!

```
REPEAT 4 [REPEAT 4 [FORWARD 25 LEFT
90] FORWARD 50 RIGHT 90]
```



Fourth Irish National Logo Contest 1990

by Michael D. Moynihan

Introduction

The fourth Irish National Logo Competition was held on June 16, 1990, at St. Patrick's College of Education in Dublin. St. Patrick's is one of the training colleges for Irish primary (elementary) teachers. From a small beginning, the competition has grown in stature and prestige over the years, and for the first time sponsorship was available.

Cara Computers agreed to sponsor the competition and to award a COMPAQ Portable Computer (LTE Model 20) to the overall winner, the student from the Elementary or Junior Division who, in the opinion of the judges, presented the best solutions. Fifty-four children took part, and each of them came home with at least a certificate.

Sponsorship enabled the organizers to circulate information to all Irish primary schools. Previously it was only the Logo Centers that were involved. Over the years there has been a growth of interest in Logo in Irish primary schools. There is now an increased awareness of the potential of Logo in the primary school, and teachers are gaining in experience, even though computers are still scarce and classes are large.

Students nominated by schools or through the Logo Centers were screened and the best were invited to take part in the National Logo competition. Because not all students had worked in traditional list processing, there was a special Graphics division. The main divisions, however, were the Elementary Division (for students 12 years of age or under) and the Junior Division (for students 15 years or under.)

In each division five problems had to be solved within a two-hour period. I would like to share the Elementary Logo problems and the winning solutions with the readers of *LX*.

A Brief History of the Competition

In 1985 Fred Klotz and Sean Close, mathematics lecturers at St. Patrick's College, had the idea that Logo could be used to develop and improve mathematical thinking and concepts, especially among mathematically able students attending primary (elementary) schools.

Because many Irish teachers work with large classes and because most primary schools in Ireland are lucky to have even one microcomputer, it was not feasible at that time to implement their ideas satisfactorily within

the normal Irish primary classroom. In order to overcome this problem, contact was made with experienced teachers who had access to a number of microcomputers and who were willing to set up a center in their local area. For many of these teachers, Logo was new and presented them with yet another challenge. Contact was made with local primary schools to seek their help in identifying mathematically able students 9 to 12 years of age who would be suitable for such a Logo course. There was an enthusiastic response from the schools. The students underwent an assessment, and those who fared best were invited to participate in Logo courses at a local center.

At present there are eight such centers in Ireland. While the primary aim of the courses is to develop mathematical concepts through the use of Logo, some of the centers have also entered students for the Logo Divisions of the ICPSC. The students enjoy the challenge of problem solving and competing internationally. Indeed, Irish students have been fortunate enough to achieve first place in the ICPSC rankings in a Logo Division each year since 1987.

Fred Klotz died in April, 1988, but his many friends who read *LX* will be pleased to know that the Dr. Fred Klotz Memorial Trophy is presented to the winner of the Elementary Division of the Irish National Logo Competition.

The Winner

The 1990 overall winner was Gavin Hurley, age 12, was ranked first in the Elementary Division and thus retained the Dr. Fred Klotz Memorial Trophy he had previously been awarded in 1989. As sponsor of the event, Cara Computers also presented Gavin with a COMPAQ portable computer which sells in Ireland for more than 3,500 pounds!

Gavin was a student at Scoil an Spioraid Naoimh, Bishopstown, Cork, until June of 1990. He has been one of the outstanding students at the Cork center for more than two years. He was in the United States with his family in 1990. His father, on leave from University College, Cork, is teaching mathematics at the University of Michigan in Ann Arbor. In 1991 Gavin will return to Ireland to resume his secondary education.

It is my hope that the problems and their suggested solutions are of interest to readers of *LX*. Remember that the solutions were written by a 12 year old during

a two-hour period under strict examination conditions and supervision. The solutions are written in Terrapin Logo for the Apple II computer.

Problems: Elementary Division

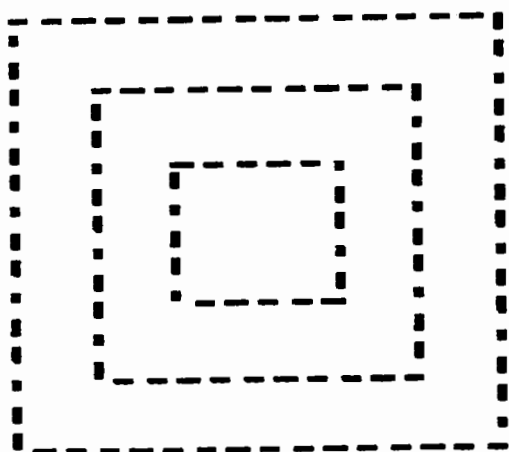
Problem 1: Spectacles

Write a program that will produce the following design:



Problem 2: Dashed Squares

Write a program that draws the following:



Problem 3: Toss a Die

An experiment consists of the following:

1. Toss a standard die and note the score.
2. If the score is 6, toss the die again and add the two scores together.

Write a program to simulate this experiment a given number of times. The output should look like the following when the experiment is repeated three times.

EXPERIMENT 3

```
score = 2
score = 11
score = 4
```

You may use the procedure PICK, which picks an item at random from :LIST

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

Problem 4: Count Vowels

Write a program that requests a word as input, and then outputs the number of vowels in that word.

Sample run:

```
TYPE IN A WORD SCREEN THE WORD
SCREEN CONTAINS 2 VOWEL(S)
```

Test your program with the words SCREEN and COMPUTER

Problem 5: Palindromes

A palindrome is a word that reads the same backwards as forwards, for example: LEVEL, DID, NAVAN. Write a program that takes a list of words as input, identifies those words that are palindromes, and prints them out.

For example,

```
PALINDROMES [SCREEN EYE COMPUTER
SEES]
```

should print out

```
EYE SEES
```

Gavin's Solutions

Gavin produced short, economical procedures with several explanations put in as comments. As an aid to the user, for example, for Problems 1 and 2, he suggests the sizes that are best suited for screen layout.

Solution to Problem 1

```
TO SPECTACLES
  ; GAVIN HURLEY
  GET.RADIUS
  DRAW
  HT
  SET.UP :RADIUS
```

```

DRAW.SPECTACLES :RADIUS
END

TO GET.RADIUS
; GAVIN HURLEY
CLEARTEXT
PR [PLEASE INPUT A RADIUS - TRY 50]
; 50 because it fits best
PR []
MAKE "RADIUS FIRST RQ
PR []
END

TO SET.UP :RADIUS
; GAVIN HURLEY
; gets to starting position which is
; where the bridge meets the left
; frame
PU
LT 90
FD :RADIUS / 2
RT 90
PD
END

TO DRAW.SPECTACLES :RADIUS
; GAVIN HURLEY
LCIRCLE :RADIUS
ARCR :RADIUS / 2 180
LCIRCLE :RADIUS
END

TO LCIRCLE :RADIUS
; GAVIN HURLEY
REPEAT 360 [FD :RADIUS * 3.14159 /
180 LT 1]
END

TO ARCR :RADIUS :STEPS
; GAVIN HURLEY
REPEAT :STEPS [FD :RADIUS * 3.14159
/ 180 RT 1]
END

```

Solution to Problem 2

```

TO DASHED.SQUARES
; GAVIN HURLEY
DRAW
HT
GET.SIZE
SET.UP
DRAW.DASHED.SQUARES :SIZE 15
END

```

```

TO GET.SIZE
CLEARTEXT
PR [PLEASE INPUT A SIZE - 150 FITS
BEST]
PR []
MAKE "BIG.SIZE FIRST RQ
MAKE "SIZE :BIG.SIZE / 30
; []
; divided by 30 because there are 30
; dashes and spaces in the first
; line
; []
END

```

```

TO SET.UP
; GAVIN HURLEY
PU
LT 135
; gets to starting position
FD :SIZE * 15 * SQRT 2
; []
; multiplied by 15 because half I
; only
; want half the diagonal.
; and SQRT 2 to find the diagonal.
; []
RT 135
PD
END

```

```

TO DRAW.DASHED.SQUARES :SIZE :NUM
; GAVIN HURLEY
IF :NUM < 5 STOP
SQ.DASH.DOT :SIZE :NUM
MOVE :SIZE
DRAW.DASHED.SQUARES :SIZE :NUM - 5
END

```

```

TO DASH.DOT :SIZE :NUM
; GAVIN HURLEY
REPEAT :NUM [FD :SIZE PU FD :SIZE
PD]
END

```

```

TO MOVE :SIZE
; GAVIN HURLEY
PU
FD :SIZE * 5 RT 90
FD :SIZE * 5 LT 90
PD
END

```

```

TO SQ.DASH.DOT :SIZE :NUM
; GAVIN HURLEY

```




```
REPEAT 4 [DASH.DOT :SIZE :NUM RT 90]
END
```

Solution to Problem 3

```
TO EXPERIMENT :NUM
; GAVIN HURLEY
TEXTSCREEN
CLEARTEXT
PR []
REPEAT :NUM [ROLL.IT]
END
```

```
TO RESULTS :NUMBER
; GAVIN HURLEY
PR SE [score =] :NUMBER
PR []
END
```

```
TO ROLL
; GAVIN HURLEY
OP 1 + RANDOM 6
; []
; I used 1 + RANDOM 6 instead of
; PICK because it is shorter
; []
END
```

```
TO ROLL.IT
; GAVIN HURLEY
MAKE "NUMBER ROLL
IF :NUMBER = 6 MAKE "NUMBER :NUMBER
+ ROLL
RESULTS :NUMBER
END
```

Solution to Problem 4

```
TO COUNT.VOWELS
; GAVIN HURLEY
MAKE "COUNTER 0
TEXTSCREEN CLEARTEXT
GET.WORD
CHECK :WORD
RESULTS :WORD :COUNTER
END
```

```
TO GET.WORD
; GAVIN HURLEY
PR [TYPE IN A WORD, PLEASE]
PR []
MAKE "WORD FIRST RQ
PR []
END
```

```
TO CHECK :WORD
; GAVIN HURLEY
```

```
IF EMPTY? :WORD STOP
TEST VOWEL? FIRST :WORD
IFT MAKE "COUNTER :COUNTER + 1
CHECK BF :WORD
END
```

```
TO RESULTS :WORD :COUNTER
; GAVIN HURLEY
PR ( SE [THE WORD] :WORD [CONTAINS]
:COUNTER [VOWEL(S)] )
PR []
END
```

```
TO VOWEL? :LETTER
; GAVIN HURLEY
OP MEMBER? :LETTER [A E I O U]
END
```

Solution to Problem 5

```
TO PALINDROMES :LIST
; GAVIN HURLEY
TEXTSCREEN
CLEARTEXT
CHECK.IT :LIST
END
```

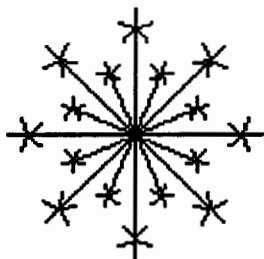
```
TO CHECK.PALINDROME :WORD
; GAVIN HURLEY
IF COUNT :WORD < 2 MAKE "IT "TRUE
STOP
; less than 2 because one could have
; 5 letter palindrome
IF FIRST :WORD = LAST :WORD
CHECK.PALINDROME BF BL :WORD ELSE
MAKE"
IT [FALSE]
END
```

```
TO CHECK.IT :LIST
IF EMPTY? :LIST PR [] PR [] STOP
CHECK.PALINDROME FIRST :LIST
IF :IT = "TRUE PRINT1 SE FIRST :LIST
" "
CHECK.IT BF :LIST
END
```

Michael D. Moynihan
Colaiste an Spioraid Naoimh
Bishopstown, Cork
Ireland

email ZZCE9001@IRUCC.VAX.IE





A Blizzard of Logo Ideas

by Dorothy Fitch

Having lived in New England all my life, I find it difficult to think of a winter without snow! (Imagine teaching where there is *no* chance for an instant holiday when school is called off because of snow!) So, this month we'll explore some ideas to try when the snow begins to fall. For those of you in warmer climes, maybe these activities will cool you off or help your students imagine what a snowy day is like.

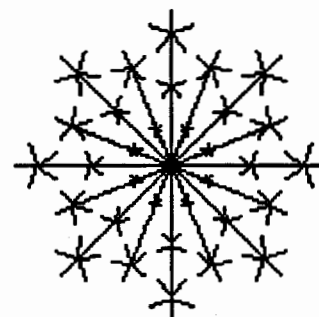
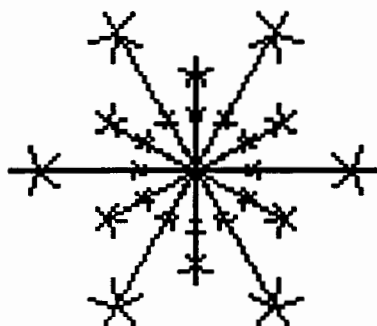
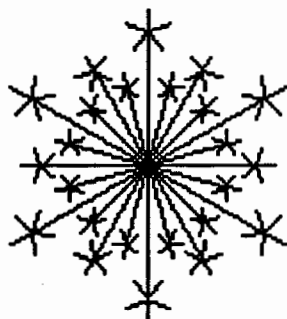
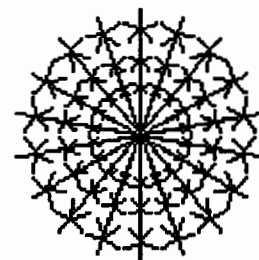
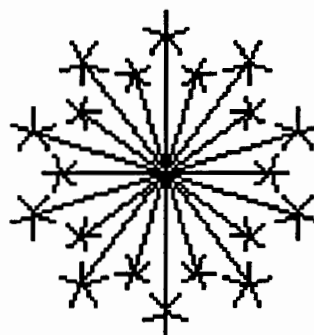
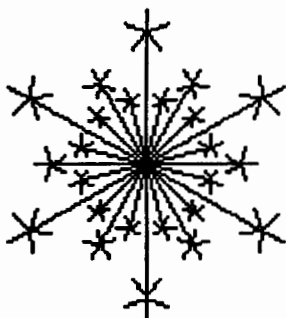
Logo Snowflake Procedures

For starters, here are few Logo snowflakes. Can you see any similarities among them? Is there an overall pattern that applies to them all? It may be difficult to see a pattern at first, but you'll soon see that the procedures are not as complicated as they might appear.

If you have used Logo for any length of time, you have probably come across the Total Turtle Trip Theorem, whether you knew it by this name or not. This theorem states:

"When the turtle travels around a shape and returns to where it started, heading in the same direction, it will have turned a total of 360° (or multiple thereof)."

The Total Turtle Trip Theorem helps us figure out how much to turn the turtle to draw a square ($360^\circ / 4$ sides = 90° per side), a hexagon ($360^\circ / 6$ sides = 60° per side), or any other regular shape. A general procedure for any shape is therefore:



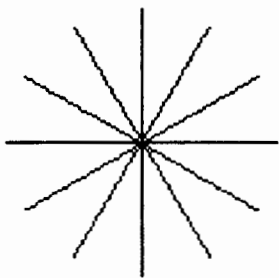
```
TO SHAPE :SIZE :NUMBER.OF.SIDES
REPEAT :NUMBER.OF.SIDES [FORWARD
:SIZE RIGHT 360/:NUMBER.OF.SIDES]
END
```

If you type

```
SHAPE 50 4
```

you will get a medium-sized square. If you type SHAPE 100 8, you will get a large octagon. If you type SHAPE 20 3, you will get a small triangle. Logo uses the first input to SHAPE for the length of each side, and calculates 360 divided by the second input (the number of sides) to determine the angle to turn.

The Total Turtle Trip Theorem can also help us draw shapes and designs like this:




The instruction to draw this design is:

```
REPEAT 12 [FORWARD 50 BACK 50
RIGHT 360/12]
```

This pattern—going forward some amount, back the same amount, and then turning—is the foundation for our snowflake. Each “arm” of our snowflake will have a similar five-pointed star at its end to give it a “flakier” appearance. The size of this star is relative to the size of the snowflake. A small snowflake will have smaller points than a large snowflake.

