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



LOGO EXCHANGE

JANUARY 1988

VOLUME 6 NUMBER 4

The winter at Valley Forge was one of the worst times during the war. Food, clothing, and other supplies did not reach the camp. This was the reason why many men died of illness.



International Council for Computers in Education



Publications

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

Edited by
Theodore C. Burrowes and Sharon K. Burrowes

An ICCE Publication

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

The ECCO Logo Project
Includes:

***Student worksheets
Teacher information sheets
Teacher-training materials
Logo II version of all materials
Apple Logo version of teacher materials***

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

To order your copy, use the order form below or call ICCE at 503/686-4414.

Name _____
Address _____
City/State _____
Postal Code/Country _____
Phone _____

Payment enclosed
 Bill me (add \$2.50 handling)
 Bill my:
 Mastercard Visa (circle one)
 card number: _____
 name: _____
 exp. date: _____

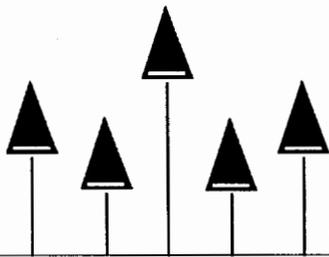
Please send me ____ copies
of *The ECCO Logo Project*
at \$20 each. Total price \$ _____

Add \$2.50 for shipping \$ _____

Add \$2.50 handling
for billed orders \$ _____

Total Amount \$ _____

Mail to: ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403 USA



LOGO EXCHANGE

VOLUME 6 NUMBER 5

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

JANUARY 1988

Founding Editor

Tom Lough

Editor-In-Chief

Sharon Burrowes Yoder

International Editor

Dennis Harper

Senior Contributing Editor

Robs Muir

Field Editors

Eduardo Cavallo

Patricia Dowling

Anne McDougall

Harry Pinxteren

Fatimata Seye Sylla

Hillel Weintraub

Contributing Editors

ASTROLUG

Eadie Adamson

Gina Bull

Glen Bull

Doug Clements

Bill Craig

Sandy Dawson

Judi Harris

Barbara Randolph

Linda Sherman

Gary Stager

Managing Editor

Anita Best

Special Interest Group Coordinator

Keith Wetzel

Advertising Director

Kathleen Geygan

SIGLogo Board of Directors

Peter Rawitsch, President

Gary Stager, Vice-President

Ted Norton, Communications

Publisher

International Council for
Computers in Education

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

POSTMASTER: Send address changes to Logo Exchange, Uofo, 1787 Agate St., Eugene, OR 97403.

CONTENTS

From the Editor		
	Sharon Burrowes Yoder	2
Monthly Musings		
	Tom Lough	3
Logo Ideas — <i>Playing with Polygons</i>		
	Eadie Adamson	4
Testudial Testimony — <i>Research on Variables, Algebra, and Logo: Part I</i>		
	Doug Clements	8
PenPoints — <i>Logo Book Reviews</i>		
	Sarah Joyce	11
LogoPals — <i>MIX, STIX, and Logo Panpals</i>		
	Barbara Randolph	12
Meet the SIGLogo Board		13
InLXual Challenges — <i>Worms as Algorithms</i>		
	Robs Muir	14
Teaching Tools — <i>Frog Algebra</i>		
	Glen Bull and Gina Bull	17
Stager's Stuff — <i>Playing with Music Part 2: Child Composers</i>		
	Gary S. Stager	22
Logo LinX — <i>'sNo Mystery</i>		
	Judi Harris	25
LXlonary		
	Bill Craig	28
Global News		
	Dennis Harper	30

SIGLogo Membership (includes *The Logo Exchange*)

	U.S.	NON-U.S.
ICCE Member	24.95	29.95
Non-ICCE Member	29.95	34.95
ICCE Membership (includes <i>The Computing Teacher</i>)	28.50	31.50

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

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

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

From the Editor

This issue brings you the first news from the SIGLogo officers. Recall that *LX* is now the journal of the ICCE SIGLogo and membership in this SIG carries with it a subscription to the *Logo Exchange*. Your temporary officers have met and drawn up ByLaws which will be published in a future issue. Take a moment to read the brief biography of each officer — I'm sure you will feel that they are excellent representatives of the Logo community. As indicated in the "Meet the Officers" article, the Bylaws will be voted on at NECC this summer and permanent officers will be elected. If you are interested in becoming an officer, please let one of the temporary officers or myself know — you don't have to be at NECC to run for office. Your participation can only strengthen the organization.

How about Logo activities in your local area? Is there a local Logo user's group? What are they doing? We'd love to publish news of meetings, conferences, or other activities so that the community of Logo users world-wide can become better acquainted. The International Column provides us with windows on what is going on world-wide — how about some news from the United States and Canada? Your comments don't have to be in polished form — just send the news to me and I'll collect them in a SIGLogo column in a future issue.

Let's make this new year (this IS the January issue!) a year that is rich in Logo activities. For myself, I would hope to see Logo broaden it's scope to a wider audience. I would like to see young people of every age using Logo as an integral part of many subject areas. I would like to see more teachers empowered and excited by the potential of the Logo environment. I would like to see universities find Logo an acceptable language for introducing computing to adults, both young and old. I would like to see the scope of Logo broaden through such vehicles as Lego/Logo or music Logos.

What are your wishes for this new year with Logo? What can you do to make those dreams come true. Join with us to make SIGLogo a vehicle for communication and change throughout the Logo community.

Sharon Burrowes Yoder

Cover: This month's cover is again part of a 12-event animated timeline of the Revolutionary War done by fifth graders at the Lafayette School in Wayne, NJ. Steven Petreoski, Andrew Rubin, Sean Vij, and Jason Meyer have documented the conditions at Valley Forge. Arlene Schoenberger was their teacher and Gary Stager their Logo consultant.

Monthly Musings

by Tom Lough

Incidental Learning

A couple of years ago, when our son, Kyser, was in his "terrible two's," I firmly decided that I would never ever take him to a toy store. How could I possibly expect him to deal with the overwhelming array of attractions? How could I resist wanting to buy him [and me!] one of everything?

As happens to most parents, I have experienced a mellowing of my position since then. As it turns out, I really enjoy taking him on an outing to our local toy store now that he is four years old. He is able to deal very well with the flood of toys. [I still have my share of trouble, though!]

During the days of Christmas shopping, we went to the toy store for a final reconnaissance, to make sure that we knew exactly what to ask ole Santa for. Kyser asked to be placed in the shopping cart, and I happily obliged. [He is much easier to manage when in the cart rather than when "on the loose."] At that moment, I realized that I had an opportunity to facilitate some special learning.

I held the cart still until he discovered that he would have to tell me to move. He soon learned that I responded only to a limited set of commands. Yep, you guessed it. Forward, back, right, and left.

Can you imagine the scene? The four-year-old lad bellowing out sequences of FORWARD FORWARD FORWARD RIGHT RIGHT FORWARD FORWARD FORWARD STOP and the father bending to obey, propelling the shopping cart into the next aisle...

Lots of things happened during that shopping trip. Kyser reinforced his concepts of the left and right directions [they are still a little "shaky"]. He realized that the RIGHT and LEFT commands resulted in a turning of his shopping cart, with no other motion. He learned how to use combinations of FORWARD or BACK and RIGHT or LEFT to get to the next toy display. In other words, he was immersed in a personal world of FORWARD, BACK, RIGHT, and

LEFT and he used the commands to accomplish personal goals.

Something happened to me also. I realized

once again the importance of providing opportunities for children to set and reach meaningful goals, and of always being alert for learning opportunities in daily life.

When we returned home that afternoon, he asked me on his own to set up Logo on the computer so he could practice.

And, oh yes, I bought him something at the toy store . . . a turtle, natch!

FD 100!

Tom Lough
Box 5341
Charlottesville, VA 22905

"Shopping is a feeling."

—True Stories

Logo Ideas

Playing with Polygons

by Eadie Adamson

Last year we were all so excited about exploring LogoWriter that we neglected some simple but classic Logo problems, other than that of drawing squares and rectangles to make cities. This year I decided that I wanted my students to have the problem-solving experience of discovering polygons and then making a tool page of shape procedures to use in other projects I had in mind. (More about those projects in a later column.)

Following Polya's steps to problem solving (mathematician George Polya's *How to Solve It*, referred to in Papert's *Mindstorms*), I introduced the idea of solving a problem by looking at what you know, where you might have seen something like it before and how it is like the new problem, thus narrowing down what you need to know. This kind of thinking is central to solving many Logo problems. I wanted my students to start seeing the possibilities of building upon what they already knew as they worked on their programming this year.

First I drew a square and an equilateral triangle. Then I asked my students to tell me what makes a square. Next we considered in what ways an equilateral triangle is like a square and in what ways it is different.

My next question was: Keeping in mind the idea that we have seen something like it before, how much do we already *know* about how to make an equilateral triangle in Logo? (We can determine the length of a side, since all are equal. We know the number of sides.) What do we need to find out in order to make a perfect equilateral triangle? (We need to find the angle.)

I suggested a way to use the computer to solve the problem. We had determined that the angle was what we needed to find, the rest we knew. We decided to solve the triangle problem as a group, using our demonstration monitor. I suggested starting by working in the Command Center with a line like this (waiting to write a procedure until we were sure of the angle):

```
CG REPEAT 3 [FORWARD 50 RIGHT 90]
```

Using the turn for a square helped to illustrate that what is correct for a square is too small an angle for a triangle. Next we tried a smaller turn than 90, then a larger turn, so we could see the result. I suggested we check what would happen with a turn of 180. By now most of the students had concluded the number we were looking for would fall between 90 and 180. With LogoWriter we could delete the number just used, change the

angle, and press Return. The old shape would be erased and the new one drawn. (Other versions of Logo have different ways of repeating a line so that it is unnecessary to type it repeatedly.)