Holy Applesauce, Batman! What's that mean?

Last issue, in this column, the open-Apple and closed-Apple symbols in Dorothy's article didn't come through in our typesetting...the result was a rather incomprehensible title—"A is for Apple! Or Control-A, or Maybe Even r-A" which should have read "A is for Apple! Or Control-A, or Maybe Even -A." Our program read the open Apple symbol as an "r" there and in the second paragraph on page 13.

Thank you to all our readers and Dorothy for helping us find this error; our apologies for any confusion this may have caused.

Ian Byington, production assistant

So, we can create this pair of procedures to draw the beginnings of a snowflake:

```
TO SF :NUMBER :LENGTH
REPEAT :NUMBER [FORWARD :LENGTH
POINT :LENGTH/6 BACK :LENGTH
RIGHT 360/:NUMBER]
END

TO POINT :SIZE
REPEAT 5 [FORWARD :SIZE BACK :SIZE
RIGHT 72]
END
```

A simple snowflake instruction would be:

```
SF 6 50
```

SF is short for SNOWFLAKE.

By combining SF and turn commands, you can generate a wide variety of snowflakes. (Each real snowflake is different, right?) Purists will tell you that a snowflake crystal is hexagonal in shape, but if you want to make an octagonal or pentagonal snowflake, be my guest!

At the end of this article are some snowflakes and the commands that created them. I'm sure you'll want to try more of your own!

With students who are already familiar with Logo and are learning about procedure inputs, you can walk them through the development of these procedures. Younger students can design amazing snowflakes just by using the SF procedure and turn commands. You might even want to add this procedure, which allows students to use fractions to turn the turtle the right amount:

```
TO TURN :AMOUNT
RIGHT 360 * :AMOUNT
END
```

Now students can type

```
TURN 1/12
```

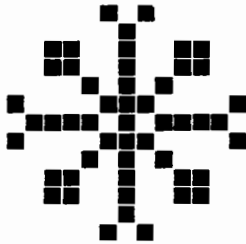
to turn the turtle one-twelfth of the way around the circle. They can use it to say, in effect, "I want to turn the turtle so that I can make 12 arms." Then, the instructions for FLAKE1, for example, could look like this:

```
TO FLAKE1
SF 8 50
SF 8 30
RIGHT 360/16
SF 8 40
SF 8 15
END
```



Logo Snowflake Shapes

Another way to play with snowflakes in Logo is to design a new turtle shape that looks like a snowflake, if your version of Logo provides a shape editor. Here is one possibility:



You can move this flake as you would the turtle. In Logo PLUS/Apple, the flake will turn as the turtle would. In other versions of Logo, the flake will always keep the same orientation.

You can even stamp this shape instead of the five-pointed star at the ends of your snowflakes. Rather than using the SF (for Snowflake) and POINT procedures above, instead use this SSF (for Stamped Snow Flake). SNOWFLAKE 9 and SNOWFLAKE 10 on the snowflake page (page 12) provide two examples.

```
TO SSF :NUMBER :LENGTH
  REPEAT :NUMBER [FORWARD :LENGTH
    STAMP BACK :LENGTH RIGHT 360/
    :NUMBER]
  END
```

If you are using Terrapin Logo for the Macintosh, replace STAMP with STAMP SHAPE in the above instruction.

An Animated Snowfall

We can use this snowflake shape to create a little snow flurry. Here are three procedures that cause snowflakes to drift down the screen one at a time and accumulate at the bottom.

Terrapin's Logo PLUS/Apple version:

```
TO SNOW
  DRAW
  PENUP
  SETXY (100 - RANDOM 200) 100
  SETSHAPE 1
  FALL
  END

TO FALL
  IF ANYOF YCOR < -95 COLORUNDER = 1
```

```
    THEN LAND
  SETHEADING 180 + (60 - RANDOM 120)
  FORWARD 10 + RANDOM 10
  FALL
  END
```

```
TO LAND
  PENDOWN
  STAMP
  PENUP
  SETXY (100 - RANDOM 200) 100
  SHOWTURTLE
  END
```

For LogoWriter, make these changes:

```
Change: DRAW
to: CG
Change: SETXY (100 - RANDOM 200)
      100
to: SETPOS LIST (100 - RANDOM
      200) 100
Change: SETSHAPE 1
to: SETSH 1
Change: IF ANYOF... line
to: IF OR YCOR < -95 COLORUNDER
      = 1 [LAND]
```

Type SNOW to start. The SNOW procedure sets up the screen, changes the turtle to the flake shape and places it at a random position at the top of the screen. Then it calls the FALL procedure, which uses RANDOM with SETHEADING and FORWARD to simulate a drifting motion for the snowflake as if falls to the ground. The FALL procedure checks to see if the flake has either reached the bottom of the screen or has touched another flake. If either is true, then Logo runs the LAND procedure, which stamps the snowflake shape on the screen and sets the next flake in action. After a while, the snow will begin to pile up!

Multiple-Turtle Snowflake Fun

Using Terrapin Logo for the Macintosh (or other versions of Logo with multiple turtles), you can have more than one snowflake falling from the sky at a time. Here's a version with six snowflakes in motion:

```
TO SNOW
  TELL [0 1 2 3 4 5]
  CLEARGRAPHICS
  SETSHAPE :FLAKE
  SHOWTURTLE
  PENUP
  EACH [SETXY (100 - RANDOM 200) 60]
  MAKE "BOTTOM -65
```

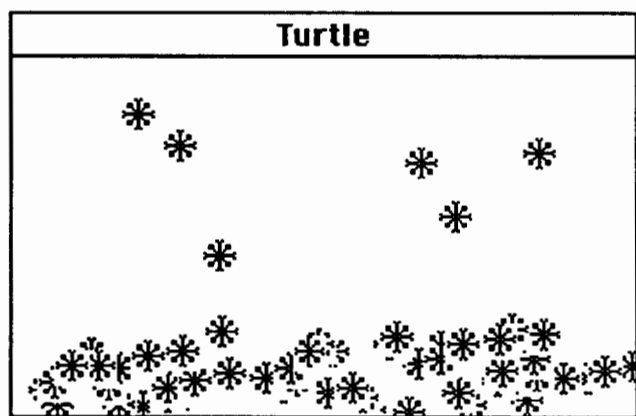


```
FALL
END
```

```
TO FALL
EACH [IF YCOR < :HEIGHT [ LAND]]
EACH [SETHEADING 180 + ( 60 - RANDOM
    120 )]
EACH [FORWARD 10 + RANDOM 10]
FALL
END
```

```
TO LAND
PENDOWN
STAMP SHAPE
PENUP
SETXY (100 - RANDOM 200) 60
SHOWTURTLE
MAKE "BOTTOM :BOTTOM + .5
END
```

These procedures are similar to those for a single turtle, but here each turtle makes its own random way to the ground. This version increases the value of the variable *BOTTOM* so that the flakes pile up without a check for touching. Here is a Logo screen after a few minutes:



Enhance your scene by changing the background to blue, adding a snowman, or having a snowplow appear after the storm is over! Just use your imagination.

Happy Logo adventures! See you in the spring.

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 workshops 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

Terrapin, Inc.

announces

Logo Math: Tools and Games

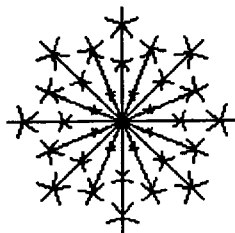
This new version of *Logo Math* is a comprehensive set of secondary math programs for use with PC Logo. *Logo Math* features GEOMETER, a complete geometric construction set; GRAPHER, an algebraic function grapher; and six additional tools for working with trigonometry, statistics, probability, and fractions. Writing in the Logo language, *Logo Math's* tools encourage exploration of their specialized domains. At any point, students may use any Logo command to help explore a math problem.

Logo Math also has six games through which students learn signed and complex numbers, Cartesian and polar coordinates, symmetry, and functions. For more information, contact

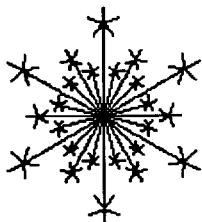
Terrapin Software
400 Riverside Street
Portland, ME 04103



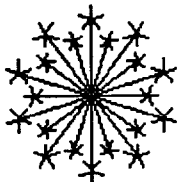
Snowflakes



TO FLAKE1
SF 8 50
SF 8 30
RIGHT 360 / 16
SF 8 40
SF 8 15
END



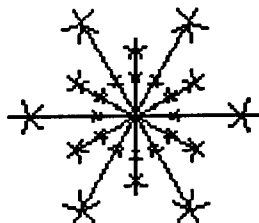
TO FLAKE2
SF 6 50
RIGHT 360 / 12
SF 6 35
RIGHT 360 / 24
SF 12 25
END



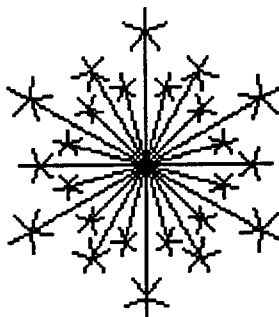
TO FLAKE3
SF 10 40
RIGHT 360 / 20
SF 10 30
END



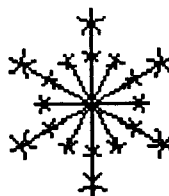
TO FLAKE4
SF 18 40
SF 18 28
SF 18 18
END



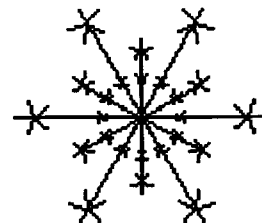
TO FLAKE5
SF 8 40
RIGHT 360 / 16
SF 8 25
END



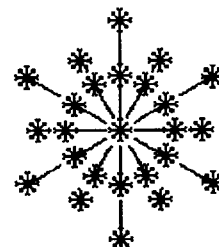
TO FLAKE6
SF 6 50
RIGHT 360 / 24
SF 12 30
RIGHT 360 / 24
SF 6 40
END



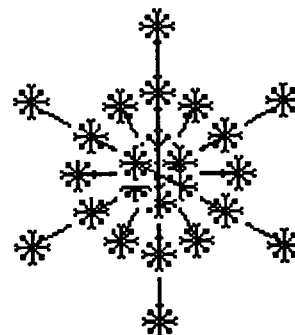
TO FLAKE7
SF 6 30
SF 12 18
END



TO FLAKE8
RIGHT 360 / 12
SF 6 40
RIGHT 360 / 12
SF 6 25
RIGHT 360 / 12
SF 12 15
END



TO FLAKE9
SF 6 60
RIGHT 360 / 12
SF 6 45
RIGHT 360 / 12
SF 12 30
END



TO FLAKE10
SF 6 55
SF 12 30
SF 6 10
END

Pattern Microworlds and Microworld Patterns

by Judi Harris

A mathematician, like a painter or poet, is a maker of patterns.

—G.H. Hardy

Young children are natural pattern-makers. Their fascination and facility with rap music, jump rope rhymes, percussive rhythms, and foreign language, for example, attest to innate interests in and abilities to detect, understand, and reproduce patterns in different contexts. If children are natural pattern-makers, then according to G.H. Hardy (and Seymour Papert, the "father of the Logo turtle,") they are also natural mathematicians.

At its most essential level, mathematics is the study of patterns. Curricular reforms of the past 20-30 years have shown us that appropriate mathematical experiences for young children include environments in which they can experiment with pattern creation. Consequently, blocks of all types (such as pattern blocks, Cuisenaire rods, and attribute blocks) have established permanent homes in many primary classrooms.

The study of patterns necessitates, to varying degrees, facility with abstraction. A first step in developing this metacognitive ability is to make representations of patterns created with manipulative objects in other media, such as drawing a Cuisenaire rod construction with crayons on paper. The fine-motor limitations of many young children may impede this process or make it distasteful to them. Use of a simple, extensible Logo microworld can ease these fine-motor restrictions while providing a naturally motivating environment for semi-abstract pattern exploration.

Building Blocks

LogoWriter's shape-defining features make it simple to create a set of blocks that can be moved around the screen and positioned according to users' preferences. These 10 pieces, modeled on intelligence test blocks, were defined as turtle shapes 1 through 10. The open square is the temporary shape for the turtle before it dons the coats of N, O, Q, R, V, W, X, Y, or Z.



Telling the turtle to change its shape (SETSH *shape number*) so that it looks like a block in the set is a simple matter. Procedures titled with block letter names function quite straightforwardly as top-level tools.

```
TO N
SETSH 1
END
```

```
TO O
SETSH 2
END
```

```
TO Q
SETSH 3
END
```

```
TO R
SETSH 4
END
```

```
TO V
SETSH 5
END
```

```
TO W
SETSH 6
END
```

```
TO X
SETSH 7
END
```

```
TO Y
SETSH 8
END
```

```
TO Z
SETSH 9
END
```

Whenever the user types the letter of the block shape that s/he wants to move and presses the Return key, the turtle obligingly assumes the selected block shape.



Typing Q gives:



Block Locomotion

Since the block is really the turtle, it can be moved to the desired location with standard turtle graphics commands (i.e., `PU FORWARD 20 LEFT 90 FORWARD 70 PD`) or *LogoWriter's* turtle move keys.



An image of the block can be left in any location with a "place tool," which makes use of the `STAMP` command:

```
TO P
PD
STAMP
END
```

When a block is at a location in which the user would like it to stay, s/he simply types P and presses the Return key. The turtle can then take on a different block shape so that the user can continue creating the pattern.

Block Impressions

Many different types of patterns can be created with tools such as these.



NRNRNR...



OVQOVQOVQ...



ZVVY VVVY ZWY ZVVY ZVVV ZVV

As you can see, patterns at very different levels of complexity can be represented graphically. The block microworld can also be used to create pictures of recognizable forms, such as this one:



Completed patterns and pictures can be saved and recalled with prompted `SAVE` and `SEE` tools.

```
TO SAVE
CC
TYPE [NAME?]
SAVEPIC WORD "" FIRST READLISTCC
END
```

```
TO SEE
CC
TYPE
TYPE [PICTURE?]
LOADPIC WORD "" FIRST READLISTCC
END
```



Block Assistance

Note that the screen always displays a set of labelled blocks for the user's reference. In this microworld, they are stored in the picture file BLOCKS, called in the procedure, BEGIN, which can be invoked each time a new pattern is begun.

```
TO STARTUP
CC
CG
HT
LOADPIC "BLOCKS
SETSH 10
PU
HOME
PD
ST
END
```

A HELP screen for teachers and older children is displayed each time the BLOCK.DRAW page is selected from the Contents Page.

```
-To begin, type BEGIN and press RETURN.
-To change the block's shape, type the
  letter above the shape you want and
  press RETURN.
-To move the block, hold down the APPLE
  key and press the 9, then use arrows.
-To place the block, press the key
  marked ESC, then type P & press RETURN.
-To clear the screen, type CG and press
  the key marked RETURN.
-To save a picture, type SAVE.
-To see a picture, type SEE.
-To see these instructions, type HELP.
N O Q R U W X Y Z
```



This image is also accessible with a HELP command:

```
-To begin, type B.
-To move blocks, use the arrow keys. To
  place a block, type P.
-To save a picture, type S.
-To see a picture, type C.
-To erase the screen, type E.
-To see these instructions, type H.
```

```
N O Q R U W X Y Z
```



```
TO HELP
CG
HT
LOADPIC "HELP
END
```

And, voila! An educational, extensible, amendable pattern playground in just 14 simple *LogoWriter* procedures, with which young children can create, save, recall, print, and share block patterns. Mr. Hardy would be pleased.

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



Snow Thoughts

Eadie Adamson

Logo Ideas

It's winter in the northern hemisphere. Thoughts turn to snow... and to some old ideas. Too often, we discard "old stuff" because we've done it before.

Snowflakes call up memories of those wonderful cutouts we used to make from folded paper. There's interesting geometry in creating a paper snowflake that relates to the way we can build a Logo procedure to draw a snowflake. Think of the multiple folds as multiple branches of a snowflake. The angled cutouts radiate from the fold. One side reflects the other.

In addition to building on some ideas about geometry, creating Logo snowflakes is a nice way to teach about building from subprocedures and about using variables. Programming snowflakes can be adjusted for many levels of Logo expertise, with each case presenting opportunities for new discoveries.

First, Some Facts About Snowflakes

I decided to begin by sharing a few snow facts with my students. Since we meet in a computer lab, I did a little of the research myself. A classroom teacher would probably find it more suitable to assign the research as a project to share in the classroom. The question of the day could be, "How much can you find out about snowflakes by.....?" Scientific facts, pictures, even snow stories could be collected.

I remembered an article called "sNo Mystery" by Judi Harris in the January 1988 issue of *Logo Exchange*. I made a few copies of the following excerpt to share with my class:

There are seven types of snow crystals: six-rayed stars, hexagonal plates, columns, bullets, needles, and irregular forms. ... Snow crystals can range in diameter from diamond dust size to 3/16 inch.

Dendritic (stellar) snow crystals are formed with six branches, spaced at 60-degree angles, radiating out from the center point. Different branches can be combined to make representations of different snow crystals. (pp. 25-26)

I read Judi's description to my students, emphasizing that there were two facts they should listen for in the last two sentences. Asking my students to think about this with Logo, I read the description a second time. I

also found some photographs and drawings of snowflakes and shared these with the class. This preparation helped give them some facts as well as some visual ideas as starting points.

Making a Branch

Before the students began working on their own individual designs, we worked out a very simple example as a class so that there was a model to follow. The Logo rules were that the turtle must begin and end in the same position, creating a "state transparent" procedure. The advantage of this became obvious only later, when some procedures that were not state transparent resulted in some odd aberrations. The single branch we programmed together turned out something like this:

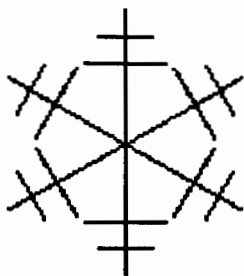
```
to branch
forward 30
right 90
forward 15 }      Draws a bar
back 30
forward 15
left 90
forward 10
right 90
forward 10 }      Draws
back 20           another
forward 10        bar
left 90
forward 10
back 50 <-1      Returns to the
end              starting point
```

After the procedure was written, we reread the description of the dendritic snowflakes. The students were challenged, remembering what had been read to them (the bold-faced sentences above), to generate a snowflake from the single branch. The 60-degree turn was either remembered, guessed at, or found by trial-and-error:

```
repeat 6 [branch right 60]
```

Everyone took some time playing with the sample, working out how to turn a single branch into a snowflake. Here's what it looked like:





The shape was not very interesting, but that was deliberate. The next step (Logo homework??) would be for each student to create a variation.

We elected to use graph paper to plan out a single "branch" of a snowflake. One square on the graph paper would equal 10 turtle steps. Away from the computer the students struggled with figuring out what the turtle should do. The chief requirement was that the turtle was to begin at the base of the branch and return to the base when the drawing was finished.

The Scale of a Snowflake

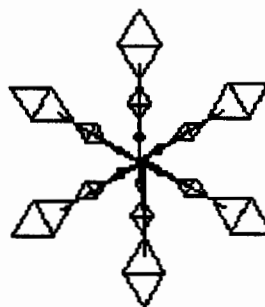
Some students explored further, using the idea of "scaling" the branches to create a variety of sizes of snowflakes. Think about what makes the line and what makes the shape. To scale down a shape, we need to change by a given factor the length of the line drawn. The turns are unaffected by the size and in fact are necessarily the same if you want to end up with the same shape you started with. It's the length of the lines drawn that needs to be changed. Here's an example of that idea, using the branch procedure above:

```
to branch :scale
forward 30 / :scale
right 90
forward 15 / :scale
back 30 / :scale
forward 15 / :scale
left 90
forward 10 / :scale
right 90
forward 10 / :scale
back 20 / :scale
forward 10 / :scale
left 90
forward 10 / :scale
back 50 / :scale
end
```

In the new branch procedure, a scale of 1 will give us our original branch. Other whole numbers will dimin-

ish the size—a nice way to think about fractions! Try both whole numbers and fractions as inputs for the scale. Experiments with decimals lead to more surprising results and can trigger an interesting discussion about numbers.

A scaled branch can be used to create interesting snowflakes without writing more procedures. Simply superimpose one size snowflake on top of another. Or make a variety of branches, all of which can be scaled, and combine them into unique flakes. Here is one student's example:



And Then There Was the Snowstorm

The idea of snowstorms grew from a few students happening to explore the *LogoWriter* activity card on creating snow with four turtles.

While they were working it began to snow. Even in New York, everyone stops and looks outside when it snows. Our lab is on the fifth floor with a tall building across the street. Wind occasionally blew down the street, then stopped, then started up again. We watched the patterns as the wind changed.

Thinking about what they observed, some students asked how they could program changes in wind direction. I always keep a diagram on the board that shows a number of headings for the turtle. Students can refer to this diagram when using the *seth* command in their programming. We renamed the headings for appropriate compass directions and wrote procedures for each heading. Here's an example:

```
to south
seth 180
end
```

Then we began experimenting:

```
tell all
pu
south
repeat 50 [forward 1]
west
repeat 25 [forward 1]
```



Students preferred "southeast" (seth 135) and "southwest" (seth 225) to other directions. These produced the most dramatic flow of form across the screen. There were fast storms and more leisurely ones. Students began watching what their companions were producing and sharing interesting headings and discussing what made the turtles go faster on some screens. Sometimes it was due simply to the difference between a GS at fast speed and an Apple IIe, but other differences, for example, the amount for forward distance, took a little more investigating.

One student explored the idea of programming wind gusts as well. What does a wind gust look like? How do the snowflakes blow? Next time there's snow, take an extra hard look at how the snowflakes move. Here's my student's version of a gust:

```
to gust
  seth 0
  forward 10
  seth 45
  forward 10
  seth 90
  forward 10
  seth 135
  forward 10
  seth 180
  forward 10
  seth 225
  forward 10
  seth 270
  forward 10
  seth 315
  forward 10
end
```

The "gusty" snowstorm was programmed like this:

```
to storm
  setup
  southwest
  gust
  southeast
  gust
  southwest
end
```

The setup procedure put all the turtles in place. Southwest and southeast combined the change in heading with the turtle movement. Here's what southwest looked like (southeast was a copy of southwest, with a change in heading):

```
to southwest
  seth 225
  repeat 100 [forward 45]
end
```

After programming a "storm," students created different shapes for each turtle. One student created two sets of shapes and wrote a pair of "changeshapes" procedures so that the storm had more variety of form. Another student stamped shapes in place, then changed shapes and continued the "storm" in order to fill the screen with snowflake shapes. Later she made colored snow as well. Eventually we put all the storms together on one page, introducing each student's variation with text in the Command Center. It was fun for everyone to watch.

These were fairly simple variations on a *Logo Writer* activity, but they provided plenty of space for individuality and for direct observation of nature. It gave us a chance to talk about many things: the nature of storms, wind directions and velocity, and even the idea of simulation, for after all, their snowstorms were simulations of the real thing happening right outside our windows. Interestingly, it seemed that every time this class worked on snowstorms, it really did snow—both outside and inside!

Telecommunications and Snowflakes

Last year on the LCSJ Logo Express bulletin board, there was a fascinating exchange from two teachers, Mark Steinberger and Jandy Bird, on classroom projects involving creating snowflakes. I stumbled into it after having worked on a similar project with one of my own fourth-grade classes. Such a resource for sharing is wonderful for teachers, who can compare teaching ideas and share work samples with one another. I came in a little too late to be a contributor for that year, but Mark and I later shared our "Snow Thoughts" at our local Logo User Group meeting. Mark's class printed and then cut out their snowflakes and suspended them from the ceiling of the classroom. What a great idea! Mark also explored (for himself) developing snowflakes from fractals. The story of snowflakes, snowstorms, and blizzards is sure to continue to grow this year.










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

Eadie Adamson is a member of The New Laboratory for Teaching and Learning of 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

Statement of Ownership, Management and Circulation (Required by 39 U.S.C. 3685). 1A. Title of Publication, Logo Exchange B. Publication No., 08886970. 2. Date of filing, September 17, 1991. 3. Frequency of issue, quarterly. 3A. No. of issues published annually, 4. B. Annual subscription price, \$29 in U.S. 4. Complete mailing address of known office of publication (not printer), ISTE, 1787 Agate St., Eugene, OR 97403-1923. 5. Complete mailing address of the headquarters of general business offices of the publisher (not printer), ISTE, 1787 Agate St., Eugene, OR 97403-1923. 6. Full names and complete mailing addresses of the publisher, editor, and managing editor, Publisher, ISTE; Editor, Sharon Yoder; Managing Editor, David Moersund; 1787 Agate St., Eugene, OR 97403-1923. 7. Owner, International Society for Technology in Education, 1787 Agate St., Eugene, OR 97403-1923. 8. Known bondholders, mortgages, and other security holders owning or holding one percent or more of total amount, None. 9. The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes has not changed during preceding 12 months. 10. Extent and Nature of Circulation. Average No. Copies Each Issue During Preceding 12 Months. A. Total no. copies (net press run), 1,550. B1. Paid Circulation, sales through dealers and carriers, street vendors and counter sales, 0. B2. Paid Circulation, mail subscriptions, 1,200. C. Total Paid Circulation (sum of 10B1 and 10B2), 1,200. D. Free distribution by mail, carrier or other means, samples, complimentary, and other free copies, 125. E. Total Distribution (Sum of C and D), 1,325. F1. Copies not distributed, office use, left over, unaccounted, spoiled after printing, 225. F2. Copies not distributed, returns from news agents, 0. G. Total (Sum of E, F1 and 2—should be equal to net press run shown in A), 1,550. Actual No. Copies of Single Issue Published Nearest to Filing Date. A. Total no. copies (net press run), 1,520. B1. Paid Circulation, sales through dealers and carriers, street vendors and counter sales, 0. B2. Paid Circulation, mail subscriptions, 1,225. C. Total Paid Circulation (sum of 10B1 and 10B2), 1,225. D. Free distribution by mail, carrier or other means, samples, complimentary, and other free copies, 115. E. Total Distribution (Sum of C and D), 1,340. F1. Copies not distributed, office use, left over, unaccounted, spoiled after printing, 180. F2. Copies not distributed, returns from news agents, 0. G. Total (Sum of E, F1 and 2—should be equal to net press run shown in A), 1,520. 11. I certify that the statements made by me above are correct and complete. Ellen Siegel, Distribution/Circulation Manager, ISTE.



All computer operating systems are slowly but surely evolving toward a graphics user interface (GUI). This style of computer can be recognized by the combination of windows, mice, and graphics. All Macintosh computers already have a graphics user interface, and most future IBM computers will have a graphic operating system such as Microsoft's Windows or IBM's OS/2. Many schools are even purchasing mice for some of their Apple II computers.

This new generation of computers will make it possible to develop multimedia applications that integrate text, graphics, sound, and video. Logo was one of the first educational applications that allowed teachers to investigate these possibilities. In this series of columns, we are exploring some of these capabilities.

In one recently developed Logo application, a class used Logo and a videodisc to display pictures of different zoo animals on the screen. The class then created a Logo program that asked students to count the number of animals shown in each picture, and graphed the results.

Charts and Graphs

Charts and graphs provide a method of interpreting scientific data. Investigations of these methods often begin at an early age. For example, the children in one first-grade class began keeping "data sheets" used to record objects that sink and float.

"Data Sheet"	
Float	Sink
leaf	rock
ball	marble
sponge	penny

Another teacher provided her students with pictures of zoo animals. The students were asked to count the animals in each picture and record the numbers in a table.
















Pictograms

Glen L. Bull and Gina L. Bull

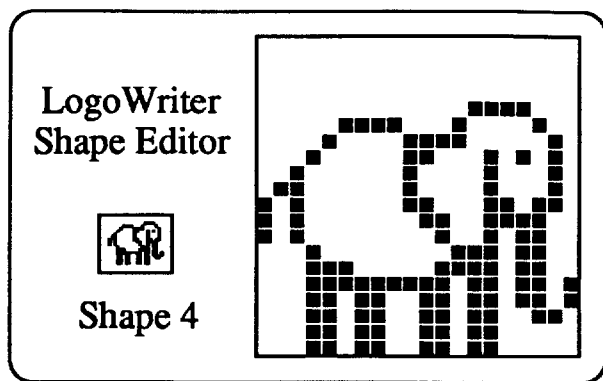
Table of Zoo Animals	
Picture	Number
Giraffe	2
Elephant	4
Camels	3
Reindeer	2
Boars	4

Creating a Pictogram

Data is often easier to compare when it is in a graphic format. The class used Logo to create a graphic chart of the data in the table. Placing the data in this form makes it easy to compare numbers of animals.

Giraffe	 
Elephant	   
Camels	  
Reindeer	 
Boars	   

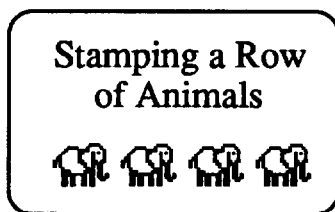
Creating a pictogram in Logo involves changing the shape of the turtle to another image. In most versions of Logo this can be accomplished with a shape editor of some type.



If Shape 4 is an elephant, the turtle can be turned into an elephant with the SETSH command in *LogoWriter*. (You will have to use the Shape Editor to create this shape first.)

```
SETSH 4
```

After the turtle has been changed to the new shape, it can be used to stamp a row of animals across the screen.



The following StampRow procedure was developed to facilitate this process. The StampRow procedure takes one input: the number of shapes to stamp. For example, to stamp a row of three elephants, the following command would be entered:

```
StampRow 3
```

The StampRow procedure calls one other procedure, StampShape, which does the work of stamping the shape. StampShape, in turn, calls the Over procedure to move the turtle over after each shape has been stamped.

```
To StampRow :Number
PD
REPEAT :Number [StampShape]
PU SETSH 10
END
```

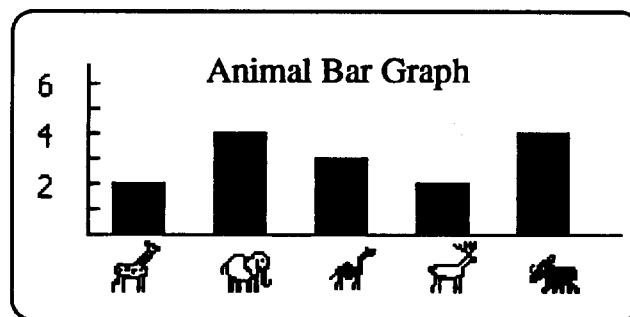
```
To StampShape
STAMP
OVER 25
END
```

```
To OVER :Distance
PU RT 90
FD :Distance
PD LT 90
END
```

From Pictogram to Bar Graph

To use the StampRow procedure, students in the class first set the turtle to the shape of the animal desired. They then used the *LogoWriter* Turtle Move keys (in Apple or IBM *LogoWriter*) or the mouse (in Macintosh *LogoWriter*) to move the turtle to the proper location on the screen. After the turtle was positioned, the StampRow procedure was used to stamp the appropriate number of animals for that row.

After the class created the pictogram of the zoo animals shown previously, they began to examine other ways in which the data might be expressed. The bar graph below is one of the other methods they developed to display the data using Logo. The bars in the chart were produced by developing a rectangle procedure to draw each column.



The transition from a table with data in it to a pictogram to a bar graph led to increasing levels of abstraction and sophistication. Over a period of several classes the students spent considerable time discussing the most effective means for displaying the data.


Adding a Multimedia Element


The applications described above could have also been developed with a number of graphing programs as well as with Logo. The class went further, and used Logo to control a videodisc player that displayed different zoo images on a television screen. The class used the *First National KidDisc*, which contains images from a trip to the San Diego Zoo, but there are a number of

other videodiscs that also have pictures of animals. (A single videodisc has the capacity to store more than 50,000 images per side.)