With the turtle hidden, it is easier to see if the angle chosen is correct. The shape should close completely, especially if you want to be able to fill it later. Sometimes it is helpful to point out (or to encourage someone else to observe) that too little turn will not close. Although sometimes creating wonderful shapes, too much turn will cause the final line to cross over the first line, provided turns are no more than 180.

We tried another way to check if the shape was perfect. First we put the turtle in the center of the screen using the Home command. Then we typed SHOW POS. A list [0 0], the coordinate position of the turtle, was reported to us (SHOW is a command in LogoWriter that puts information in the Command Center.). After trying a new angle for a triangle, typing SHOW POS helped us to be sure that the turtle had returned to where it started. If, after typing SHOW POS, the report is not [0 0] it was clear that the angle was not correct, and we needed to try again.

I had to get my students comfortable with the idea of using SHOW POS to get a report of the turtle's position. I gave them just a simple explanation of the first number indicating position in left-right direction, and the second meaning up-down.

Once the perfect angle was found, we wrote a procedure with a variable size with which to experiment. I allowed time to explore designs before challenging my students to find the other shapes.

Later we would use SHOW POS after using LogoWriter's turtle-move keys to position the turtle. Using SHOW POS you can find a starting position for the turtle to begin drawing. First place the turtle, then type

```
SHOW POS
```

You will see a list representing the current turtle position in the Command Center. Then type

```
PU SETPOS [ the list reported by SHOW POS goes here ] PD
```

before drawing a shape. (This also a neat trick to use with the LogoWriter Flashing Sign idea when you want more than one sign to flash at a time. Can you figure out how to use SETPOS to do this?)

We used a polygon worksheet (see the end of this article),

Logo Ideas

first to find a pentagon and a hexagon, then to find all the regular polygons up to a 20 or 36-sided figure. Although I'm not overly fond of using worksheets, we needed to keep track of what we were finding, and I also wanted students to be able to draw some conclusions from their results. Additionally, they could paste these sheets into their notebooks (we store our disks in a hard-backed composition book, keeping notes and disk together).

This worksheet provides a good opportunity to assign a little research into the naming of polygons, since all the shapes on the worksheet are not named. See how many notice the "rule of 360" hinted at by the "x" and "=" on each line of the worksheet. I also asked my students, after they had found a number of shapes, to try to come to some conclusions about the sets of numbers they had collected. Is there a relationship between the repeats (number of sides) and the angles? (There is an inverse relationship: angles get smaller as the number of sides increase.)

Each student may want to find all the shapes on their own. One of my fourth grade students was obsessively determined to find every single shape on the worksheet and spent extra time in the computer room working on the project. In fact, it was difficult to get him write procedures until he had found *every* polygon.

If you have another agenda, though, and don't want to spend an enormous time on what eventually becomes a somewhat repetitive task, you might want to assign shapes to different teams. Each team could have the task of finding several and reporting to the group. The reporting task should include discussion of how to solve the problem: who just guessed and was right, who tried a number and had to narrow it down, who found another shape by accident.

Some shapes which you might think would be perfect, such as a nonagon, turn out to be less than so. LogoWriter rounds to four decimal places, so occasionally you cannot get a perfect [0 0] position. I showed my students how to watch the numbers SHOW POS reported, and how to narrow down their guess to as near [0 0] as possible. We used a general rule of thumb: the second number is less than zero if the turtle has not yet neared the center, more than zero if the turn has been too large.

Some of my students began experimenting with decimal places to find the perfect shape. Eventually we used SHOW 360/13 (or whatever sided figure we needed) to find the angle ahead of time. When even that wasn't perfect, some students made excursions into added decimal places, a nice opportunity to talk about decimals outside of math class!

This project prompted some students to "invent" or almost "invent" the all-purpose "poly" procedure:

```
TO POLY :SIDES :SIZE
  REPEAT :SIDES [FORWARD :SIZE ->
                 RIGHT 360/:SIDES]
END
```

You may or may not want to take this digression, although I find that for some age groups it makes a nice kind of summation for the polygon project.

From here the exercise may take its own turn, depending upon the interest and ability level of your students. Incrementing (adding to) variables in a recursive procedure allows wonderful growing shapes to be drawn rapidly. It might be a nice next step to show your students if they begin writing TRI 5, TRI 10, TRI 30, and so on. For example, a "new" way to do just that for them is:

```
TO TRI :SIZE
  REPEAT 3 [FORWARD :SIZE RIGHT 90]
  TRI :SIZE + 5
END
```

Use of negative numbers can be introduced to change direction. Stop rules might be needed next. Islamic design or tessellation could be included as well.

For the POLY-WISE:

If you're working with LogoWriter, try having all four turtles draw mirror images of the designs. If there is a procedure for left and a procedure for right for a given shape, which turtles need negative inputs for the angle to create a four-directional design? If you have only a procedure which turns right, can you still have the four turtles draw in mirror image? I had fun playing with this idea myself. Here's what I came up with, based on a POLYR and POLYL pair of procedures:

```
TO POLYR :SIDES :SIZE
  REPEAT :SIDES [FORWARD :SIZE ->
                 RIGHT 360 / :SIDES]
END
```

```
TO POLYL :SIDES :SIZE
  REPEAT :SIDES [FORWARD :SIZE ->
                 LEFT 360 / :SIDES]
END
```

Now get all four turtles to work!

Logo Ideas

```

TO FOUR :SIDES :SIZE
  TELL 0
    POLYR :SIDES :SIZE
  TELL 1
    POLYL :SIDES (- :SIZE)
  TELL 2
    POLYR :SIDES (- :SIZE)
  TELL 3
    POLYL :SIDES :SIZE
END

```

Now make it fancier. First move the turtles so the design is more centered:

```

TO START
  RG
  TELL ALL
    HT
    PU
    RIGHT 90
    FORWARD 19
    LEFT 90
    FORWARD 20
    PD
END

```

Add a little recursive trick (can you predict what this does?):

Attention SIGLogo Members

SIGLogo Meets at NECC '88

SIGLogo members will hold their annual meeting at NECC '88 in Dallas, TX, on June 15, 1988 from 7:00 - 9:00 PM. The agenda and meeting place will be announced later. Plan now to attend.

Mailbox too Full?

The sale of mailing labels brings ICCE additional income for its activities (and to pay its staff). However, if you feel your mailbox is a little too full, you may request that we remove your name from the list from which mailing labels are generated and sold. Please notify Bill Park at ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905.

```

TO POLYSPI :SIDES :SIZE
  IF :SIZE > 45 [STOP]  <-You could set a max
  FOUR :SIDES :SIZE    here with another
                        variable.
  POLYSPI :SIZES (:SIZE + 5)
  FOUR :SIDES (:SIDES - :SIZE)
END

```

Type START to position the turtles, then try POLYSPI with 6, 7, 8 or 9 sides and a starting size of 5. What do you get? Where might you add a turn to make a still more complex design? Experiment with changing colors too!

Here are some other neat polygon ideas, adapted from the new LogoWriter Primary materials:

- Make a robot construction set from polygons.
- Decide on the appropriate polygons.
- Write procedures with variables OR, if your students aren't ready for variables, write each shape in a small, medium and large size (decide together how big each will be).
- Combine the shapes on a single page.
- Name the page ROBOTTOOLS.
- Save the tools on everyone's disk.
- Now choose a new page for the robot, name it ROBOT. Type GETTOOLS "ROBOTTOOLS to load the shapes into the workspace.

Design a friendly robot! If you've used SHOW POS, try using turtle-move keys and SHOW POS, then writing PU SETPOS (etc.) to position each part. Don't forget to name the robot.

If your students need a rectangle, you might suggest piling several squares on top of one another, then filling with the same color.

Extend the idea by writing a robot story. Connect the pictures and the pages or write the story around the robot picture. What kind of robot adventure might your class think up? What about a scrambled adventure? Make the story interactive by asking the reader to make choices.

Send us your best pictures and stories. Maybe your page will end up on a *Logo Exchange* cover!

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

An ICCE Copy Me! Page

Logo Ideas

POLYGON WORKSHEET

Following the same method we used to find the angle for a triangle, find the angles for a PENTAGON (5 sides) and a HEXAGON (6 sides). Fill in the chart below but keep the repeat numbers in order (3, 4, 5, 6 and so on). Supply names for whatever shapes you can.

SHAPE:	REPEAT:	x	ANGLE:	=
<u>TRIANGLE</u>	_____	x	_____	=
<u>SQUARE</u>	_____	x	_____	=
<u>PENTAGON</u>	_____	x	_____	=
<u>HEXAGON</u>	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=
_____	_____	x	_____	=

Now experiment with these shapes. Here are some ideas:

1. Can you write a variable procedure for each? Call the variable :SIDE.
2. Can you use a pentagon and draw a flower with five petals?
3. Can you make a windmill with triangles?
4. Can you add a triangle to each side of a pentagon to make a star?
5. Use three of the above shapes to create a design.

Testudinal Testimony

Research on Variables, Algebra, and Logo: Part I

by Douglas H. Clements

To begin this column, we will present a "classic" algebra problem. Don't stop reading just because algebra isn't your "thing." Later in the article, we will see how the difficulties involved in solving algebra word problems relate to the learning to Logo. And now, the problem:

Write an equation using the variables S and P to represent this statement: "There are six times as many students as professors at this university." Use S for the number of students and P for the number of professors.

Soloway, Lochhead, and Clement (1982) found that their university students frequently erred on this problem. Most of these students wrote: $6S = 1P$ (be honest to yourself at least, did you?). The researchers claimed that throughout their schooling, students develop limited, special purpose translation algorithms (e.g., trivial word-to-symbol matching). Such algorithms work for many textbook problems, but do not involve a true understanding of algebra.

On the "students and professors" problem, students used two flawed strategies. In the first, they simply assumed that the order of the key words in the problem statement could be mapped directly into the order of symbols appearing in the algebraic equation. In the second, they confused the semantics of the equation. They interpreted the reversed equation, $6S = P$, as stating that a large group of students are associated with a small group of professors. The letter P stood for "a professor" rather than "the number of professors." The equal sign expresses a comparison or association, rather than an equivalence. Thus, to these students, the equation indicated that there were more S's than P's. Their "equation" is actually a passive picture representing the different sizes of the groups. (Davis, 1984, would say that instead of using an equation frame, or script, the students had used a label frame, as one might in the following situation: 100 cm = 1 m.)

In contrast, the correct equation, $S = 6P$, expresses an active operation being performed on one number (the number of professors) to yield another number (the number of students). It does not describe the sizes of groups literally. It describes an equivalence relation that would occur if one were to make the group of professors six times larger. In one student's words, "If you want to even out the number of students to the number of professors, you'd have to have six times as many professors" (Soloway et al., 1982, p. 175). This student viewed the equation in a procedural manner—as an instruction to *act*. In brief, the incorrect answer is a *description* of a situation, whereas the correct answer represents a *prescription* for action.

Based on this idea, the researchers asked: Would students placed in an environment which induced them to take a more active view of equations make fewer errors? A computer programming environment was the clear choice. A program is a prescription for action. Students from a computer programming course were asked to write either an equation or a computer program to represent statements similar to the "students and professors" problem. As predicted, they got significantly more problems correct when writing a program than when writing an equation. Similarly, they were more likely correctly to read a program than an equation.

Programming and the Variable Concept

Why would programming facilitate solving problems involving variables? Soloway and his colleagues suggested the following reasons:

- *Unambiguous semantics of programming language constructions.* That is, while symbols are often open to a variety of interpretations in mathematics, programming languages require that only one interpretation be associated with each symbol.
- *Explicitness required by the syntax of programming languages.* For example, the equation's "6S" could be construed as "six students." In contrast, programming's "6 * S" is more likely to mean the operation of multiplication.
- *Viewing an "equation" in a programming language as an active input/output transformation.* That is, in programming there is an input that is operated on to produce an output.
- *The practice of decomposing a problem into explicit steps.*
- *The practice of debugging programs* (equations in a mathematics course are rarely "run").

Although many programming advocates emphasize the importance of debugging, videotapes revealed that the other factors, such as the input/output idea, were more important in this case.

Other Encouraging Reports

Results similar to Soloway et al.'s have been reported, using both Logo and BASIC. Researchers have found that programming enhances the understanding of variables for students from the primary grades to high school, even in comparison to other treatments (Carmichael, Burnett, Higginson, Moore, & Pollard, 1985; Findlayson, 1984; Hart, 1982; Milner, 1973; Oprea, 1985). In one study, four average-ability fourth graders were presented with six sets of tasks during individual pre-treatment interviews (Nelson, 1986). They then used Logo to solve problems involving rectangles, formulas and equations, and number sequences. After this experience, the tasks were presented again. Two students used literal symbols in an improved manner. During the pre-Logo interviews, they had used a correspondence between the letters of the alphabet and the positive integers to assign values to literal symbols in equations (i.e., A = 1, B = 2, etc.).

RESEARCH ON VARIABLES, ALGEBRA, AND LOGO --- CONTINUED

However, after Logo, they determined each literal symbol's value correctly. All the students could use literal symbols in formulas after Logo, whereas none could before.

Varying Variables

Nelson (1986) also reported that students exhibited different perceptions of the uses of literal symbols in Logo programming, such as names for memory locations, missing numbers in equations and formulas, missing dimensions of rectangles, and recursive symbols. It may be useful to think about how the concept of variable *can* vary in the Logo environment. Sutherland (1987?) identified the following categories:

- *One variable input to a procedure* (not operated on within the procedure)
- *Variable input as scale factor*
- *More than one variable input to a procedure* (often used to avoid expressing a general relationship between variables within a procedure)
- *Variable input operated on within a procedure*
- *Variable input to a general superprocedure which calls a variable subprocedure*
- *Variable within an assignment statement*
- *Variable input to define a mathematical function in Logo*
- *Variable in a conditional expression*

Varying Contexts

A related question is: Will students be able to use the variable idea if the *context* varies? That is, will they be able to generalize the idea as used in Logo to other situations?

After providing Logo experiences, Lehrer and Smith (1986) asked third grade students how to modify a procedure which drew a square with sides of a constant length so that it could take a variable input. Almost all the students (96%) correctly modified the procedure. Then they were asked what a variable was. The responses of 38% were tied to a specific procedure; 29% were tied to Logo (but not a single procedure); 13% were general ("it is something that varies, like a name, number, anything"); and 29% made no response. Thus, most of these third graders' conceptualizations were bound to the Logo context, with little generalization.

Noss (1985) studied the learning of children in grades 3, 4, and 5 for 18 months. Exploratory interviews were conducted to answer two questions: How many children use the Logo ideas of naming and inputs to facilitate the conceptualization and symbolization of the concept of algebraic variable? In what ways are children able to use their Logo-based experience to assist the process of formalization?

Questions were posed so as to encourage children to construct a formalism, rather than interpret existing symbols. For

example, one item stated, "Peter has some marbles. Jane has some marbles. What could you write for the number of marbles Peter and Jane have altogether?" Another showed a square and asked, "What could you write for the distance all around it."

Six of eight children interviewed were able to suggest names for the unknowns. Interesting, the two who were not able to do so had not used variables in the context of Logo to any extent. One, when prompted to think of Logo, was able to say only, "We make up procedure names." Thus, it may be that any conceptual connections was based on the idea of naming inputs to procedures, rather than the idea of naming per se. Children had more difficulty with the marble item than items involving length or other continuous quantities, possibly because their use of variables in Logo rarely involved discrete quantities.

One limitation is that children were interviewed by the person who always came to the school to work with them on Logo and that the task was posed as one of formulating rules for other children who had learned Logo. Nevertheless, only one child began work by using the "dots" (colon) convention for naming variables. Three others adopted it while working. Nicola apparently began answering the marble question without considering Logo, but used her Logo knowledge as an aide to expression:

Nicola: You could use the input again (she had not previously referred to inputs).

Interviewer: Alright, show me how.

N: (writes) :Peter + :Jane = all the marbles

I: Can you read it out?

N: Peter plus Jane equals all the marbles. You use those two as the inputs, with as many marbles as you want to.

I: So what are the dots in front of Peter and Jane?

N: They're to represent that it's an input.

I: But this isn't a Logo program is it?

N: I know, but if it was...just to say that it's an input.

I: So what does the input actually mean there then?

N: That you can type in however size you want it or how many you want it. How ever many they want. How many they want Peter to have, and how many they want Jane to have.

The interviews provided evidence that children made such connections between Logo work and formalizations of rules, and that children constructed names that represented a range of numbers (rather than a single, specific number, as is the natural tendency of children). For instance, Julie proposed that she "make a word for" certain numbers in the problems, even though "You don't know what they [the numbers] are." It appears that children used the metaphor of typing in a value at the keyboard to help them conceptualize a range of numbers, while only necessarily considering one at a time. Again, the two children who had not used Logo variables extensively where the two most likely to demonstrate a "specific number" misconception. One, for example, made up labels, but seemed confused by the idea that the value would vary. He resolved his problem by assigning

RESEARCH ON VARIABLES, ALGEBRA, AND LOGO — CONTINUED

specific numbers; he wrote

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10

and said, "Now... we can have a certain number up to... we can have any number up to... we can go up to 26."

Some students began with Logo-influenced formulations, and, with prompting, moved toward more abstract algebraic ones. For instance, in a problem in which the number of green blocks was always four greater than then number of red blocks, one boy moved from:

```
IF :REDS = 10 [ MAKE :GREENS 14 ]
to
MAKE :GREENS FOUR MORE THAN REDS
to
G. = R. + 4
```

Thus, he seems to have used knowledge of Logo as a catalyst in moving from a descriptive, specific-number statement to a generalized algebraic equation.

Noss concluded that Logo can benefit children in the development of the concept of variable. However, he emphasized that this does not mean that children necessarily have gained specific information about variables or algebra. They may have, however, gained a conceptual framework—based on intuitions and "primitive conceptions" from Logo experiences—upon which later algebraic learning can be based.

Lawler (1986) has shown how the idea of the systematic incrementing of variables developed in a Logo context and was subsequently extended to other contexts. His 8-year-old son, Robby, had been introduced to variables through extensive work with a POLYSPI procedure. This work had emphasized the notion that focussing on the systematic changes of a single variable was a fruitful way to understanding the results from complex interactions of several variables. Robby next explored the effects of changing the inputs to an INSPI procedure (in which the amount of turn is incremented, rather than the distance). He used the idea of incrementing a (potentially interesting) variable to explore the variety of designs produced with this procedure.