For example, when a picture of a herd of elephants is displayed on the television monitor, Logo asks the question, "How many ELEPHANTS are in the picture?" Using Logo to control the videodisc player and display the graphed data makes it possible to synchronize the picture and the graphed results. This type of interaction would be more difficult to achieve with a conventional graphing program. (In the illustration below, the accompanying picture on the television screen is not shown, since we have not yet identified a method of connecting a television monitor to *Logo Exchange*—there seems to be a problem of finding an extension cord long enough to reach from ISTE headquarters in Eugene, Oregon, to *your* house. <grin>)

How Many Animals Are in Each Picture?

Giraffe 

Elephant 

Camels

Reindeer

Boars

How many ELEPHANTS are in the picture? _

A videodisc player that can be controlled by a computer can be purchased for \$650. Several past columns in *Logo Exchange* have described the use of videodisc players with Logo. This particular class used a Pioneer 2200 videodisc player and *LogoWriter* for the Macintosh. The Macintosh version of *LogoWriter* has a built-in SEND command that makes it particularly easy to send a command to a device such as a videodisc player attached to the serial port, but similar procedures can be developed for other versions of *LogoWriter* and for other dialects of Logo, such as *Logo Plus*. In this instance, the following procedure makes it possible to find any of the 54,000 frames on the videodisc:

```
To Find :Number
SEND WORD :Number "SE
SEND CarriageReturn
End
```

```
To CarriageReturn
OUTPUT Char 13
End
```

For example, to find Frame 9,783, the student would type

```
Find 9783
```

In order for the command to work, of course, a videodisc player with the appropriate cable must be connected to the serial port of your computer.

Summary

The applications described here did not begin or end with the computer. The class also developed hand-drawn pictograms and bar graphs. However, the professional appearance of charts and graphs created with the computer were very satisfying.

Although several excellent charting and graphing programs are now available, use of a flexible program such as Logo makes it possible to incorporate elements such as sound and moving pictures from a videodisc. Multimedia elements open the way for even more sophisticated inquiries. For example, in one segment on the videodisc, the number of elephants in the herd changed as individual elephants moved on and off the screen. This led the class to a consideration of the complexities of charting events over time.

At present multimedia devices such as videodisc players are less common than they will be in the future. (A few years ago, it was also unusual to see a computer in the classroom.) As these peripherals become more common, new built-in commands will undoubtedly be added to Logo to make it possible to take advantage of these capabilities.

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

...beyond Piaget: Logo learning under a new light...

by A. J. (Sandy) Dawson

What a year it has been for me thus far, but I know that on the surface of it most people would conclude that it has been one long holiday. Wouldn't you, if you heard that someone you knew had spent the month of February in Hawaii, the month of March and half of April in Australia, the rest of April and half of May driving across Canada from Vancouver to Montreal, part of June exploring the eastern Canadian provinces of New Brunswick and Prince Edward Island, and then, just to top it off, had spent the month of July in Italy and Britain? But as with many things, appearances can be deceiving, because the time in Hawaii was spent at the University studying the teaching of Algebra I and a bit of non-Euclidean Geometry (e.g., see Gary Martin's column in the May, 1991, issue of *Logo Exchange* (Vol. 9, No. 8)).

In Australia I participated in the LME5 (Logo and Mathematics Education) conference in Cairns. Eastern Canada was the location of a couple of conferences that focused on mathematics and computers. The 15th Annual PME (Psychology of Mathematics Education) conference held in Assisi, Italy, included an examination of pedagogical issues surrounding the use of computers in and for mathematics instruction. And, finally, in England, the first-ever BCME (British Congress of Mathematics Education) had as one of its plenary addresses a report by Celia Hoyles on the three-year-long microworlds project she and her colleagues have recently completed.

I wouldn't even attempt to deny that these conferences were held in some rather wonderful and exotic places, or that I didn't enjoy the locations immensely. I would claim, however, that the main attraction of the conferences was not their locations, but the people who were there, the ideas they expressed, and the discussions that were engendered. It is three of these ideas I want to address in this column.

Development and Learning

At the LME5 conference in Cairns, Bruce McMillan, of the University of Otago Dunedin, New Zealand, advanced some Vygotskian notions that question the Piagetian idea that "development" must precede "learning," that children can only learn things when they have developed to a certain stage. In North America this hierarchy of concepts translates into a fixed mathematics curriculum that mandates, for example, that

children can only learn subtraction after they have learned addition, and that both of these activities must precede the learning of multiplication and division. On the contrary, Vygotsky argues that development and learning at the very least proceed hand-in-hand, and in many cases "good learning" is that which is well in the advance of development" (Vygotsky, p. 89).

Twenty years ago, Caleb Gattegno (1970) argued that there was only a temporal hierarchy, a time sequence in which ideas are met, but this hierarchy was not predetermined by the structure of mathematics. Indeed, Gattegno contends that children should meet algebra before they are introduced to arithmetic, because children intuitively used algebraic ideas as they acquired their mother tongue.

These are indeed challenging ideas, and they were echoed and extended by Willie Dörfler (1991) in his plenary address to the PME in Assisi, when he asserted "that by adequate didactical means much of the so-called higher level mathematics is clearly approachable, and the traditional systematic bottom-up approach could be avoided or partly circumvented" (p. 22). He goes on to say that his "approach to meaning and understanding is quite incompatible with Piagetian stage theories for cognitive development" (p. 23). And, finally, in terms of the use of the computer and particularly Logo in the teaching and learning of mathematics, we find Hoyles, Noss, and Sutherland (1991) contending that "the computer has the potential to overturn many of the assumptions about what children can and cannot do, the 'hierarchies' of understanding that have been painstakingly drawn up, and the 'readiness' of pupils to understand this or that mathematical concept" (p. 4).

"What has all this got to do with me?" you might well be asking. Just this—we have assumed for years that there is a more-or-less fixed sequence and composition to school mathematics, and that if this content could not be "learned" by children in that particular order, then there was something the matter with the children. This assumption is now being severely challenged, not just by armchair philosophers but by individuals moving into classroom situations armed with computers and software, including—but not restricted to—Logo. These researchers are finding that when given the freedom to explore their own constructions, children can develop mathematical ideas far beyond



those that conventional wisdom would suggest. Have you questioned your course syllabus recently? Have you explored how the use of computers and calculators might in fact enable students to engage mathematical ideas in different sequences than that which is laid down in the state or provincial curriculum? See the work of Gattegno (1974) for one suggestion of alternate sequences and content.

Assimilation and Accommodation

Of the three ideas I want to discuss in this column, the second is this: contrary, again, to what Piaget claimed, McMillan (in press) argues that Vygotsky "instead of emphasizing internal equilibration,...pointed to the significance of the dialectic between persons (especially the dialogue concerning the solution of problems) as the fundamental integrating function of human intelligence." Gattegno would even go further than this, however. He contended that the new—new ideas or new concepts or new ways of seeing the world—were not assimilated and accommodated, as Piaget maintained. Instead, he argued that the new integrates and subordinates the old to itself, and hence that learning takes place when the learner moves away from a state of equilibrium. Gattegno would not place as much emphasis on the dialectic between persons as Vygotsky does, reasoning that it is individuals who must ultimately organize their own views of the world. These views will be influenced by the feedback received from others, but it is still an individual task to construct the meanings attached to these dialogues.

Though not commenting on this latter issue specifically, Dörfler (1991) does say that "a kind of objectification and standardization occurs in the process of social communication (negotiation of meaning), and subjective deviations can there be corrected" (p. 22). Hence, while acknowledging the importance of the social dialectic, Dörfler, like Gattegno, seems to emphasize the centrality of the individual in the meaning-making process. The issue to me is not one of which is more central, because surely both individual and social processes are required. What is more interesting is the challenge to Piaget's now conventional view that one's knowledge grows by a process of assimilation and accommodation.

Teacher Intervention

The third idea harkens back to the debates that arose subsequent to the release of Papert's *Mindstorms*. It relates to the issue of how much a teacher should intervene, when this should occur, and under what circumstances. Older readers—older in the sense of having lived through the debates about whether or not Papert was claiming that when using Logo children would learn without help from teachers—will be inter-

ested to see that Hoyles et al. (1991) address this issue again in their report. They argue that those earlier discussions missed the point that the crucial role was played by the "culture" that Papert insisted should surround the use of the computer and of Logo. Where that culture was present—a culture in large part created by the teacher—the learner could and did become less dependent on the teacher. This, however, does not imply the teacher is not necessary in the classroom. Indeed, Hoyle, Noss, and Sutherland conclude that "research seems to indicate clearly that pedagogic considerations (including teacher-intervention) are a crucial component of the medium if children are to encounter and utilize the ideas which are embedded within it" (p. 3).

So the debate about whether teachers should intervene with children when they are using Logo is now over. The debate has moved to the issues of "the explicitness and timing of the pedagogical intervention[s] [that] are crucial in determining what kind of mathematical ideas pupils encounter, and the scope of the understanding they develop" (Hoyles et al., p. 3). Once again I am reminded of Gattegno and his concept of "forcing awareness." Embodied in this concept is the teacher's understanding of the essence of what is most necessary to a learner at a particular point in time for that learner to integrate previous learnings by subordinating them to this new idea. It is the artistry that every teacher seeks. Too often the search is conducted within the teacher, when in fact the light should be focused on the learner. It is in the learner that the teacher will find the answer as to what is most appropriate for the student. The teacher cannot find it within herself or himself because the teacher has previously understood the essence. In Gattegno's terms, teachers must subordinate their teaching to the learning of the student.

Nowhere is this more critical than when the teacher is contemplating an intervention into what the learner is doing. The teacher's role is to try to focus the learner's awareness—to "force" that awareness in a particular direction—by presenting learners with activities that are as unambiguous as possible, activities that pinpoint attention to the most important aspect of that which the student is trying to learn. Since each learner is an individual, and what that learner might need can vary widely from day-to-day, or almost from moment-to-moment, it is clear that teachers have a most demanding and challenging task every time students are engaged in dialogue.

The Complexity of Teaching

That is probably a good note on which to end this column, namely, with a recognition of the difficult and demanding task teachers face. The conversations I have had with people in the course of my travels this year to

New!

The Top 10 Reasons

Why You Should Upgrade to Terrapin Logo for the Macintosh 1.2!

(with apologies to Mr. David Letterman)

10. You can go from application to application in MultiFinder— your Logo design will be there waiting for you when you come back.

9. You can cut, copy, paste, shrink, enlarge, stamp, or crop a graphics region.

8. You can paste your Logo design directly into Word®, PageMaker®, MacPaint®, or any other program that can grab graphics from the Clipboard or the Scrapbook.

7. You can paste text from the Logo window into the Turtle window.

6. You can print any window (or your workspace) with a click of the mouse button.

5. You can resize, reposition, and reset your windows more easily with new commands.

4. You can move and turn the turtle with the mouse (or play with any of the other 24 new tool procedures).

3. You can load the Vehicles shapes file and have a road race.

2. You can use all your current Logo programs.

1. You can save the world! (or at least your complete Logo environment, using the new SAVEWORLD command)

To upgrade, send us your original Logo disk and \$25. Lab packs: return all disks along with \$25 for the first disk and \$7.50 for each additional disk. Macintosh site license upgrade: \$100.

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

many different countries, and at many conferences, all reinforce the high regard in which teachers are held, and the growing recognition of just how complex the teacher's job is. In these times of teacher-bashing in North America, such a recognition is indeed welcome.

Note: In my last MathWorlds column I mentioned several new books that used Logo as an instructional medium for the teaching and learning of secondary and tertiary mathematics. While in Great Britain I came across another such book, *Logo and Mathematics*, written by T. J. Fletcher, W. W. Milner, and F. R. Watson, which may be of interest to MathWorld readers. It is available for £10 from KMEP, Department of Education, University of Keele, KEELE Staffordshire, ST5 5BG, UK. Please make cheques payable to the University of Keele (KMEP).

References

- Dörfler, W. (1991). Meaning: Image schemata and protocols. In F. Furinghetti (Ed.), *Proceedings of the Fifteenth International Conference for the Psychology of Mathematics Education* (Vol. 1, pp. 17-32). Genova, Italy: Università di Genova.
- Gattegno, C. (1970). *What we owe children: The subordination of teaching to learning*. New York: Outerbridge and Dienstfrey.
- Gattegno, C. (1974). *The Common sense of teaching mathematics*. New York: Educational Solutions, Inc.
- Hoyle, C., Noss, R., & Sutherland, R. (1991). *Final report*

of the Microworlds Project: 1986-1989. Department of Mathematics, Statistics and Computing, Institute of Education, University of London.

McMillan, B. (in press). Logo as a framework for collaborative learning. In L. Nevile (Ed.), *Proceedings of the Fifth International Conference for Logo and Mathematics Education*. Cairns, Australia. : Australian Council for Educational Research.

Vygotsky, L. S. (1978). *Mind in society*. Cambridge, MA: Harvard University Press. As reported in McMillan, B. (in 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 address: Userdaws@sfu.bitnet



Sanity and the Single Computer

(Part 1)

by Judi Menken

The general trend in computers in education today, at least at the elementary level, seems to be toward putting computers back into classrooms. The hope is that they will be incorporated into the workings and philosophy of the particular class to better serve the interests of the classroom teacher and specific needs of her students. I notice inservice courses like "The One-Computer Classroom—Management Techniques" being offered at regional New York City Technical Assistance Centers. Generally the new wave in New York is already common practice in other parts of the country. However, unless others have found a way to put more than 60 minutes in an hour, teachers must still have the same problem of how to get more and more accomplished without running out of classroom space, time, or energy.

In the eight years I have been using computers and Logo, I have only used it in the setting of my primary grade classrooms. Over the years, I have evolved from a Texas Instruments 99A dinosaur of a set-up, complete with "state of the art" tape cassette that invariably went blank as my group of six 5-year-olds waited patiently, to my current two Apple IIe's and a printer. During those years, I have acquired a good deal of experience in how to make it possible for one teacher to establish a working computer center in a child-centered, curriculum-integrated environment. There are some tricks I'd like to share. It is certainly possible for young children to be virtually self-sufficient, not only in experimenting with Logo but also in properly operating and maintaining the equipment.

First, I see the computers as just a part of my classroom design. The computer area is one of the work areas of the room. The computer is another path toward the students' practicing and assimilating the skills and concepts of the early childhood world. It isn't going to be every student's favorite activity or strength (even though it may be a "biggie" with the parents). The computer area should not push in the boundaries of the block area or replace the hamster cage or workbench.

There are certain prerequisites, however, that must be accommodated. The computers should be in a quiet area of the room, have a good light source, be away from water sources, be relatively free of dust and chalk, and have a reasonable proximity to outlets.

Our computer area is in a corner of the rug/library area, right where the wall-length windows meet the

blackboard. We only use the board to hang Logo work, but the dust from the outside, well....

The outlets are about eight feet away, and the multiple plugs accommodate the record player and cassette player as well. The maze of cords hanging in back are tucked under the rug.

The single most important factor in managing an integrated day classroom is a physical arrangement that is accessible to the students, with adequate seating, comfortable traffic patterns (so that students can easily move about), easy access to materials, and a place where supplies are clearly marked, coded, and placed. The students will be able to run their own room if the expectations are clear and reasonable and the physical set-up supports and clarifies those expectations.

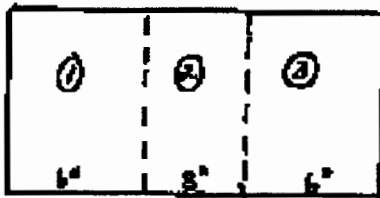
The student who is the computer monitor for the month takes the Logo disk of programs I have made for the class out of the box every morning. The front side is marked with a big red dot. Even the youngest, least sophisticated student can remember "red dot up, close the door, flip the switch." Remember, this is the Nintendo, VCR generation. The responsibility of being the only one except the teacher allowed to turn the computers on and off carries its own weight of serious attention. The same student reverses the process every day at clean-up time. There are student monitors who have similar responsibilities for the paints, the chairs, feeding the animals, and so forth—the students share the responsibility of running their classroom.

Now the machines are on, ready for a volunteer to sit down when he/she has finished an assignment and is allowed to choose an activity. Next problem: How to keep my reading or math group from being punctured with cries of "Judi, how do I...?" or "The turtle won't..." from across the room.