Perhaps most interesting is Robby's later experience with a paper-cutting puzzle. Briefly, in this puzzle one tapes two loops perpendicularly and cuts around their middles. Surprisingly, a square is produced. Robby did not stop there, however. He joined and cut three circles. After a while, he found that two rectangles were produced. Four circles yielded four squares. Five produced two identical non-planar shapes and so on. Robby then hit upon an idea: "Hey! I've got a new theory: the odd-numbered circles make two and the evens all stay together." When asked how he had gotten the idea for the exploration, he explained, "It's just like what we did at Logo with the shape

families. I changed one thing, a little at a time" (p. 78). So we see that Robby saw the earlier Logo activities as the embodiment of the powerful idea of systematically incrementing a variable. When asked to provide his theory, he constructed a seven circle puzzle, expecting—and demonstrating—that this did indeed produce two figures. Hypothesis testing had emerged from a "non-theoretical" but orderly investigation of interesting effects. Lawler claims that incrementing variables had become a method of determining "what's what."

In summary, students' ability to generalize their Logo-based notion of variable may depend to a great degree on the depth of their Logo experience and the support given them in making the abstraction and generalization. This is not surprising, considering research results from other domains. Nevertheless, there is some evidence that Logo provides an "entry" to the use of the powerful tool of algebra.

It is also no surprise that research has uncovered many difficulties and misconceptions experienced by students. The next column will discuss such problems along with research-based suggestions for helping students deal with them.

[Doug Clements, 401 White Hall, Kent State University, Kent, OH 44242.]

References

- Carmichael, H. W., Burnett, J. D., Higginson, W. C., Moore, B. G., & Pollard, P. J. (1985). *Computers, children and classrooms: A multisite evaluation of the creative use of microcomputers by elementary school children*. Toronto, Ontario, Canada: Ministry of Education.
- Davis, R. B. (1984). *Learning mathematics: The cognitive science approach to mathematics education*. Norwood, NJ: Ablex Publishing Co.
- Findlayson, H. M. (1984). *The transfer of mathematical problem solving skills from Logo experience*. D.A.I. Research Paper No. 238. University of Edinburgh, Edinburgh, Scotland.
- Hart, M. (1982). *Using computers to understand mathematics, four years on*. *Mathematics Teaching*, 98, 52-54.
- Lawler, R. W. (1986). Part I. *Natural learning: people, computers and everyday number knowledge*. In R. Lawler, B. du Boulay, M. Hughes, & H. Macleod (Eds.), *Cognition and computers: Studies in learning* (pp. 9-80). Chichester, England: Ellis Horwood Limited.
- Lehrer, R., & Smith, P. (1986, April). *Logo learning: Is more better?* Paper presented at the annual meeting of the American Educational Research Association, San Francisco.
- Milner, S. (1973, February). *The effects of computer programming on performance in mathematics*. Paper presented at the annual meeting of the American Education Research Association, New Orleans, LA. (ERIC Document Reproduction Service No. ED 076 391)
- Nelson, G. T. (1986). Development of fourth-graders' concept of literal symbols through computer-oriented problem-solving activities. *Dissertation Abstracts International*, 47, 2607A. (University Microfilms No. DA8526359)

RESEARCH ON VARIABLES, ALGEBRA, AND LOGO — CONTINUED

- Noss, R. (1985). *Creating a mathematical environment through programming: A study of young children learning Logo*. Unpublished doctoral dissertation, Chelsea College, University of London, London, U.K.
- Oprea, J. M. (1984). The effects of computer programming on a student's mathematical generalization and understanding of variables. *Dissertation Abstracts International*, 46, 369A. (University Microfilms No. DA88504061)
- Soloway, E., Lochhead, J., & Clement, J. (1982). Does computer programming enhance problem solving ability? Some positive evidence on algebra word problems. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), *Computer Literacy* (pp. 171-185). New York: Academic Press.
- Sutherland, R. (1987?). *What are the links between variable in Logo and variable in algebra?* Unpublished manuscript, University of London Institute of Education, London, England.

LOGO BOOK REVIEWS — WITH SARAH JOYCE

TEACHING WITH LOGO: Building Blocks for Learning. By Molly Watt and Dan Watt. xix and 364 pp.; diags., appendices, bibliogr., index. Menio Park, California: Addison-Wesley, 1986

The use of Logo in schools has received increasing attention in recent years. Among the many publications on this topic is this book by Molly and Dan Watt which is not only a practical guide for teachers and parents but also a description of a philosophy of teaching.

Logo is an interactive activity by means of which the students learn through discovery. It is the authors' intent that this book be approached in a similar fashion. The reader is invited to explore the suggested activities as would his or her students. Thus, readers should not only learn about Logo but should also become acquainted with their own learning style. In addition, the reader is treated in a manner that is friendly, casual and always supportive. Throughout this process, the authors appear to be modelling the teaching methods used in their own classrooms and, in so doing, provide an excellent example of good pedagogical practice.

The Watts are well qualified to undertake a book such as this. Dan Watt acquired experience as a programmer when a research assistant at M.I.T. during a sabbatical leave from his seven year teaching position. While at M.I.T. he continued his association with the classroom by working regularly with a small group of fifth grade students. Molly Watt, on the other hand, came later to the field of computers in education after many years as a public school teacher. One of her educational emphases is to ensure a supportive learning environment for students.

LOGO BOOK REVIEWS — CONTINUED

The combination of their learning and teaching styles helps to make this a well thought-out and practical book. Both authors are enthusiastic, dedicated and reflective teachers who have twelve years of Logo teaching experience between them.

"Teaching with Logo" is beautifully produced and easy to read. The book is organized around the learning experiences of five hypothetical grade five students and their teachers (the Watts) as they proceed through a series of clearly presented classroom activities. The diary entries which were kept by the students after each lesson are included in the text. This format and the style - large child type printing - not only models a method for teachers to observe and listen to their students' ideas, but demonstrates the Watts' commitment to child-centred education. The teachers' notes, which are also included, provide the Watts a vehicle for emphasizing certain teaching strategies.

The book has nine chapters and an Afterword. The brief introductory chapter describes twenty ideas such as recursion, the Total Turtle Trip Theorem, debugging, naming and variables, which the authors identify as being 'embedded' in the Logo language. This chapter also acts as a guide to the use of the book.

The next seven chapters lead students to the point where they are programming at an intermediate level using five common starting points in Logo. Chapter 8 provides specific suggestions for teachers and chapter 9 contains a set of activities for use by advanced students. The Afterword summarizes some hopes, dreams and problems regarding the future of Logo use in schools, and underscores the need for improved understanding of the potential and of the limitations of Logo. Diagrams and illustrations are clear, comprehensive and enhance the text. An annotated bibliography on Logo is included together with a list of other sources which have guided the philosophical and educational point of view of the authors. The procedures in the book are written in Terrapin/Krell dialect of Logo, but there is also an appendix providing a listing of these procedure for LCSI dialects of Logo for Apple, I.B.M. and Atari.

The main strength of the book lies in its practicality. It is written as a guide and the authors never lose sight of their audience. They are teachers first, proponents of Logo second. Although there is a chapter devoted specifically to the role of the teacher, this topic is also addressed throughout the book. The teachers' work and thoughts in both the computer environment and the classroom environment are described. There are, for example, guidelines for running classroom meetings, a checklist for students to ensure that they have understood the expectations of the teacher, a demonstrated method of maintaining class records and finally, each chapter concludes with an example of

LOGO BOOK REVIEWS -- CONTINUED

teachers' notes on the individual students. This attention to instructional details enhances the value of the book for both teacher and the parents.

This book does not see the introduction of Logo into the classrooms the sole answer to many of the ills present there. It includes assessments of both the potential and limitations of Logo. The Watts see Logo as a growing culture that needs to be developed and enriched and one of the aims of this book is that it be a contribution to this culture. It is a welcome addition to it.

It is tempting to read completely through the book without using a computer but its intent, as in Logo itself, is to allow teachers to learn through self directed projects and explorations. It is, therefore, more beneficial for the reader to work through the Logo examples on a computer whilst reading. The authors' claim to incorporate the instructional role of the teacher, but some readers who are unfamiliar with Logo may find this insufficient and may run into difficulties. These, however, are attributable to the nature of Logo itself rather than any deficiencies in the authors' presentation. At this point, an ally in the computer culture would be an asset.

To one who has experienced a great deal of frustration in both trying to learn Logo and in understanding its place in education, this book has shown what a computer learning environment can be when organized and operated by enthusiastic, dedicated and caring teachers. It models a complete and detailed teaching method which would not be difficult for the average teacher or parent to follow. Anyone who has thought about the nature of Logo and its place in education will appreciate the reflections and the insights in the book. Others will gain a great deal from the honest appraisal of its potential.

This well written and attractively produced book achieves its aims. It is valuable not only for its contribution to the understanding of the teaching of Logo but also for its contribution to the understanding of the meaning of teaching. It will educate readers who are eager to become involved with the teaching of Logo and will prove a useful addition to their libraries.

Ms. Sarah Joyce, 418 Edinburgh Street, Burnaby, B.C. Canada V5C 1R9.

LogoPals

"MIX, STIX, and Logo Penpals"

by Barbara Randolph

A new opportunity now exists for Logo educators and students who would like to teleconference with others who are working with Logo. MIX is a new educators information exchange with two conferences of special interest to Logo users. Its "Logo Conference" has been set up especially for Logo educators.

Logo students can participate in STIX, a student information exchange. Ardu Erlich, a junior high school teacher in St. Paul, MN, is the moderator. She's inviting Logo students to use their modems to become involved in Logo Penpals. The letters you write will include some fun projects and the programs you write for them.

Those who are interested can write to: MIX, EMS-McGraw Hill, 9855 West 78th Street, Eden Prairie, Minnesota, 55344 (USA) and inquire about a subscription to MIX. (Or you can call 1-800-622-6310). I would like to thank Grif Wigley, the editor of MIX, and Ardu for informing us about these new Logo electronic mail activities.

Logo students worldwide who would like to write to each other via the postal mail can become one of our LogoPals. Have them write a letter, telling their favorite Logo programs and projects, their hobbies and interests, and their age and grade. They can be matched with other girls and boys in our LogoPal network. Requests for penpals from certain places will be accommodated when possible.

Children in the USA are requested to send a self-addressed stamped envelope. Those students outside the USA should include international postal coupons (purchased at the post office) for a 1-ounce or 28-gram reply. To save on postage, a teacher can send one package containing a group of letters.

Write to: **LogoPals**
c/o Barbara Randolph
1455 East 56th Street
Chicago, Illinois 60637
USA

If you would like to establish correspondence with another teacher and class in the United States or in another country in the world, then join the Logo Class Penpal Network. You will be able to share Logo teaching ideas, procedures and possibly create projects to work on together. To obtain a free application kit, send a long stamped self addressed envelope to:

Logo Class Penpal Network
University of Virginia
Curry School of Education
Ruffner Hall
Charlottesville, Virginia 22903
USA

Take advantage of these opportunities to communicate with other Logo teachers and learners. We have so much to share with each other—and it's fun too!

Barbara Randolph is a library and instructional media center teacher in the Chicago Public Schools.

Meet the SIGLogo Board

SIGLogo is the Special Interest Group for Logo-Using Educators of the ICCE. As a subscriber to the *Logo Exchange* you become a member of this newly formed group. SIGLogo is governed by a Board of Directors that includes the President, Past-President, President-Elect, Vice President and Communications Officer. ICCE appointed three acting officers to draft the SIGLogo Bylaws. Peter Rawitsch, from Guilderland Center, New York, is the Acting President. Gary S. Stager, from Wayne New Jersey, is the Acting Vice-President. The Communications Officer is Theodore M. Norton, from Pelham, Massachusetts. These three individuals bring diverse Logo backgrounds to the SIGLogo Board.

Peter Rawitsch teaches first grade in Glenmont, New York. He tells his students that "you can do anything in Logo" and provides them with opportunities to make their dreams come true. Peter co-founded the Albany-Schenectady-Troy Logo Users' Group (ASTROLUG) three years ago. He is co-editor of the ASTROLUG newsletter, *Pendown*. Peter has taught Logo courses at the State University of New York in Albany. He enjoys using songs to illustrate Logo concepts and ideas. He is working on creating new visual and musical learning models.

Gary S. Stager is the Director of Training for the Network for Action in Microcomputer Education (NAME). NAME provides computer training and support to over 70 school districts in New Jersey. Gary's interest in the curriculum applications of Logo has led him to develop a variety of tool procedures and microworlds. He has presented his work at LOGO '86 and the East and West Coast Logo Conferences. His session "If There Had Been Logo...I Might Have Learned Math," focused on providing children with a learning environment to explore and manipulate mathematical concepts. Gary is currently getting ready to host the second annual New Jersey Educational Computing Conference on March 25th and 26th in Somerset.

Theodore M. Norton holds a doctorate in political theory from New York University and is currently earning a second doctorate in Cognitive Studies of Computers in Learning, at the University of Massachusetts-Amherst. He has taught social theory, cultural criticism, and history at Vassar College, Hampshire College, and New York University. Ted is interested in computing, cultural issues, and education; his work with Logo has helped him focus his research in these areas. He looks forward to the day when Logo will support the rich visual and sound resources offered by some of the newer microcomputers. He welcomes continuing discussion on Logo as philosophy of education.

The first business meeting of the Board members was held in early December near Albany, New York. Using the Bylaws of the SIGTE as a model, they proposed that the purpose of SIGLogo is to "support and promote the use of Logo as a tool for learning and as a language for computer science." A copy of the complete Bylaws will appear in a future issue of the *Logo Exchange*. Members of SIGLogo will have an opportunity to vote on the Bylaws this June at NECC '88 in Dallas, Texas. Peter, Gary, and Ted welcome your thoughts and concerns about SIGLogo. Members can write to them at the following addresses.

Peter Rawitsch

P. O. Box 254

Guilderland Center, New York 12085

Gary Stager

12 Locust Place

Wayne, New Jersey 07470

Ted Norton

3 Pine Tree Circle

Pelham, Massachusetts 01002

InLXual Challenges

Worms as Algorithms

by Robs Muir

In this competitive and highly-paid profession of writing computing challenges for major journals, it is considered unwise to praise a colleague's column too highly. There are just so many prestigious magazines to go around.

It is even more galling to be in a position of having to recommend another author's work which is so *blatantly* patterned after this column. Never mind that "Recreational Computing" predated this column by a mere 30 days. Never mind that "Recreational Computing" gets more space. Never mind that it is published in *Scientific American*. The tragedy is that A. K. Dewdney is such a great writer! (If your Jr. High or High School students haven't read Dewdney's *Planiverse*, they have missed a universe of new ideas.)

However as a duty to you, gentle reader, I feel bound to inform you of the monthly column published in *Scientific American* titled "Recreational Computing." More specifically, I'd like to direct your attention to the December, 1987 issue which was subtitled, "Simple special effects illustrate the art of converting algorithms into programs." It is as good introduction to this continuing series as we could hope for and can serve as a delightful springboard for this month's challenge.

Special Effects

Solving a problem (either on or off of a computer) can be more easily facilitated through some specific techniques. One such technique is to develop a solution using an *algorithmic* language—that is, a sufficiently detailed description of the steps required to reach the goal. If the algorithm is written at a high enough level, solutions to a problem can often be seen

more clearly. This is one of the overlooked benefits of computer programming in education; the true goal of educational programming is not merely to control a computer, it is to provide a rich environment in which to experience problem-solving heuristics combined with immediate feedback.

Dewdney's "Recreational Computing" article of December details how an algorithm can be translated into a computer program. The "challenge" is to build a special effect—a cybernetic worm—which will "crawl" realistically across the screen. Dewdney carefully details how to solve this challenge using an algorithmic language, and how to then translate this algorithm into a BASIC program. Well... since Logo already has turtles, why not add WORMS to our glass menagerie?

Managing by Wandering About

Before we build a complicated WORM, how would we get a turtle to "crawl" about? A simple linear crawl might be done like this:

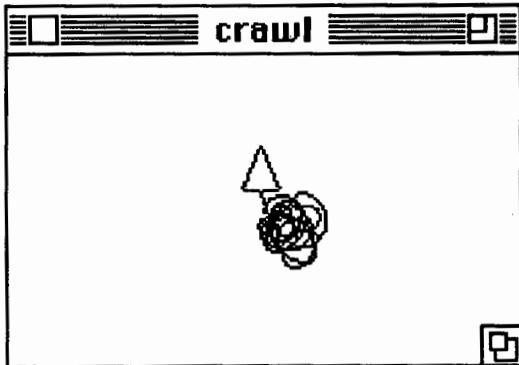
```
TO CRAWL
  FORWARD 1
  CRAWL
END
```

To build a 2-dimensional wandering turtle, we could add a line that would cause the turtle to randomly turn some amount before CRAWLING again. Since Logo's RANDOM operator produces a value between zero and its input value, we can then use this as an input to a RIGHT command. Perhaps a new CRAWL might look like this:

```
TO CRAWL
  RIGHT (RANDOM 90)
  FORWARD 1
  CRAWL
END
```

INLXAL CHALLENGES — CONTINUED

CRAWL



How do we fix this *listing* behavior? (Real turtles don't always move to the right, nor do I expect do real worms.) What might we do if we wanted a random turn to the right or the left? A terribly BASIC approach might look like this

```
MAKE "VALUE (RANDOM 2)
IF :VALUE = 0 [RIGHT (RANDOM 90)]
IF :VALUE = 1 [LEFT (RANDOM 90)]
```

Just stick all of this into CRAWL in place of the RIGHT (RANDOM 90) line.

In an algorithmic language, this bit of code might read

```
Assign to "VALUE either a 0 or a 1.
If :value = 0 then turn right some
    random amount between 0 and 89,
or
```

```
If :value = 1 then turn left some
    random amount between 0 and 89.
```

Most Logophiles would agree, this is indeed *ugly!* Is there an easier way?

What would this line do?

```
RIGHT (90 - RANDOM 180)
```

Read this line backwards; that is, parse it the way that the Logo interpreter must:

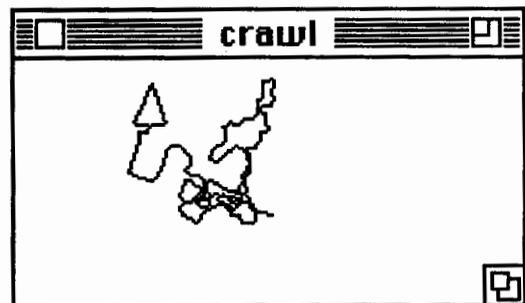
Get a random number between 0 and 179.. subtract it from 90...then use this to cause the turtle to turn RIGHT.

If Logo "chooses" a number such as 150 and then subtracts it from 90, we are left with a negative number (-60). If the turtle turns RIGHT -60, it will be the same as turning LEFT 60.