To begin, there is a startup page listing the menu of single-key programs contained on the disk. (See the end of this article for a model program to provide such programs for students.) Students have been taught these "games" at whole-group meetings or small-group lessons using mock-ups or other manipulatives before. Each is accompanied by a cue card sitting on the same table, again encouraging students to try to resolve their questions without adult involvement and leading them toward the use of written symbols to convey necessary information. (See the diagram below for a sample card and instructions on how to make a cardboard easel.)

To make CUE CARDS for independent computer work.

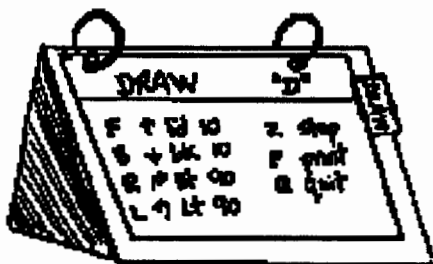
You need: 1 piece of cardboard: 9" x 15"
(cut up carton is great)
masking tape
index cards
2 binder rings



Fold on scoring to form a triangle. Tape along the top.



Use scissors to make holes. Punch matching holes on index cards. Put clearly printed cues on cards, one per Instant program. Use different colored markers for easy recognition. Add tabs to cards for easy retrieval.



As you add more programs to the repertoire, just add cards!

Of course there are regular glitches that prompt shouts for help. Leslie Thyberg's "Ask three before me." has to be the golden rule for any classroom, but, again, not just for computer work. It's another parallel between Logo and the curriculum in general; students should be working collaboratively in many contexts: opening a paint jar, explaining an assignment, soothing hurt feelings. Students turn to each other for help as well as to the adult(s) in the room.

Students have to ask me if they can print their work only because the color ribbons we use do not last long, so we can't have screen prints of purple with no other detail. Also, since other activities are in progress all

around the room, the constant buzz of the printer would drive us all to distraction. But once I give the nod, the student simply needs to type "P".

The screen will direct the student to type his or her name and then the date. All the student needs to do is make sure the printer is on and press the Return key for the next direction. There is a double-check procedure that allows the student to get back into his or her drawing if "p" was typed by mistake.

Very young students can handle sophisticated tasks independently if the process is broken into clear predictable steps. The system of the student managing the machinery independently mirrors the process of using Logo on that machine—simple steps that combine under the student's control and guidance to effect powerful results.

The Startup Program

Below is a sample startup program to provide menu for students. The names of the programs available would depend on the ones your students are using.

```
to startup
cc
ht
ct
print [G. Geoboard]
print []
print [D. Draw]
print []
print [S. Super.draw]
print []
print [SH. Shapes.draw]
print []
print [T. Target]
```

etc.

```
print []
(type [Type the letter for the game
you want to play. Press RETURN.]
char 13)
wait 50
cc
end

to g
ct
getpage "geoboard
end
```

... and so on for each program available on your menu. The programs listed must be on the same disk. If the children are up to using more programs than fit on one

screen of choices, add the print statement to your original menu:

```
print [MORE. to see other games]
```

and add more to the list of procedures:

```
to more
  getpage "others
end
```

"Others" calls up another page with its own startup program that lists your other choices.

In addition, all the programs may not fit on one side of your disk. Put them on the flip-side of the disk and use the following response for when the student types the letter(s) for that program:

```
to tm (or whatever your program is named)
  flipside
end
```

```
to flipside
  ct
  print []
  repeat 2 [print []]
  print [Take out disk and turn it
    over.]
  repeat 2 [print []]
  print [Press ESC.]
end
```

Nonreaders can be shown what to do and have no problem managing on their own after that. In fact, the whole language philosophy is brought into play. Students begin to "read" necessary symbols in the context of retrieving the program they want.

The following is the first basic single-key program they work with. We are setting up independent routines and processes now as much as "teaching Logo." Once these ideas are set, it's easy to build in sophistication.

The Draw Program

```
to startup
  st
  cg
  pu
  setpos [-100 50]
  setc 2
  pd
  seth 0
  stamp
  pu
```

```
right 90
forward 20
label [Hi! I am the Turtle.]
setpos [-100 30]
label [Let's draw a picture.]
setpos [-100 10] ht
label [Type GO when you are ready.]
end
```

```
to doodle
  cc
  name readchar "key
  if :key = "f [print "f forward]
  if :key = "b [print "b back 10]
  if :key = "l [print "l left 90]
  if :key = "r [print "r right 90]
  if :key = "q [ct cg doodle]
  if :key = "p [printout stopall]
  doodle
end
```

```
to go
  setc 1
  cg
  ct
  pd
  seth 0
  st
  doodle
end
```

```
to double.check
  (type [Are you ready to PRINT? (Y
    or N?)]) char 32)
  name readlistcc "okay
  if :okay = [n] [rerun]
end
```

```
to printout
  cc
  double.check
  cc
  type [Is the printer ON?]
  wait 30
  cc
  type [What is your name?]
  name readlistcc "artist
  type char 13
  (type [What is today's date?] char
    32)
  name readlistcc "date
  unlock
  printscreen
  cg
  I.D.
```




```

end

to I.D.
pu
setpos [-100 75 ]
ht
(label :artist [made this picture])
setpos [-100 55]
(label "on :date)
printscreen
cg
st
cc
lock
end

```

```

to rerun
cc
type [O.K. You can draw again.]
wait 30
doodle
end

```

Judi Menken has taught primary grade students for 14 years, the last 10 at a small, alternative public school in the middle of Manhattan. She uses Logo within an integrated-curriculum classroom filled with guinea pigs, books, paint, pencils, and, (oh yeah) computers. It *can* be done!

Judi Menken
 East Harlem Block School
 1615 Madison Avenue
 New York, New York 10029
 212/860-7967

Logo Computer Systems, Inc.

announces

LogoEnsemble

its newest learning environment.

LogoEnsemble combines Logo with a database manager, word processor, graphics system, and communications program.

LogoEnsemble for the Apple II GS has all the features that made *LogoWriter* such a versatile and popular program plus an easy-to-use database manager.

LogoEnsemble has database management capabilities that give students the opportunity to analyze, organize, collect, and explore data and discover relationships. New Logo commands such as **average** and **total** give the user spreadsheet-like capabilities. Word-processing features include a choice of 40 or 80 columns, cut, copy, paste, search, and replace as well as bold, underline, and italic. The copy and paste features can be used to copy information from a database to a *LogoEnsemble* page in order to create reports, graphs, or illustrations.

For more information, contact Logo Computer Systems, Inc., at 1-800-321-5646.

Unlock your school's potential by ordering your copy of *Education, Technology, and Paradigms of Change for the 21st Century*.

David Thornburg, leading educator, author, and designer, presents new ways of teaching and learning in the classroom.

Technologies converge with new concepts of learning and intelligence to provide unprecedented opportunities for all students.

Price: \$12.00
 (member price: \$10.80)
 plus \$4.50 shipping
 Canadian orders add
 7% GST (Reg.
 #128828431)



"Teachers have to be willing to move from the paradigm of lecture-based instruction to one in which they serve as catalysts and facilitators of student-based learning...."

David D. Thornburg, PhD
*Education, Technology,
 and Paradigms of Change
 for the 21st Century*

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 the customer.



The John Wayne Theory of Learning Lists

by Mark Horney

I was watching the late-night movies the other evening and happened to see the old John Wayne film *Hondo*. Toward the end of the film, Wayne discovers a young boy who can't swim. He unceremoniously picks up the boy by his britches, hauls him down to the fishing pond, and throws him in. The boy spits and sputters, thrashes about for a few moments, and then to the horror and amazement of his mother watching from the porch starts swimming like he was born to it. The Wayne theory of straight-ahead instruction triumphs again.

While I've never tried to teach someone to swim with this technique, I have used it to teach high school students the mysteries of recursive list manipulation. I resorted to this extreme after more humane lesson plans failed. What finally worked was very simple. I pulled together a set of unfinished list manipulation functions, demonstrated how to write procedures for one or two, described the standards I expected these list procedures to exhibit, and then I stepped aside. Predictably, my students moaned and complained. They struggled, they rebelled, they conspired to help one another, and I even caught one or two reading the textbook. Eventually, they succeeded.

Now let me hasten to add that I didn't leave them to sink or swim. I spent my time giving subtle hints and encouragement, providing deviously constructed test lists, pointing out that some recalcitrant procedures require paper-and-pencil analysis, and generally upholding standards of elegant Logo programming. Because I wasn't tied to the lectern, I was able to concentrate my efforts on the students who needed help most, and let my advanced students push ahead. Since my students weren't stuck at their desks listening to me, they were able to listen to and help one another. They found out for themselves the benefits of collaborative learning and the expediency of looking up answers in the book. They learned these valuable lessons, and they also learned about lists.

Two things are needed to implement the John Wayne Theory of Learning Lists—a suitable set of list functions with which the students can struggle, and standards for judging when they should be awarded their (metaphorical) water wings.

The Pond

What follows are 35 specifications for list functions, which I've ordered more or less by difficulty. There

isn't anything special about these functions; almost any group of list functions will do. This particular group has a large number of functions using lists like vectors and matrices since other assignments in my class required such functions. Whether you use this or some other group of functions, the functions should have some important features. The list of functions should:

- be *long and varied*. (Students should have a sustained experience, and encounter many different kinds of problems.)
- have both *easy and hard* functions.
- have some *ambiguous* functions. (This forces students to consider the ambiguities that, in reality, exist in all problems.)

TO LINEPRINT :list

Print *list* with each word on a separate line.

TO ISLIST :list

Returns TRUE if *list* is a list, otherwise returns FALSE.

TO NUMERICALLIST :list

Returns TRUE if *list* consists of numbers, otherwise returns FALSE.

TO REVERSE :list

Returns *list* with its words in reverse order.

TO COUNT :list

Returns the number of words of *list*

TO APPEND :list1 :list2

Returns a list with the words of *list2* appended to the end of *list1*.

TO INTERWEAVE :list1 :list2

Returns a list of words taken alternately from *list1* and then *list2*. Assumes *list1* and *list2* are the same length.

TO LISTMAKE :length :word

Returns a list of *length* words

TO SAME :list1 :list2

Returns TRUE if *list2* is exactly the same as *list1*, otherwise returns FALSE.

TO GET :number :list
Returns the word *number* from *list*. Assumes
number ≤ length of *list*.

TO PUT :number :word :list
Returns a list with *word* substituted for the
number word of *list*. Assumes *number* ≤ length
of *list*.

TO SUM :list
Returns the sum of the words of *list*. Assumes
list consists of numbers.

TO PRODUCT :list
Returns the product of the words of *list*. As-
sumes *list* consists of numbers.

TO POWER :base :exponent
Returns *base* raised to the power *exponent*. As-
sumes *base* and *exponent* are whole numbers ≥
1.

TO AVERAGE :list
Returns the average of the words of *list*. As-
sumes *list* consists of numbers.

TO MEDIAN :list
Returns the median of *list*.

TO MIN :list
Returns the smallest number of *list*. Assumes
list consists of numbers.

TO MAX :list
Returns the largest number of *list*. Assumes *list*
consists of numbers.

TO RANDOM :number :min :max
Returns a list of *number* random numbers each
≤ *max* and ≥ *min*.

TO FLATTEN :list
Returns a list of the words found in *list* or any
sublist of *list*.

TO SORT :list
Returns *list* sorted into alphanumerical order.

TO TALLY :list
Return a list of lists with the sublists giving the
number of occurrences of each word of *list*.

TO CONTAIN :word :list
Returns TRUE if *list* contains *word*, otherwise
returns FALSE.

TO FIND :word :list
Returns the position of *word* in *list* if *list* con-
tains *word*, otherwise returns FALSE.

TO VECTORADD :vector1 :vector2
Returns a list of numbers that are the sums of
corresponding words from *vector1* and *vector2*.
Assumes *vector1* and *vector2* are of the same
length and are composed of numbers.

TO SCALARMULTIPLY :scalar :vector
Returns a list of numbers consisting of the
words of *vector* each multiplied by *scalar*. As-
sumes *vector* words and *scalar* consist of num-
bers.

TO VECTORMULTIPLE :vector1 :vector2
Returns a list of numbers that are the products
of corresponding words from *vector1* and *vec-
tor2*. Assumes *vector1* and *vector2* are of the
same length and consist of numbers.

TO DOTPRODUCT :vector1 :vector2
Returns a number that is the sum of the prod-
ucts of corresponding numbers from *vector1*
and *vector2*. Assumes *vector1* and *vector2* are of
the same length and consist of numbers.

TO MATRIX :rows :columns :entry
Returns a matrix with *rows* and *columns* of
entry. Assumes that *rows* and *columns* are whole
numbers and that an NxM matrix is formed by
a list of N sublists, each consisting of M words.

TO IDENTYMATRIX :rows :columns
Returns an identity matrix of *rows* by *columns*.

TO MATRIXGET :row :column :matrix
Returns the entry from the *row* and *column* of
matrix. Assumes *matrix* has *row* rows and *col-
umn* columns.

TO MATRIXPUT :row :column :matrix
:word
Returns *matrix* with *word* substituted as the *row*
by *column* entry of *matrix*. Assumes *matrix* has
row rows and *column* columns.

TO MATRIXADD :matrix1 :matrix2
Returns a matrix which is the component-wise
sum of *matrix1* and *matrix2*. Assumes all matrix
words are numbers and that *matrix1* and *ma-
trix2* have the same number of rows and col-
umns.

TO MATRIXSCALARMULTIPLY :number
:matrix

Returns *matrix* with each entry multiplied by *number*. Assumes *matrix* entries and *number* are numbers.

TO MATRIXVECTORMULTIPLY :vector
:matrix

Returns a matrix that is the product of *vector* and *matrix*. Assumes *vector* and *matrix* entries are numbers and that the length of *vector* is equal to the number of rows of *matrix*.

TO MATRIXMULTIPLY :matrix1 :matrix2

Returns a matrix that is the matrix product of *matrix1* and *matrix2*. Assumes all matrix words are numbers and that *matrix1* and *matrix2* have the same number of rows and columns.

Standards

It isn't enough for students to simply hack together procedures for these list functions. There must be standards for students to work toward. The standards I established fell into two categories: Truth and Beauty. The standards for Truth *must* be met by every procedure:

- Procedures must be *effective*. They must always return the appropriate value when given the proper parameters.
- Procedures must be *compatible*. Every procedure must accept appropriate outputs from any other.
- Procedures must be *innocent*. Procedures must never require any further assumptions than those specified in the function definition.
- Procedures must be *primitive*. Procedures must not use any special Logo commands.

I have seven standards for Beauty:

- Procedures should be *independent*. No function should depend on the presence of another.
- Procedures should be *singular*. List functions should be implemented with a single procedure and not a cluster of special purpose procedures.
- Procedures should be *parsimonious*. Procedures should limit the use of variables.

- Procedures should be *local*. Procedures should avoid global variables.
- Procedures should be *clear*. Procedure algorithms should be readily apparent.
- Procedures should be *efficient*. Procedures should contain no more steps than are necessary for effectiveness, and no fewer than are needed for clarity.
- Procedures should be *robust*. Procedures should require fewer or less restrictive assumptions whenever possible.

While the need for standards of Truth is self-evident, the rationale for Beauty is more subjective and more difficult to communicate to students. My students were slow to accept the idea that characteristics that may be impossible to achieve in some procedures were still important. But eventually, they stopped sputtering and thrashing about and saw that Beauty was important in its own right, that the effort to achieve it had hidden benefits, like an increase in creativity, that could be had regardless of the final virtue of any particular procedure.

Let me reiterate: there is nothing special about either this list of functions or this group of standards. All the John Wayne Theory requires are a list and some standards. Pick those that best fit the circumstances, and then, as a prominent shoe company urges, JUST DO IT.

Mark Horney spent 13 years as a middle and high school teacher before moving to the University of Oregon to complete a degree in educational computing. He currently works as a research associate on Project Outline which investigates the use of computer applications for electronic studying. Dr. Horney also works in the area of hypertext.

Mark Horney
P.O. Box 3697
Eugene, OR 97403

mhorney@oregon.uoregon.edu

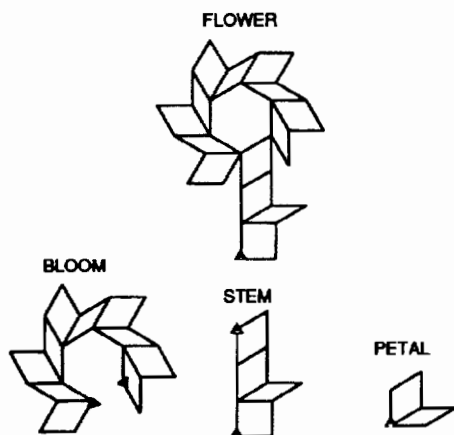
Higher-Level Math Thinking: Part I

by Douglas H. Clements

Pretend you are a ninth-grader taking a final exam. You have a generalized rhombus procedure as follows.

```
TO RHOMBUS :SIDE :ANGLE
REPEAT ____ [FORWARD :SIDE RIGHT ____
FORWARD :SIDE RIGHT ____ ]
END
```

You are to fill in the missing values or variables in this procedure on the RHOMBUS MADNESS worksheet before going to the computer. Then, create the following design shown on the worksheet using the following three rhombus structures: RHOMBUS 25 60, RHOMBUS 25 30, and RHOMBUS 25 90. Use the three subprocedures BLOOM, STEM, and PETAL.



Please work through that exam now. Get a friend or student to solve the task as well.

The Return of the Atlanta-Emory Logo Project

Way back when (Clements, 1987), I discussed the Atlanta-Emory Logo Project, in which Olive, Lankenau, and Scally investigated the use of Logo to teach geometric relationships to ninth graders. I also promised to keep you informed. New information is available, and it is interesting. The RHOMBUS MADNESS task was one of the final exam measures.

In the new publication, John Olive (1991) strives to understand the results of the project by pulling together several different theories of learning geometry. I'll review them briefly here, because you'll need to be familiar with them to appreciate Olive's results. Those already familiar may wish to skip the next section. Those who wish to have more information can see previous articles on the theories (Battista & Clements, 1988; Clements, 1987; Clements & Battista, 1987).

Theories of Learning Mathematics

van Hiele's Levels of Geometric Thinking

I've discussed this model frequently in this column, but Olive (1991) uses slightly different levels.

Visual: The student sees figures in their entirety and is unable to analyze their properties.

Descriptive: The student describes the properties and relations of figures and can use them in inductive arguments, but only in a case-specific manner. He or she can use these properties to form definitions of the figures but cannot yet identify class inclusions (e.g., that all squares are rectangles).

Theoretical: The student generalizes the logical relations that exist among figures and their parts and operates on those relations deductively.

According to van Hiele, students don't move from one level to the next without instruction that passes through a series of phases. If, instead, teachers use concepts and language from a higher level, students will merely memorize instead of understanding important relationships. This leads us to Skemp's (1976) view of mathematical understanding and his idea of "instrumental mathematics."

Skemp and Learning With Understanding

Richard Skemp (1976) believes that there are "two effectively different subjects being taught under the same name 'mathematics'" (p. 22). One subject, *instrumental mathematics*, consists of "rules without reasons." The other, *relational mathematics*, is "knowing both what to do and why." It involves building up conceptual structures from which a student can produce an

unlimited number of rules to fit an unlimited set of situations.

Biggs and Collis' SOLO Taxonomy

The SOLO (Structure of Observed Learning Outcomes) taxonomy (Biggs & Collis, 1982) helps evaluate the *quality* of a student's response to a task. I'll give a general description and then an example of how a student might respond at that level using a generalized Logo procedure POLY.

```
TO POLY: TIMES :SIDE :ANGLE
REPEAT :TIMES [ FORWARD :SIDE RIGHT
:ANGLE ]
END
```

Prestructural: Learners do not engage in the task, or they respond inappropriately. POLY 40

Unistructural: Learners are able to use only one piece of information. POLY 50 50 50

Multistructural: Learners are able to use several pieces of information but can't relate them. POLY 20 30 40

(different numbers but no meaningful relationship between the numbers and their role in POLY)

Relating: Learners integrate the separate pieces of information to produce a variable solution to the task. POLY 4 50 90

Extended abstract: Learners can derive a general principle from the related pieces of information that they can apply to new situations.

```
TO TRIANGLE :SIDE
POLY 3 :SIDE 120
END (use of POLY within another
procedure's definition)
```

Combining Skemp's theory with SOLO, we see that students can move through the SOLO levels in two ways. They may move relationally, by thinking about one's own investigations. Or they may move instrumentally, by memorizing what some text or teacher said. If students learn by rote, they are building on sand. They can't relate new knowledge to old knowledge, so they have limited their future learning to mere memorization.

Three Theories and the Learning of Geometry

It is no surprise, then, that students' learning of geometry is so poor. Higher concepts are often taught instrumentally. Even when students learn some elementary geometry concepts relationally, there are gaps. For example, curricula often ignore the descriptive level in the van Hiele hierarchy. Students learn about more shapes but not more about those shapes. Similarly, the middle levels of the SOLO taxonomy are frequently omitted. Curricula do not ask students to relate and integrate their knowledge of shapes.

Goals of the Project

The goal of the Atlanta-Emory Logo Project was to fill in these gaps. Logo helped break the vicious cycles of rote learning by giving students experience relating ideas to each other.

Why? How? According to Olive (1991), turtle geometry has all the necessary ingredients for generating relational learning. Students can relate personally to a learning experience. They can relate this experience to their existing ideas. They can reflect on what they know.

Did it work? Did the three theories help explain what happened? Olive analyzed the Logo work of the 30 ninth-grade students in the project to answer these questions. Mostly he examined dribble files of the students' Logo work on four tasks, including RHOM-

ISTE's First Conference!

**October 1-3, 1992
Denver, Colorado**

**Reserve these dates
and watch for
more information**

BUS MADNESS. (Dribble files are records of everything students type in, which is stored automatically on disk.)

Results

Ninth-grade students in two inner-city high schools took a course. The focus was on informal explorations of topics such as polygons, circles, and transformations. High school mathematics teachers implemented a Logo learning environment based on a guided discovery approach. Students generally worked in pairs. (Specific activities can be found in Clements [1987], as well as in the project's reports; see the address at the end of this column.)

The students' remarks about their experiences in the Logo class were very positive. They found the class intriguing and exciting. They perceived the relaxed atmosphere of the class to be better than that of their other classes, because the students and teacher were more comfortable with one another. The students reported being less frustrated and more involved in the Logo class than they were with their other classes. They believed that they had learned a great deal about mathematics, especially about angles, quadrilaterals, and the Pythagorean theorem. They thought Logo had helped them with problem solving.

Olive's (1991) discussion of an excerpt from a student's dribble file may help make the results and the theories more concrete. The student, Vincent, was working on RHOMBUS MADNESS:

He successfully defined his generalized RHOMBUS procedure and immediately used it to create PETAL and STEM without any difficulties. He did not make PETAL state transparent, however, which caused some problems later. On the other hand, he made STEM state transparent but immediately realized that he did not want STEM that way and changed it. Vincent took an unusual approach in creating BLOOM. He began by defining BLOOM with only the turning angle (between STEM and BLOOM); he then tried out each of his attempts for the rest of the BLOOM structure by adding them to his original definition in the editor. (p. 103)

What does this strategy indicate about Vincent's thinking? Can you connect it to the theories? According to Olive, Vincent's BLOOM strategy indicated a relating-level of use of procedures in the SOLO taxonomy and a descriptive approach to the task in the van Hiele model. Also, Vincent started the task between the *relat-*

ing and extended abstract levels of the SOLO taxonomy, because he used a generalized RHOMBUS well. Olive continues:

Vincent ran into difficulties when he continued to work on BLOOM. His strategy was to use RHOMBUS instead of PETAL as his building block. He ended up reconstructing the PETAL structure but had trouble figuring out the angle for adding the next part. He went through a period of multistructural explorations with occasional unistructural commands. Finally, he seemed to realize that BLOOM consisted of five PETAL procedures and started to use PETAL as his building block for BLOOM. He quickly arrived at the correct relationship between each PETAL in his BLOOM procedure. However, he apparently assumed his PETAL procedure to be state transparent, which it was not, so his BLOOM procedure did not work. After some explorations he made PETAL state transparent and immediately realized that he needed to change STEM and did so without any problems.... From this point he had no problems with BLOOM and finally combined his procedures into a superprocedure called FLOWER. (pp. 103-104).

What do you say about Vincent's work? What SOLO level did he show? What van Hiele level? Does he show relational understanding?

Here are Olive's interpretations. Except for problems in BLOOM, Vincent appeared to have relational understanding of Logo and of geometry, especially angles. He usually worked on the descriptive level of the van Hiele model. For example, he used the editor a lot. He used a visual approach when he ran into difficulties, but returned to the descriptive level quickly.

Finally, most of his work was at the relating level of the SOLO taxonomy. For example, when he changed PETAL and then STEM, he showed a relating-level use of subprocedures as building blocks. His use of generalized procedures shows some understanding of the extended abstract level. Again, when he had difficulty, he regressed to the multistructural and unistructural levels, but recovered to higher levels.

You may wish to consider your own solution of the RHOMBUS MADNESS task, and the solution of any others you had solve it. What levels in the van Hiele model and in the SOLO taxonomy can you see? Do the solutions show instrumental or relational thinking?



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)

Other Results—To Be Continued

Vincent's work seems to show a fairly high level of response. Also, there is a hint that the different theories are connected. What about the rest of the students? What levels were they on? How did the levels relate to one another? Were some SOLO levels required for learning ideas at higher van Hiele levels? What implications do the results have for teaching? In the next issue, we answer these questions.

References

- Battista, M. T., & Clements, D. H. (1988). A case for a Logo-based elementary school geometry curriculum. *Arithmetic Teacher*, 36, 11-17.
- Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning: The SOLO taxonomy (Structure of Observed Learning Outcome)*. New York: Academic Press.
- Clements, D. H. (1987). The Atlanta-Emory Logo Project. *Logo Exchange*, 6(2), 22-25.
- Clements, D. H., & Battista, M. T. (1987). Why Logo for learning mathematics? Logo-based geometry. *Logo Exchange*, 5(6), 20-21.
- Olive, J. (1991). Logo programming and geometric understanding: An in-depth study. *Journal for Research in Mathematics Education*, 22, 90-111.

Skemp, R. (1976). Relational understanding and instrumental understanding. *Mathematics Teaching*, 77, 20-26.

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 in 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 in 1991. He is currently working with several colleagues 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

Starting With Scribbles

by Eleonora Badilla-Saxe

Our work with *LogoWriter* in Costa Rican primary school laboratories, as well as education in general, is undergoing great change. The most convincing sign of this change is evidenced by the presence of the computer screens that line the walls of our laboratories. Yet we must not forget that the most significant change is that which transpires in the minds of the children; the changes we see in the number of computer screens is merely a reflection of this development process.

The Costa Rican experience in employing computers in education was initiated approximately two years ago, and at this point it is still too early to scientifically affirm the real process by which the Costa Rican children learn while using *LogoWriter*. However, I will venture to make a brief analogy between the children's learning process while using *LogoWriter* and while using other tools.

We have repeatedly read or heard it said by Dr. Seymour Papert, creator of Logo, that the computer as an educational tool may be compared to the pencil, another instrument of learning. If I may be permitted, then, the analogy I wish to establish specifically refers to the process by which children learn to utilize both of these instruments.

I should make it clear from the start that in this mental exercise, I have confined myself to the analysis of a process visible in the designs seen on the computer screen. That does not mean in any way, however, that I am ignoring the very important mental process that develops in the users of *LogoWriter*. It means only that the product that appears on the computer screen is "the window through which we can peer into the minds of the children," as one teacher at Hennigan School in Boston has said.

I. The Scribbling Stage

In giving a small child a pencil or crayon, the objective is to initiate a process by which the child learns to utilize and master the writing tool in a way that will allow him/her to draw and write more complicated things each time.

According to the description of this process by Viktor Lowenfeld and Lambert Brittain, the individual holding in his/her hands a pencil (crayon, chalk, etc.) for the first time accomplishes "a permanent register that generally takes the form of scribbling."

Normally, adults interpret this scribbling as a waste of time, and we hurry to push the child to make more defined designs.

However, scribbling is a very important step in a child's development, "for it is the beginning of expression, that not only leads (the child) to drawing and painting, but also to the written word." That is to say, scribbling is the stage of development that allows the child to become familiar with the instrument he/she is using, to experiment with it, and, little by little, to gain mastery over it.

Generally, the scribbling stage is divided into three principle categories: disorderly scribbling, controlled scribbling, and scribbling with a name.

Lowenfeld and Brittain might describe these categories in the following manner:

1.1 Disorderly Scribbling

The first lines drawn with a pencil (crayon, chalk, etc.) generally do not make any sense, and the child does not seem to realize that, with these same lines, he/she can make anything that he/she wants. Some parents or teachers try to discern something recognizable in this scribbling. However, while a child is in the disorderly scribbling stage, drawing a picture of something "real" is inconceivable.

1.2 Controlled Scribbling

At some point, the child will discover that there is a connection between his/her movements and the lines that he/she executes on the paper. Although there is no apparent difference between some scribbles and others, achieving control over his/her movements using the pencil is a vital experience for the child.

The scribbles at this point are much more elaborate and, in some cases, the child discovers with great happiness certain relationships between what he/she has drawn and something in his/her environment.

1.3 Scribbling Given a Name

This new stage is of great consequence in the child's development. At this time, the child begins to give names to his/her scribbles, although one can still not recognize from the picture what the child has indicated. This disposition to give a name to the scribbling is very significant, for it indicates that the child's thinking has changed.

1.4 The Importance of Color

Color is an element to be considered in the scribbling stage. In the first two categories, it is important that the colors be contrasting (white on black or vice-versa). This contrast facilitates an appreciation for what has been done.

When the child enters the stage of giving names to his/her scribbling, he/she truly wishes to use different colors for different meanings. One of the first stages in the process of perceiving color is simply distinguishing between different colors.

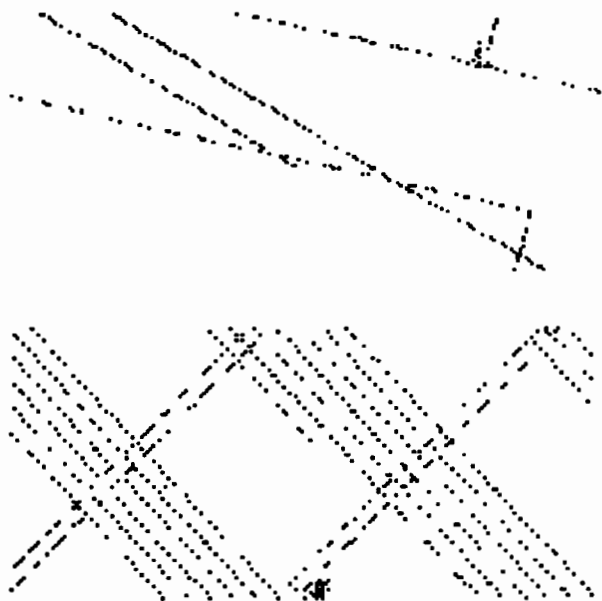
1.5 The Scribbling Present in Logo

1.5.1 Disorderly Scribbling

It is frequently observed that the child's first contacts with *LogoWriter* (almost regardless of age) result in scribbling on the computer screen.

The free exploration of the possibilities of the turtle, using the first commands generally learned—forward, back, right, and left—generates disorderly “stripes” on the screen that vary according to the input by which they are accompanied. Initially, short lines are produced by using numbers like 10, 40, 90, etc. When the student discovers that the turtle responds to “large” numbers—1000, 4000, 5789, 6732, etc.—the scribbling then encompasses the entire screen.