Here is the new, improved CRAWL

```
TO CRAWL
  RIGHT 90 - (RANDOM 180)
  FORWARD 1
  CRAWL
END
```

CRAWL



Once the basic procedure is defined, it is a simple and enjoyable process to refine the CRAWL procedure so that it approximates the behavior of "organic" life forms. Perhaps the FORWARD distance should be adjusted so that the turtle covers more ground during each recursion. Perhaps the turning angles should be restricted so that the turtle doesn't keep spinning on its heels.

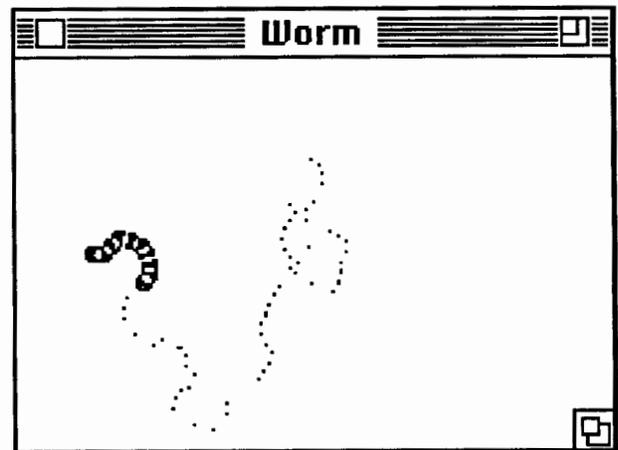
Dewdney suggests an approach to building a worm that crawls much like our turtle. Since he is writing for an audience that is (apparently) unaccustomed to powerful languages like Logo, Dewdney's algorithm includes some "high school" trigonometry to provide the rudiments of turtle-graphics and some semblance of randomly determined new positions. We can ignore this.

INLXUAL CHALLENGES — CONTINUED

Dewdney's worm is a nothing more than a chain of small circles. To create movement, this segmented worm moves by drawing a new circle at the "head" and then erasing one of the circles at the "tail" of the worm. The most difficult task in this program is keeping track of where the train of circles is so that the last of the series can be erased at the appropriate moment. One data structure suggested by Dewdney is the *array*. While most Logos do not support arrays (notable exceptions include *Object Logo*, *LSCI Logo* for the Macintosh, and *Terrapin Logo* for the Mac, among others), all respectable Logos include *lists* which can serve the same purpose.

Using a list to store the positions of each segment of the worm is just one of several possible variations of Dewdney's challenge which we can tap using Logo. Drawing and undrawing the worm segments could be done using the STAMP command in LogoWriter. Turtles need not play a role; if you choose, you could follow Dewdney's algorithm very closely. The choice is yours.

I submit evidence that WORMS do exist in Logo. My Object Logo worm crawls very believably across the screen; as you can see, it even leaves a little trail of "slime" on the sidewalk. The possibilities for building Logo games and projects from this special effect will flood in upon you as you refine the behavior of your Logo worms.



Happy bug hunting!

Another Challenge?!

Tom Lough has contributed the following mathematical Logo challenge. Build a random number generator which will return a set of numbers drawn from a normal population, when given as input the mean, the standard deviation, and the number of score numbers needed. For example,

```
?SHOW NORMAL 500 100 10
[505 487 465 529 236 710 550 422 635
350]
```

Any takers?

Robs Muir, 1688 Denver Ave., Claremont, CA 91711
CompuServe: 70357,3403
BITnet: MUIRR@CLARGRAD

Teaching Tools

Frog Algebra

by Glen Bull and Gina Bull

"...in school math 'analytic geometry' has become synonymous with the representation of curves by equations. As a result every educated person vaguely remembers that $y = x^2$ is the equation of a parabola. And although most parents have very little idea of why anyone should know this, they become indignant when their children do not."

Seymour Papert, *Mindstorms*, p. 52

One of the first versions of Logo developed for a microcomputer was created for the Texas Instruments 99/4 personal computer. Our TI microcomputer did not even have disk drives; programs were saved on cassette tape.

The TI Logo had a wonderful feature — it was possible to change the shape of the turtle. One of the first Logo programs which we developed made use of this feature to create a leaping frog. The frog changed its shape as it jumped through the air. In reality, it was a turtle in frog's clothing.

Recent versions of Logo such as LogoWriter and LCSII Logo II for the Apple also make it easy to change the shape of the turtle. The ease with which this can be accomplished caused us to resurrect ... Frog Algebra!

Knee-Deep in Frog Algebra

Many find frog algebra a ribbiting subject. It is not necessary to change the shape of the turtle to experience frog algebra — it can be accomplished with any version of Logo. The remarks by Papert at the beginning of the column were the stimulus for our explorations. In a discussion of why certain topics are included in the traditional mathematics curriculum, Papert mentioned in passing that the formula $y=x^2$ produces a parabola. This half-remembered bit of

algebraic trivia gave us the clue to the solution of a Logo puzzle we had encountered.

A student in a Logo class had created a frog shape in TI Logo, and had requested a procedure to make the frog jump. Our first effort proved unsatisfactory.

```
TO JUMP :DISTANCE
  FORWARD :DISTANCE
  RIGHT 90
  FORWARD :DISTANCE
  LEFT 90
  BACK :DISTANCE
END
```

This procedure produced a jump which looked like this:

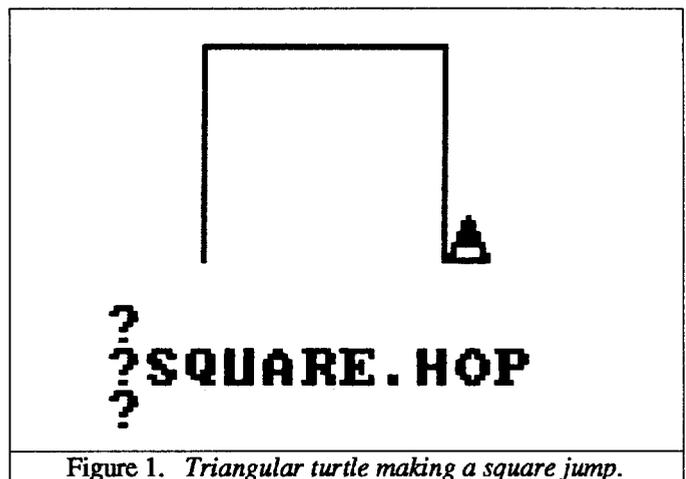


Figure 1. Triangular turtle making a square jump.

The student correctly observed that very seldom does an object in nature move straight up and straight across. A dragonfly or a hummingbird might move in this way ... but not a frog. Papert's remarks suggested a possible solution. The arc of a parabola more nearly represents the path of a leap. How then to make the turtle trace the path of a parabola?

TEACHING TOOLS — CONTINUED

Parabolic Polliwogs

The formula for a parabola is an algebraic function. We first created a Logo procedure which was the equivalent of the function in algebra.

```
TO FUNCTION1 :NUMBER
  IF :NUMBER > 3 [STOP]
  TYPE :NUMBER * :NUMBER
  FUNCTION1 :NUMBER + 1
END
```

The FUNCTION1 procedure generates values which lie on a parabolic curve.

```
FUNCTION1 -3
9410149
```

The numbers generated represent the height of the turtle at each point along the x-axis from -3 to +3. However, the numbers printed by the FUNCTION1 procedure needed spaces between them. Also, our leap was upside down.

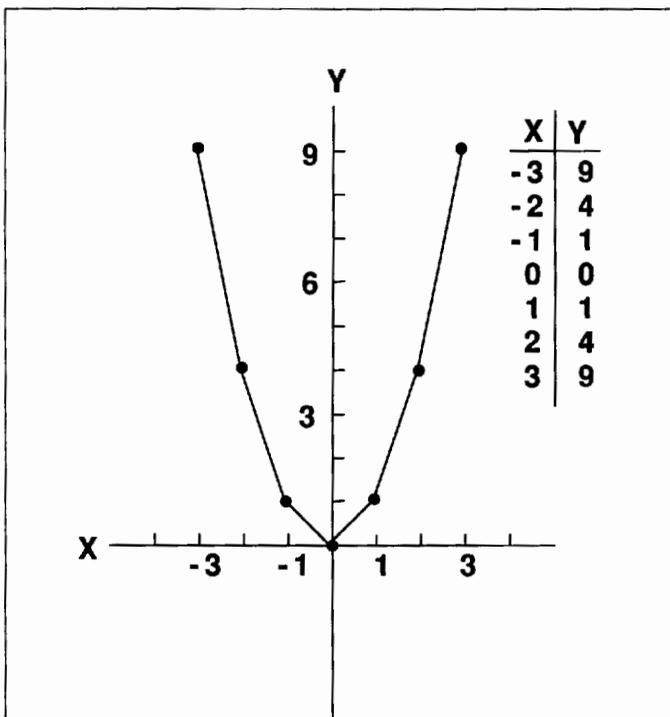


Figure 2. Plot of a parabola on a Cartesian grid

Multiplying by minus one inverts the parabola. Adding a space makes the numbers easier to read.

```
TO FUNCTION2 :NUMBER
  IF :NUMBER > 8 [STOP]
  TYPE SENTENCE ->
    ( -1 * :NUMBER * :NUMBER ) SPACE
  FUNCTION2 :NUMBER + 1
END

TO SPACE
  OUTPUT CHAR 32
END
```

CHARacter 32 is a space in most computer languages, including Logo. This character is used to insert a space between each number.