Borrowing terminology from Lowenfeld and Brittain, this first stage would be disorderly scribbling; it is unintentional but serves to introduce the child to the world of the turtle. It allows him/her to explore using the space of the computer screen and the possibilities of the turtle.



Alejandra León. 7 years.

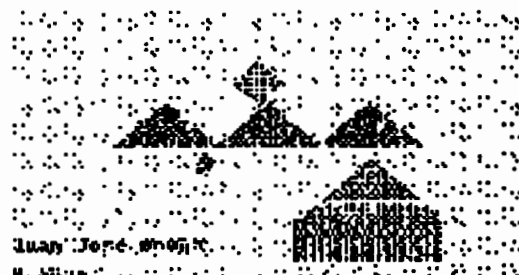
On occasion, one can perceive that using Logo for the first time to “scribble” produces satisfaction in the child. We would be able to increase the level of satisfaction if the scribbling could be done in different colors.

Let us, then, allow the child to experiment with this new instrument. He/she is learning to utilize it and deserves time to exercise his/her mental and motor skills.

A child that has been through this process before will possibly start to gain control over the turtle and, gradually, his/her scribbling comes to reflect his/her intentions. At this point, we are able to recognize familiar objects from our environment between the lines drawn by the turtle, as directed by the student.

1.5.2 Controlled Scribbling

The students who have passed through this process before will possibly start to have control of the turtle and little by little their scribbling will show an intention. We could at this moment possibly recognize some objects from our daily life that are found among the lines that the turtle leaves.



Juan José Acuña. 8 years.

1.5.3. Scribbling Giving a Name

Eventually, the moment arrives when the child points to the screen and says, “This is my house.” Depending on the age of the child, the person in charge of guiding his/her learning with Logo is then able to label the screen or help the child to learn how it is done.



SANTA

Cristian Conteras. 7 years.

It is important that this process be respected in the student. We should not perceive it as a waste of time or resources, for it is not. It is an investment that the apprentice must make in order to gain mastery, little by little, over his/her new tool.

II. The Preschematic Stage

According to Lowenfeld and Brittain, this stage arises directly from the last scribbling stages. It is a very important stage for the child because it reflects the conscious creation of forms that have some relation to the world around them. In the scribbling stage, the child is involved primarily in a kinesthetic activity. In this new stage, however, the child is trying to establish a relationship with that which he wishes to represent. This activity brings great satisfaction to the child.

These pictures are important not only for the child but for the teacher as well, for they are a testament to the child's mental process.

In this stage, there is little relation between the color chosen and the object represented. A man could be red, blue, or yellow. The use of color is a fascinating experience. Although the child does not wish to establish any definite relationship between color and object, he/she enjoys using color to suit his/her taste.

2.1 The Preschematic Stage in Logo

When the child's designs using the turtle begin to correspond to a particular intention, we can feel satisfied, for it is evidence of progress—progress in the child's mental process on the one hand and progress in the child's mastery of the computer as an educational tool on the other.

At this stage, if the child has already achieved the creation of a conscious design, it is likely that he/she will soon be able to use Logo to gain knowledge specifically related to the various subject areas covered by the basic school curriculum. It is recommended at this point, if the child's age permits, that he/she be allowed to use the Logo word processor to complement his/her creations.



Daniella Capra. 9 years.

III. The Schematic Stage

After much experimentation, the child is able to represent the concepts that he/she has about human beings and their environment. The scheme the child uses can be determined by the way in which the child sees something, the affective meaning the child assigns to it, the tactile impression of the object, or the way in which it functions or behaves. The principle discovery during this stage is the existence of an order in spatial relations.

The child will no longer think about "a man, a tree, a car" without relating them to each other. At this point, he/she is able to organize them spatially: "I am on the ground, the car is near me, the grass grows in the ground." For this reason, it is very probable that the baseline will appear as part of the scenery.

At this stage of the process, it is common for different time sequences or spatially distinct images to appear in the same design. The child is able to use various planes or elevations. Also, "X-Ray" designs may appear. These designs are used to describe the interior and exterior of a closed environment simultaneously.

During this stage, the child may possibly establish a definite color for each object. Thus, the same color will always be used for the same object, i.e., "the sky is blue," (always) and "the roof is red," (always). This process of generalization is very important because it leads to abstract thought and demonstrates that the child is able to use his/her own experience to generalize about other situations.

3.1 The Schemes Utilizing LogoWriter

As the child feels more comfortable with the machine and begins to identify with the turtle, he/she continues to gain mastery over Logo's possibilities. Little by little the child needs more primitives and begins to write his/her own procedures. This process is reflected in the appearance of more defined schemes on the computer screens. Some children also introduce a baseline and, at times, we find "X-Ray" designs.



Gabriela Burgués. 11 years.



Each time, the designs are more complex as a result of the child's mental structure becoming more complex each time. At this point, the child is in control. The child masters the tool better each time and executes clearer concepts or ideas.

IV. Toward More Complex Stages

With every learning experience, including the experience of working with *LogoWriter*, the child continues to progress toward more complex stages. His/her work both in the notebook and on the screen reflects this progress.

When the teacher detects that the child has progressed to the schematic stage, it means that the child has gained mastery over the turtle and soon will advance to other stages. This is the time to start to propose significant projects to the child that on the one hand permit him/her to refine his/her mastery of Logo language and on the other hand to relate the work realized in the laboratory to that developed in the classroom.

4.1 Relating Work to the Curriculum

Within an open learning environment, the person in charge of guiding the students through the laboratory experience can suggest the general framework of the project, that is, propose it or the themes with which to work. It is highly recommended, then, that the laboratory directors consult with the regular classroom teachers or the grade coordinator as to the themes in the regular curriculum that can be developed through the projects using *LogoWriter*.

Once the theme has been chosen, it is necessary to give each student the opportunity to develop his/her creativity and initiative by approaching the project in a individual manner.

4.2 The Role of the Laboratory Directors

4.2.1 Facilitating Learning

The Logo environment proposed by Dr. Papert suggests that the students construct their own knowledge through the exploration of the Logo language as a tool. This implies that recommendations exist as to what the teacher might do in order to achieve this environment.

Sharnée Chait and Judy LeGallais, authors of the chapter entitled "The Focus on Projects" in the teachers' manual published by the LCSi company, give us some suggestions in this respect:

- Allow exploration:

The students need time to experiment with new ideas.

- Offer ideas without always supplying the answers:

The director of the laboratory can help students to analyze their problem areas. There are those who have a criteria for knowing when to offer a direct answer and when to allow the student to find it out on his/her own. If a student feels overwhelmed or frustrated by a particularly confusing situation, it is important to provide direct answers so that the student overcomes the problem immediately. In other cases, it is more useful to give suggestions or pose questions and allow the student to try to discover the answer by himself/herself.

- Introduce new concepts and techniques when the students are prepared:

The exact time you choose to introduce a new Logo concept can make a great difference in the size of the effect it has on students. We should try to figure out if they are ready or not beforehand.

- Review the concepts and techniques learned:

Frequently, at the completion of a project the students have applied new concepts and techniques without truly understanding the process involved. Reviewing the concepts and techniques after the students have already applied them in a project reinforces what they have learned.

4.2.2 Evaluating

Chait and LeGallais think that one of the best ways by which students evaluate a completed job is when they are encouraged to show each other what they have achieved in their respective projects. In this manner, they will realize that it is possible to accomplish a project by different methods, to fulfill a purpose via various avenues, and to resolve a problem in many ways.

This gives them the opportunity to speak about the kinds of problems they encountered and how they might solve them.

For his/her part, the teacher can get an idea of the students' progress both by observing the results of the project and, more importantly, by analyzing the students' attitude when confronted with:

- mistakes. (What types of mistakes do the students make? How do they solve them? How do they utilize the opportunity to learn something new?)

- the challenge of being creative. (Do they change or modify details in order to make the project more personal?)
- relating to their classmates. (Do they accept group work or working in pairs? Do they cooperate with each other? Support new ideas? Accept the ideas of others?)



Daniel Solano R. 10 years.



Jorge Manuel Castro, Asdrubal Jose Alvarado.

4.3 Advantages of the Project

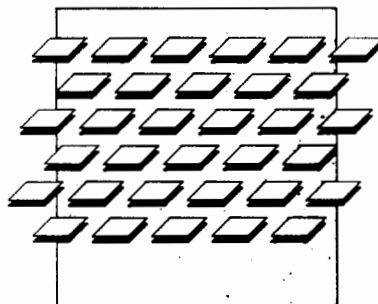
For Chait and LeGallais, projects allow the students to control their own learning process because the students establish the projects' purpose, plan it, construct it, analyze it, correct the mistakes, and evaluate the final product all by themselves. Also, projects give the students the opportunity to develop initiative and creativity.

On the other hand, the students learn in a significant context. That is, the project becomes their own creation; there is an affective bond established between project and student. This bond motivates them to learn new concepts (about the theme or Logo language) so that they may be able to complement their creation.

In addition, the learning is individualized, and each student or pair of students is able to work at his/her/their own pace and in accordance with his/her/their own style.

This emphasis on individual learning makes evident the comprehension and confusion levels of each individual student, which would be very difficult if all the students accomplished a project in exactly the same way.

Learning With Computers: Effective Teaching Strategies



by Ken Ryba and Bill Anderson

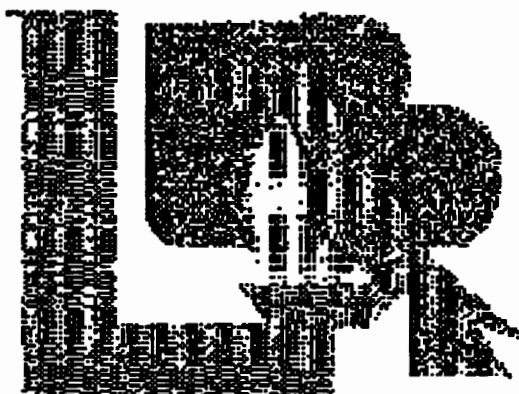
Learning With Computers: Effective Teaching Strategies is filled with suggestions to help students become independent thinkers and productive members of learning teams. Excellent for university classes and as a resource for K-12 teachers. Drawing on cognitive theory and research, Ken Ryba and Bill Anderson 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.





Colored birds
birds without colors
do not make any harm to them
because they feel pain.

They are so sweet and innocent
and we must love them
they are so strong as they fly
and we must not kill them.

They are so beautiful
very nice too,
and they look like
rose petals.

They come in many colors
and one bird is the "viudilla"
pink, blue, red,
white and yellow.

We must take care of them,
because they don't bother us,
and we must feed them
so that they would not exterminate.



Aves coloradas
aves sin color
no las maltratemos
porque sienten dolor.

Son tan dulces e inocentes
y debemos de amarlas
son tan fuertes al volar
y no debemos de matarlas.

Son tan bellas,
también hermosas,
y parecen
pétalos de rosa.

Hay de muchos colores,
y una ave es la viudilla,
rosadas, azules, rojas,
blancas y amarillas.

Debemos cuidarlas,
porque no nos hacen nada,
y debemos alimentarlas



Reports From Costa Rica, Venezuela, Brazil, Japan and Israel

by Dennis Harper

Logo Exchange Continental Editors

Africa	Asia	Australia	Europe	Latin America
Fatimata Seye Sylla	Marie Tada	Anne McDougall	Harry Pinxteren	José Valente
UNESCO/BREDA	St. Mary's Int. Sch.	Monash Univ.	Logo Centrum Nederland	NIED
BP 3311 Dakar	6-19 Seta 1-Chome	6 Riverside Dr.	P.O. Box 1408	UNICAMP
Senegal, West Africa	Setagaya-Ku	East Kew 3120	BK Nijmegen 6501	13082 Campinas
	Tokyo 158, Japan	Victoria, Australia	Netherlands	São Paulo, Brazil

This international Logo column goes to Costa Rica, Venezuela, Brazil, Japan, and Israel. José Valente writes about the Latin American countries, Marie Tada brings us up to date on Logo in Japan, and Anne McDougall reports on the recent Computers in Education Group of Victoria's annual conference in Melbourne.

Report from Latin America

From José Valente

"Logo is dead" or "Logo is old fashioned" or "We stopped using Logo to teach computer programming" are very common statements among educators in the United States. However, Latin American Logo usage is a far different story.

In the past three months I have had the opportunity to attend several international conferences on computers in education and to run Logo workshops in several countries in the Americas. These experiences have provided me with a great opportunity to interact with people concerned with the use of computers in education. The conclusion is that in Latin America, Logo is very much alive. In these countries it is possible to see major projects concerning computers in education that are Logo-based.

This report will give an overview about what is happening in terms of Logo usage in Costa Rica, Venezuela, and Brazil and discuss possible reasons for the discrepancy between what is happening there and what is happening in the United States. The information is from my visits to 1991 conferences in Brasilia in April, Venezuela and Toronto in May, and Costa Rica in June.

Computers in Education in Costa Rica

An historical overview of the Costa Rican project about the use of computers in education was published in the October, 1988 issue of *Logo Exchange* (Vol. 7, No. 2). It described the beginning of the project, its goal, the

involvement of a strong collaboration between public and private enterprises, the training of 12 educational leaders from Costa Rica at MIT and the training of other teachers in Costa Rica by these 12 educational leaders.

This project started in 1988. It was a political platform of the ex-president, Mr. Oscar Arias, to renew the educational system in Costa Rica. The goal of the project was to "put a computer in every school" in the country. To implement this idea, the Omar Dengo Foundation was created, and it has been responsible for the development and implementation of this plan. At the beginning of the project, the Foundation invited all the major computer manufacturers of the world to present proposals for both equipment and their educational uses. The international contest was "won" by IBM, which, together with Seymour Papert, presented what was considered the best proposal. This means that Logo was the educational methodology adopted to implement computers in education in Costa Rica.

After two-and-a-half years, the project is fully in progress. Today there are 130 laboratories (with 20 computers each) distributed in practically all the national territories. These laboratories are used by 115,500 students (about 30% of the elementary and middle school population). They use Logo to develop activities related to the language, mathematics, and science curricula. The students work in pairs for the equivalent of two regular classes per week.

Computers in Education in Venezuela

The goal of the current federal administration in Venezuela is also to place a computer in every school. The ministry of education has a plan to start with a few centers strategically located in the country and to disseminate the computers gradually to each school.

However, there are already two major projects, "Vision" and "Tricolor," fully in development in Venezuela. These programs provide computer experience to students from the schools sponsored by the oil indus-



try. These schools were established to provide education to the children of people working in oil fields, which in general are located in regions distant from the urban centers.

These two projects were started in 1990 as a collaboration between branches of *Petróleos de Venezuela* and *IBM*. The methodology used in "Vision" and "Tricolor" was inspired by the *Costa Rica Computers in Education Project*. Logo is used to integrate computers in curriculum subjects related to language, mathematics, and science. *IBM's* role is to provide the methodological support, to train teachers, and to provide assistance to guarantee the quality and continuation of the projects. Today, there are 20 schools participating in these two projects—which means that another 14,000 students are using Logo.

It is too early to predict whether Venezuela will implement computers in education through a model similar to the one implemented in Costa Rica. However, there is already know-how within the country to show that the experiences of students from the oil industry schools could certainly be extended to public school students. We have to wait and see which path the federal government project will take.

Computers in Education in Brazil

An overview of the Brazilian plan for implementing computers in education was described in the September 1988 issue of *Logo Exchange* (Vol. 7, No. 1). In short, the plan started in 1983 with five research centers established by the federal government, and in 1987 a project was initiated to set up Centers of Informatics in Education (CIEd) in each of the 26 states of Brazil.

The creation of CIEds throughout Brazil is today the main objective of the computer and education sector of the Ministry of Education. The CIEd is an attempt to gradually introduce the computer into the public education system. Brazil has 600,000 schools, 1.2 million teachers, and 32 million students. Instead of placing one computer in every public school, the government's goal is to establish a computer center in every city, along the model of the local library. The function of these centers is to serve students and teachers, to train teachers, and to disseminate throughout the community the different usages of computers.

The first CIEd was established by the end of 1988, and today 18 CIEds are functioning. Nationwide, about 1,000 teachers have been trained in these centers and 10,000 students have used these facilities to develop education-related activities. About 80% of the activities happening in these centers is Logo-based, with the remaining 20% constituted primarily of word processing and databases.