When FUNCTION2 is run, it generates a list of the squares of all the numbers between -8 and +8. Each square is multiplied by minus one to make each number negative.

```
FUNCTION2 -8
-64 -49 -36 -25 -16 -9 -4 -1 0 -1 -4
-9 -16 -25 -36 -49 -64
```

If this list of numbers is plotted, it produces a parabola. If the values are used to position the turtle, it appears to leap.

The HOP Procedure

We will use this list of values repeatedly with the HOP procedure, so it is convenient to type the values as part of a procedure:

```
TO PARABOLA
  OUTPUT [-64 -49 -36 -25 -16 -9 ->
    -4 -1 0 -1 -4 -9 -16 ->
    -25 -36 -49 -64]
END
```

Once you have typed in the list of numbers, be sure to save the PARABOLA procedure to disk in case your computer croaks. The HOP procedure uses the list of values in PARABOLA. HOP moves the

TEACHING TOOLS — CONTINUED

turtle to the first value in the PARABOLA list, then to the next, and so on, as it moves through the arc of the parabola.

The HOP procedure uses SETPOS (SETXY in Terrapin) to shift the turtle from point to point. This version of HOP is for Apple Logo, Apple Logo II, or IBM Logo. Versions for Terrapin Logo and for LogoWriter are listed at the end of the column.

```
TO HOP :PLACE
  IF EMPTY? :PLACE [STOP]
  SETPOS LIST (FIRST POS) + 5 (FIRST
    :PLACE)
  HOP BUTFIRST :PLACE
END
```

HOP shifts the turtle through the list of values output by the parabola procedure.

HOP PARABOLA

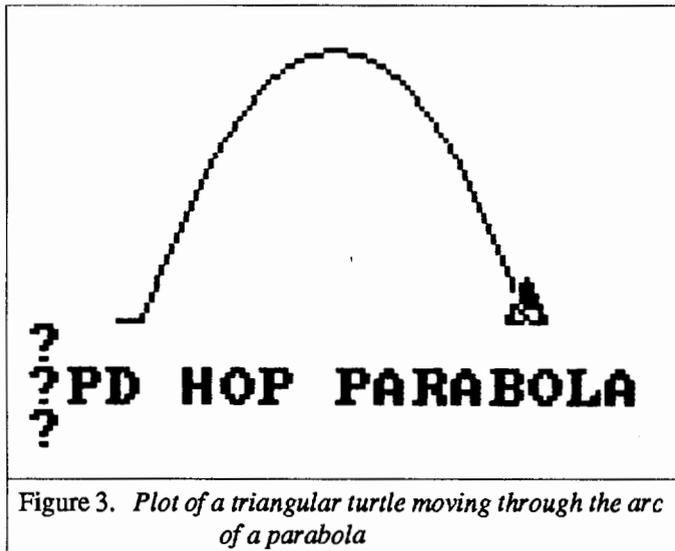


Figure 3. Plot of a triangular turtle moving through the arc of a parabola

The HOP procedure always produces a leap that travels the same distance. We leave it as an exercise for the reader to enhance the HOP procedure by adding variables which allow the turtle to travel different distances.

A Turtle in Frog's Clothing

The traditional triangular turtle shape can be used

with the HOP procedure. However, it is fun to change the turtle's shape into a frog as it leaps. Three different frog shapes can be used to produce an effective animation. The three shapes are:

- a sitting frog
- a frog leaping up
- a frog leaping down

It is possible to change the turtle's shape in a number of different versions of Logo. We suspect that few LX readers use TI Logo any longer, but changing the turtle's shape is particularly easy in LCSII Logo II for the Apple, LogoWriter (for IBM, Apple, or Commodore), and Commodore Logo. It is also possible to change the shape of the turtle in other versions such as IBM Log, Terrapin Logo, and Object Logo.

LCSII Logo for the Apple II, LogoWriter, and Commodore Logo provides a grid with squares which can be filled in. These squares define the shape of the turtle, using the same principle as needlepoint. The grid in LogoWriter for the IBM is 16 by 16. The figure below shows our version of the three frog shapes. However, you can experiment with your own frog shapes (or other animated figures). A high jumper would be a natural candidate for a parabola.

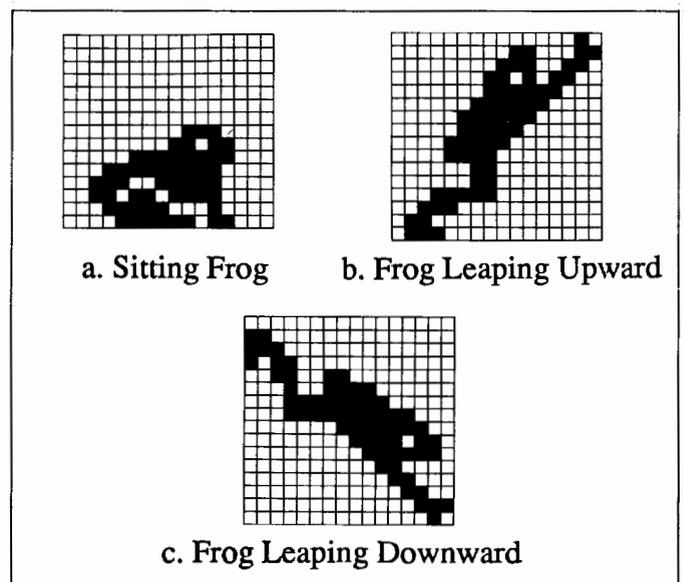


Figure 4. Diagrams of three different frog figures developed on 16 x 16 grids

TEACHING TOOLS — CONTINUED

The frog begins the leap in the sitting position (Figure 4a). As it leaps upwards, the shape is changed to an frog stretching upwards (Figure 4b). As the frog passes the top of the arc, the shape is changed to a frog pointing downward (Figure 4c). When the frog reaches the ground again, the shape is changed back to a sitting frog.

The trick is to determine when the frog should change from one shape to the next. There are several ways to accomplish this. We have chosen to modify HOP by adding a CHANGE.SHAPE? procedure. The CHANGE.SHAPE? procedure asks the question, "Is it time to change the frog's shape?" The modified HOP procedure is called LEAP. The LEAP procedure shown is for LogoWriter. The LEAP procedure for other versions of Logo would be created by adding CHANGE.SHAPE? to the HOP procedure for that version.

```
TO LEAP :PLACE
  IF EMPTY? :PLACE [STOP]
  CHANGE.SHAPE? (COUNT :PLACE)
  SETPOS LIST (FIRST POS) + 5 ->
    (FIRST :PLACE)
  LEAP BUTFIRST :PLACE
END
```

The CHANGE.SHAPE? procedure used by LEAP checks the position of the turtle and changes the shape of the turtle at the appropriate locations. There are 17 positions in the list output by the PARABOLA procedure. When there are 16 positions left, the shape is set to Shape 2 (Frog Up). When there are only 9 positions left, the shape is set to Shape 3 (Frog Down). When there is one position left, the shape is set to Shape 1 (Frog Sitting).

```
TO CHANGE.SHAPE? :POSITION
  IF :POSITION = 16 [SETSH 2]
  IF :POSITION = 9 [SETSH 3 STOP]
  IF :POSITION = 1 [SETSH 1 STOP]
END
```



Figure 5. Diagram of an animated turtle changing shapes as it moves through the arc of a parabola

Leaping to Conclusions

Any equation can be substituted for the parabola function used in this instance. A quadratic equation will work as well as a parabola. A sine wave will give the turtle a roller coaster ride. (See "Turtles, Lists, and Spaceships" from the May 1986 NLX column for a more extensive discussion of this possibility.) A computer graph of an algebraic function gives students a quick look at its overall shape. Plots which formerly took hours can be crunched by the computer in moments.

A more convenient way of capturing the output of different algebraic functions is needed to encourage extensive experimentation, since students can not be expected to type the numbers into a procedure each time as we did with PARABOLA. In LogoWriter this can be accomplished by using a FUNCTION procedure to generate the numbers in the command center, and then using the Cut and Paste facilities to paste the numbers into a procedure such as PARABOLA.

A different method which will work for Apple Logo II, LCSII Logo for the Apple II, and IBM Logo was suggested in last month's column, "Logophiles

FROG ALGEBRA — CONTINUED

and Logo Files." The data can be sent to a file on disk as it is generated:

```
DRIBBLE "DATA
FUNCTION2 -8
NODRIBBLE
```

After the data generated by the function is stored on a file on disk, the file can be edited to convert the data into a procedure, using the EDITFILE command:

```
EDITFILE "DATA
```

Edit the file by deleting the NODRIBBLE at the end of the file. (Remember that everything that appears on the screen will be sent to the file after DRIBBLE is typed.) Then place a TO PARABOLA on the line before the data, and an END on the line after the data. Place brackets around the numbers, so the file looks like this:

```
TO PARABOLA
  OUTPUT [-64 -49 -36 -25 -16 ->
          -9 -4 -1 0 -1 -4 -9 ->
          -16 -25 -36 -49 -64]
END
```

Exit the editor in the normal way. You should hear the disk drive whirl as the changes are saved to disk. Next load the procedure from disk into the workspace of the computer:

```
LOAD "DATA
```

Once you master this method of sending data to a file on disk, you can use it to capture data which would be too tedious and time consuming to retype. (At the same time, you will increase your familiarity with the concept of files.)

It would also be possible to capture the output of the function to be graphed in the form of a global variable. No matter which of several alternative techniques is used, generating the numbers which are later graphed gives students an intimate understand-

ing of the relationship between the numbers and the graph produced.

Why Frog Algebra?