With the large demand for the implementation of computers in education in Brazil, the federal govern-

ment has revised the original plan. Beginning this year, the federal government has issued the National Program for Informatics in Education (PRONINFE), which extends the research projects in informatics in education to all the universities interested in this area and to major cities interested in having a CIEd. The goal is to create 550 CIEds by the end of 1993.

The purpose of introducing computers in education is not simply to be up to date with the "new technologies." If it were, this project would be far too costly. The government would like the computer to be the catalyst of a change in the educational approach currently used. The new approach would be one that promotes learning instead of teaching, that puts the control of the learning process in the hands of the students, and that helps the teacher to understand that education is not only the transference of knowledge but also a slow process of constructing it.

Thus, one of the computer-in-education approaches most prevalent in Brazil has been Logo, as opposed to computer assisted instruction. The Logo approach is supportive of the goals established by the Ministry.

Another major project concerning the implementation of computers in education is happening at the municipal level. This project was issued by Paulo Freire, the Secretary of Education in the city of São Paulo. In this educational system, there are 576 schools (356 at the K-8 level). The implementation of computers in this education project started last year. About 60 elementary and middle schools will have computers in education as part of the school activities by the end of 1991. The methodology being used is totally based upon Logo. The full details about this project will be described in this column in a future issue of *LX*.

Discussion

The summaries above describe only the countries I recently visited. However, there are other projects in Latin American countries, such as Chile and Mexico, that use Logo as the main methodology. Thus, it is not true that "Logo is dead" or "Logo is old fashioned," as I heard from several participants of the "Eighth International Conference on Technology and Education" in Toronto last May.

In fact, it surprised me to see the lack of interest in Logo by the authors and participants of that conference. This is one of the largest conferences on computers in education. There are 336 articles published in the conference proceedings, and the classification of these articles according to countries and Logo-related activities is shown below.

	No. of articles published	No. of Logo-related articles
U.S.	177	6
Canada	73	6
Europe and Asia	67	2
Latin America	19	9

(Editor's Note: NECC '91 in Phoenix had 270 papers in its proceedings, 19 of which emphasized Logo, 29 multimedia, 35 *HyperCard*, and 42 telecommunications.)

It could be that Logo is still being used extensively in the United States and Canada but its users do not bother to publish their work or choose to participate in conferences. However, it seems that major projects in the U.S. deliberately decided to stop using Logo. For example, the associate director of the California Technology Project (which has a budget of \$500 million per year to buy hardware and software for the state of California educational system) has stated that the project has stopped using Logo to teach computer programming. Logo is only used to develop Logo/Logo-related activities.

Is the impression that Logo use is declining in the U.S. a correct one? If so, why is this happening? One possible explanation is that Latin American countries are behind the times. The Logo momentum that occurred in the U.S. 10 years ago is just now hitting Latin America, and, as with any other "fashion," this educational movement in Latin America may follow the swinging-pendulum theory. If this is the case, Logo will probably have the same fate in Latin America as in the U.S.

However, there is a major difference between the educational system in the U.S. and in the majority of Latin American countries. In the U.S., the system is controlled mainly by the local community; in Latin American countries, it is controlled by the state or federal government. This has much to do with the distribution of money to buy computers and software. Since there is almost no money available for purchasing software, there are two solutions to the usage of computers in education. First, each school or pool of schools can develop its own educational software. This is what some private schools are doing. This is necessary since the market for educational software is small and does not give much incentive to software houses to produce educational software; therefore, the software options on the commercial market are very limited. However, the cost of this solution is extremely high.

The second solution is to purchase several multi-purpose software packages. This permits the development of computer activities in several domains and allows students from different grades and with differ-

ent capabilities to participate in the process of educational computing. This is the trend of computers in education in Latin American countries. Logo fulfills this requirement.

Thus, it seems that the economic situation in most Latin American countries has caused educational systems to keep using Logo. This can be a good example in which a little bit less can mean more. However, I am not advocating the "misery path" as a way of solving the problems of using computers in education (although the governments would be extremely pleased with this type of solution). We still have a long way to go. First, we have to look at what has happened in countries such as the U.S. to find out the reasons Logo has become old-fashioned. It could be for educational reasons, and/or it could be due to the fact that Logo did not keep up with the development of computers-in-education technology as other alternatives became more interesting. Thus, the work to do is to add to Logo whatever it is missing. Second, we have to keep track of our own experiences in order to find out whether by sticking to Logo we are getting what we think is important educationally. If not, we should look over the fence and make the move to other ways of using computers in education. In this case we had better swing the pendulum fast!

Report from Japan

From Marie Tada

Just about a year has gone by since my last report to *Logo Exchange* from Asia, so here is a summary of recent Logo events in Japan. First, I was invited to attend the Conference on Computers and Mathematics Education, which took place in the spring of 1991. One heavily attended talk was entitled "Another Side of Logo." Mr. Shoda, a high school mathematics teacher, had a good number of examples, available on individual disks, of procedures that demonstrated mathematics and programming principles. He gave numerous reasons why *Logo Writer* may be considered a good vehicle for discovery and exploration in the area of mathematics. I thought this talk was significant from the fact that it was part of the conference at all and that so many high school teachers were interested in learning more about using Logo in their classes.

When one visits a Japanese school, one cannot help but feel that the Japanese have been slow to introduce computer related technology into the educational scene. The Japanese Ministry of Education has made great strides in recent years in planning for the introduction of technology into the schools. However, a lot of problems have stemmed from the difficulty of adapting the computer (especially the word processor) to the intricacies of the Japanese language, with its thousands of Chinese characters (Kanji) and two phonetic syllabaries



of 50 characters each. It is no wonder that the fax machine was taken most seriously here in Japan from the start as a way of sending hand-written messages without having to type out all those characters first! Japanese-language word processors have become easier to use in recent years and the boom in laptop computers with Japanese brand names is testimony to the popularity of computer use in modern Japanese life.

In education, things tend to move conservatively in any society and perhaps even more so in Japan. However, from early on, Logo has been a part of the planned computer education curriculum. According to a report presented by Professor Haruo Nishinosono at the WCCE '90 conference in Sydney, the Center for Educational Computing (CEC) was established in 1986 in Japan as a joint project of the Ministry of Trade and Industry (MITI) and the Ministry of Education, Science, and Culture. After much discussion, a unified set of specifications for the use of BASIC and Logo was established in February 1990, along with some agreements on data formats that would make it easier for software and hardware developers to exchange ideas and work within some guidelines. Ultimately, this led to more computer related materials being made available to the schools.

I visited a model school for technology use in Chiba prefecture and was impressed with the commitment that the school had made to making Logo a significant part of its computer literacy curriculum. The school administration and teachers were very interested in CAI and the creation of computer guided lessons using authoring tools. Such work was necessary due to the lack of commercially available software in the Japanese language. However, the teachers found that the children were not overly enthusiastic about the CAI lessons but were very receptive to the use of *LogoWriter*, with its colorful graphics and the potential for individual exploration. The school was very interested in having the teachers and students become more proficient in the use of Logo in the learning process in the years to come. One of the biggest problems they were confronting was the lack of preparation on the uses of technology in the teacher education programs of Japan. Also, the fact that Japanese teachers are expected to move to different schools every few years took away a core group of technology-using teachers just when things were beginning to "gel." In some cases, teachers became threatened by the speed with which the children would go beyond their teachers. In the strict hierarchical order of Japan, this can create a sensitive situation.

After the Conference on Computers and Mathematics Education, I was fortunate to be able to discuss new developments on the Japanese Logo scene with Ms. Kimura and Mr. Yokoyama at Logo Japan. I found out that a new version of Japanese *LogoWriter*, *LogoWriter*

II, would soon be available and that it would meet the above-mentioned CEC standards. Although the program is now only available for NEC computers, it is hoped that an IBM version will be available by the fall. Logo Japan is also making available a Japanese version of *Lego/Logo* called *Lego.Lex*. The *LogoWriter II* version has been written with the idea that the program will also be used at the junior and senior high school level in Japan. In order to find out what teachers at the various levels need or want, Logo Japan has been hosting workshops that allow teachers to give feedback about programs under development and to suggest directions Logo should take.

Mr. Yokoyama, head of marketing at Logo Japan, explained that Logo use in Japan is presently divided along the following lines: 60% elementary schools, 30% junior high schools, 5% high schools/specialty schools, and 5% teacher education institutions and other institutions of learning. The elementary school technology-related curriculum in Japan should be fully implemented in 1992, the lower secondary in 1993, and the upper secondary in 1994. In information-processing classes, students are to learn about computer structures, programming, applications, and computers in society. In home technology and other vocational classes, computer studies are also included and Logo can be selected as a part of the curriculum. According to Mr. Yokoyama, 30,000 schools are now using Logo from the northernmost island of Hokkaido to the southern island of Okinawa.

A very impressive part of the *LogoWriter II* package lies in the fact that one can use it totally in Japanese, totally in English, or in a mixed Japanese/English mode. I couldn't help but think that the "playing-with-learning" philosophy can in this way be extended even further—into the area of foreign language instruction.

More emphasis has been put on Japanese word processing in the new version. A command of more than 2,800 Chinese characters is needed for basic literacy in Japan (despite this, Japan has a literacy rate of over 99%). Thus, to avoid frustration, dictionaries are sold according to grade level so that the user will not have to confront characters that have not yet been studied. *LogoWriter II* allows printing in 80 columns in black-and-white or in color. One can use SETFORMAT to set the format for page length, margins, and so forth.

Ms. Kimura gave me a guided tour of some of the features of *LogoWriter II*. Some of the "new" features I recognize and welcome from "older" versions of Logo are FULLSCREEN and WINDOW. A great new feature is the ability to mix colors with 4,096 possible color variations. For example, color 2 can be changed by setting the intensity of red, green, and blue:

SETPALETTE 2[5 3 4]

I find that the SHAPES page has enormous appeal to students. Each page in *LogoWriter II* has a SHAPES page attached with 150 possible shapes. Each shape is made up of 32 by 32 dots, which gives a much finer resolution. Using the mouse, the shape can be reversed or turned. BIGSTAMP is a command to resize the shape and give it a designated width and height. Sixteen colors can be used within the shape, and a variable colored pixel can be assigned using *"*"*. One can save a shape to the shape editor with the SNAP command, and SNAPCOLOR takes the shape to the editor along with its color! Another nice feature is the ability to print out the SHAPES page.

The command TURTLEMOUSE allows one to manipulate the turtle with the mouse. By clicking on PENDOWN, the mouse can be moved to draw free-hand. A click will put the pen up and, once the pen is inside the shape, a double click will put the pen down and fill the shape with color.

The mouse can also be programmed

```
WHENCLICK 1 [SETSH SHAPE + 1]
```

which will increase the number of the shape for the turtle under command by one. WHENCLICK takes a number from one to four. WHENCLICK 1 is for a left click, 2 for a left double click, 3 for a right click, and 4 for a right double click.

There are 32 turtles (0 to 31) available, but it would appear at first that there are only four. The others are under these four turtles and can be moved independently. There is also a TurtleNumber 32, which is called the "special turtle." It is a square by default, but the shape can be changed by snapping the graphic underneath. The user can type

```
TELL 32  
ST
```

and set the size of the square to go around a graphic (SETSIZE [120 120] -120 dots by 120 dots). The user can then type SNAP to make the turtle number 32 the shape that is beneath it. Then, if

```
FORWARD 100
```

is typed, turtle number 32 will move the shape beneath it forward 100 turtle steps.

Labels (e.g., the word "hello") can be held in the clipboard and pasted to a shape and then moved together as a unit. The command

```
WHENBUTTON 2[PRINT [HELLO!]]
```

will cause the word "Hello" to be printed on the screen when you click on turtle number 2.

Icons at the bottom of the screen can also be programmed. For example, the command

```
SETICON 1 "line [FD 100]
```

will result in the turtle moving forward 100 when the icon line at the bottom of the screen is clicked on. In the same way,

```
SETICON 2 "squ PD [REPEAT 4[FORWARD  
100 RIGHT 90]]
```

will result in a square being made when the second icon, *squ*, is clicked on. Up to 10 icons can be programmed in this way.

To help teachers use *LogoWriter II* in the mathematics curriculum, some built-in features are available. These include natural logarithms, exponents, pi, number formats, square roots, and more. For example,

```
POWER 2 4
```

will get 2 raised to the fourth power.

I was excited about new list-processing commands that could make the creation of games much easier. ADDITEM allows one to add items to a list in a certain position; DELITEM allows you to delete a certain item from a list. For example, the rank of an item can be found by typing

```
PRINT RANK Z :A
```

This lets the user know where the item Z is in the list. This is the reverse of ITEM, which allows you to indicate a member of the list to act upon. With RANK, you can find out where the item is so that you can act upon it.

That fact that Logo is a most suitable program for integrating the computer into the curriculum is becoming much more accepted in Japanese education. The fact that the Japanese Ministry of Education is behind the introduction of Logo into the schools means that the future of Logo in Japan should be bright indeed.

Report from Australia

From Anne McDougall

Logo work being carried out in the state of Victoria was highlighted at this year's annual conference of the Computers in Education Group of Victoria, which was held in Melbourne in July.



- Bob Aikenhead's workshop entitled "Using Logo as a Problem-Solving Tool" was very well received.
- Meenah Mattison presented an interesting session entitled "The Theory Behind the Use of *Lego TC Logo* in the Primary Classroom."
- John Turner presented a fascinating new approach to teaching recursion in his session called "Logo Recursion: Control Before Understanding."
- David Rasmussen showed some very useful ideas in his session entitled "Getting the Most From *LogoWriter*."
- John deFigueiredo extended the *LogoWriter* word-processing facilities with procedures for justification and centering in a session called "Writing Text- Handling Procedures in *LogoWriter*."
- Jeff Richardson provoked some lively discussion in his workshop "Programming the Macintosh With *LogoWriter* (or Why *HyperCard* is NO GOOD)."

I believe we are seeing the impact of Logo ideas in the theory and practice of educational computing much more strongly than we are seeing it in work directly using Logo. Both the opening and closing keynote speakers at the conference were educators widely known for their Logo work, yet neither spoke directly about Logo—although the "Logo thinking" was very evident in the arguments they used. Carolyn Dowling (coauthor of *Lets Talk Turtle* and *How Turtles Talk*) spoke to the conference theme, "Computers—Contributing to Chaos or Change?" and Gary Stager (former president of ISTE's SIGLogo and regular *Logo Exchange* contributor) closed the conference with a talk entitled "Computers and Learning Versus the 1990s." These two excellent addresses, although totally different in style, had many common themes.

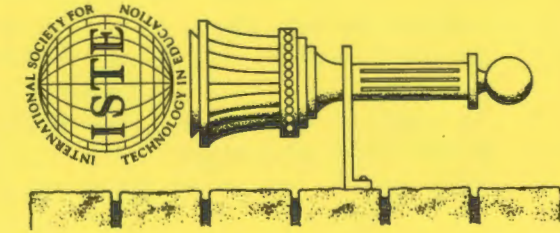
Carolyn expressed concern that the computer was no longer contributing to either chaos or change in education because computers were being used in very conservative ways in classroom practice, often imitating business or commercial uses. For example, word processing, spreadsheets, desktop publishing, and other uses are becoming the stuff of schools' computing practice. She argued that educators should think carefully about possible unexpected outcomes from the uses of computers for writing and from innovations such as laptop computers for students. She also called

for a return to the "chaos" of early computing work in schools, where teachers were trying new ideas and at the same time reflecting energetically about the potential for these innovations to strengthen their students' learning. Gary declared that "curriculum is the enemy," issued strong warnings about the extensively available "software du jour," and argued strongly that educational uses of computers should be as vehicles for self-expression and communication.

The proceedings of the conference, *Computers—Contributing to Chaos or Change?*, can be obtained from the Computers in Education Group of Victoria, P.O. Box 1061, Richmond North, Victoria 3121, Australia, at a cost of \$25 Australian (including postage and handling).



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**