One answer to the question, "Why Frog Algebra" is "Why Turtle Geometry?". A more serious answer is found in an editorial by Dave Moursund in the November 1987 issue of *The Computing Teacher*. In an article titled "Mathematics Education: A Modest Proposal for Change," Moursund suggests that computer graphics be used as a general-purpose aid to solving one-variable equations in high school algebra classes. Graphing equations provides a useful method for understanding the relationship between an equation and the picture it produces. Frog Algebra even makes it fun!

Glen Bull and Gina Bull, Curry School of Education, Ruffner Hall, University of Virginia, Charlottesville, VA 22903; CompuServe: 72477,1637.

Appendix

(Does a frog have an appendix?)

Alternate Versions of HOP

Terrapin Version

```
TO HOP :PLACE
  IF EMPTY? :PLACE STOP
  SETXY XCOR + 5 (FIRST :PLACE)
  HOP BUTFIRST :PLACE
END
```

LogoWriter Version

```
TO HOP :PLACE
  IF EMPTY? :PLACE [STOP]
  SETPOS LIST (FIRST POS) + 5 ->
    (FIRST :PLACE)
  HOP BUTFIRST :PLACE
END
```

Stager's Stuff

"Playing With Music" Part 2: Child Composers

by Gary S. Stager

In last month's article, I discussed how Logo and music can be used as a vehicle for problem solving and interdisciplinary learning. A number of Logo tools were provided for exploring musical concepts. This article will demonstrate how Logo can be used as a tool for musical composition and notation and will provide the necessary Logo procedures. All of the Logo procedures in this article depend on the STARTUP, NOTE, and PLAY procedures from last month's article.

It is unfortunate that in many schools the only subject disliked by students more than math is music. This is quite puzzling due to children's love of music outside of school. One possible reason why students may not like music class is that much of school music is a passive activity. Students are often asked to regurgitate songs chosen by a teacher and rarely have the opportunity to experience the personal creative satisfaction of musical composition. When students do compose their own music, they rarely have the performance skills necessary to hear their creation. This problem is as common among conservatory students who can't find an orchestra to play their work as it is among third graders. Logo provides an environment in which a student can compose and hear their own composition.

Tune Blocks

One microworld you can create for students is a collection of tune blocks. Tune blocks are melodic fragments in the form of Logo procedures with which students can "build" musical compositions in the same playful way they would build a bridge with wooden blocks or a castle with LEGO™. With tune blocks, students can create compositions by combining Logo procedures in ways that are aesthetically pleasing to them. There are no right or wrong ways to

combine tune blocks. Students should be encouraged to express themselves.

The simplicity of composing with tune blocks combined with the primitive music capabilities inherent in Logo force student composers to focus on fundamental musical concepts including: pitch, duration, sequence, melody, and tonality.

There are very simple rules for using tune blocks. You may repeat the blocks as often as you wish, however, you must use every tune block at least once in a composition. If a couple of tune blocks sound like the beginning of a song, you might want to create a subprocedure including them called BEGINNING or FIRST.PART. Break compositions down into their component parts as you do with turtle graphic procedures.

Tune blocks should be created from songs your students are not familiar with. This allows them to be creative without the pressure of building the "correct" composition. If tune blocks are of a familiar song, the identity of the song should remain a mystery to the children. Procedure names for tune blocks should be simple to type and should not provide clues for composing. Single letters are good names for tune block procedures.

The following is a collection of tune blocks for you to play with. Can you assemble the puzzle and name the tune?

```

TO T1
  PLAY [ 5 5 4 4 2 2 0 ] ->
        [ 4 4 4 4 4 4 8 ]
END

TO T2
  PLAY [ 7 7 5 5 4 4 2 ] ->
        [ 4 4 4 4 4 4 8 ]
END

TO T3
  PLAY [ 0 0 7 7 9 9 7 ] ->
        [ 4 4 4 4 4 4 8 ]
END

```

"PLAYING WITH MUSIC" — CONTINUED

```

TO T4
  PLAY [ 7 7 5 5 4 4 2 ] ->
        [ 4 4 4 4 4 4 8 ]
END

TO T5
  PLAY [ 5 5 4 4 2 2 0 ] ->
        [ 4 4 4 4 4 4 8 ]
END

TO T6
  PLAY [ 0 0 7 7 9 9 7 ] ->
        [ 4 4 4 4 4 4 8 ]
END

TO SONG1
  T3
  T1
  T2
  T2
  T3
  T1
END

TO SONG2
  T3
  T1
  T2
  T4
  T5
  T6
END

```

You may have noticed that SONG1 and SONG2 are identical songs, yet SONG2 uses six tune blocks instead of three. Having extra blocks, even the same ones, challenge some student's perceptions.

Try these tune blocks:

```

TO S1
  PLAY [ 6 9 14 13 11 13 11 9 ] ->
        [ 3 3 3 3 2 2 2 6 ]
END

TO S2
  PLAY [ 6 9 2 ] [ 3 3 6 ]
END

TO S3
  PLAY [ 2 6 9 6 4 6 4 2 ] ->
        [ 3 3 3 3 2 2 2 6 ]
END

```

Tone Poems

Using some of the tools provided in last month's article, (PLAY, UP, DOWN, TRANSPOSE, RTUNE) students can create tone poems or musical stories. For example, you can tell a story by writing musical procedures that remind you of a bird chirping, or rain falling, or footsteps and combining these sound effects (subprocedures) into a coherent composition.

Serial Music and Atonality

Serial music is an important, complex, and misunderstood style of Twentieth Century music. The style was developed by Arnold Schoenberg and discards most of the principles of tonal (key) music. Every composition is based on a series containing all twelve tones in the chromatic scale (C C# D D# ...) in an order chosen by the composer. There are a few basic rules of atonal composition.

- 1) The order stays the same throughout the composition although certain manipulations are permitted. The most important rule is that no note may be repeated until every note in the series has been played.
- 2) The octave position of a note may be changed at will.
- 3) In addition to the original form of the tone row (series), the series may be played in its inversion (upside down), retrograde (backwards), and retrograde inversion form (upside down and backwards).
- 4) Each of the four possible forms of the series may be used starting on any position in the scale. This provides 48 different possibilities for each particular series.
- 5) The rhythms, harmony, and counterpoint of a series may be changed in an unlimited variety of ways.

The SERIAL procedure shown below takes two lists as inputs, a pitch list and a duration list, and

"PLAYING WITH MUSIC" — CONTINUED

randomly plays a row without repeating a pitch or duration until each pitch and duration is played. The randomness isn't completely consistent with the rules above, but if you are interested in a programming challenge, you can write a procedure to make a global variable to keep track of the randomly created series (for future modifications). You may also wish to write an INVERSION procedure which would change the direction of all melodic intervals.

```

TO SERIAL :PITCHES :DURATIONS
  IF OR (EMPTY? :PITCHES) ->
    (EMPTY? :DURATIONS) [ STOP ]
  MAKE "P PICK :PITCHES
  MAKE "D PICK :DURATIONS
  NOTE :P :D
  SERIAL (DELETE :P :PITCHES) ->
    (DELETE :D :DURATIONS)
END

TO DELETE :WORD :LIST
  IF NOT MEMBER? :WORD :LIST ->
    [ OUTPUT :LIST ]
  IF EQUAL? :WORD FIRST :LIST ->
    [ OUTPUT BUTFIRST :LIST ]
  OUTPUT SENTENCE (FIRST :LIST) ->
    DELETE :WORD BUTFIRST :LIST
END

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

This procedure will play a list of pitches and durations backwards.

```

TO RETRO :PITCHES :DURATIONS
  IF OR (EMPTY? :PITCHES) ->
    (EMPTY? :DURATIONS) [STOP]
  NOTE LAST :PITCHES LAST :DURATIONS
  RETRO BUTLAST :PITCHES ->
    BUTLAST :DURATIONS
END
  
```

Here are a couple of avant-garde music procedures that work much the same way as PLAY and RETRO. Can you tell what they do differently to the same list of pitches and durations?

```

TO CRAZY1 :PITCHES :DURATIONS
  IF OR (EMPTY? :PITCHES) ->
    (EMPTY? :DURATIONS) [ STOP ]
  NOTE FIRST :PITCHES LAST :DURATIONS
  CRAZY1 BUTFIRST :PITCHES ->
    BUTLAST :DURATIONS
END
  
```

```

TO CRAZY2 :PITCHES :DURATIONS
  IF OR (EMPTY? :PITCHES) ->
    (EMPTY? :DURATIONS) [ STOP ]
  NOTE LAST :PITCHES FIRST :DURATIONS
  CRAZY2 BL :PITCHES BF :DURATIONS
END
  
```

NOTE: In LCSI versions of Logo other than LogoWriter, replace MEMBER? with MEMBERP and EQUAL? with EQUALP, and EMPTY? with EMPTYTP.

Graphic Notation

One of the most difficult concepts for children to understand is musical notation. Music is the most abstract art form with the most formal language imposed on it. Young students should be encouraged to develop their own systems of notation for their compositions. Logo can be used to display a score while a composition is being played. Of course, in LogoWriter, you could actually have actual musical notes, rests, and staves drawn, but this is not desirable now.

The following new NOTE procedure will graph a pitch as it is played. It is written for LogoWriter and should be modified by the reader to efficiently use the screen dimensions of your version of Logo. It's not perfect, but it is a start. Type RESET.GRAPH before you play a composition.

```

TO NOTE :PITCH :DURATION
  IF :PITCH + :TONIC > 80 [STOP]
  GRAPH :PITCH :DURATION
  TONE ITEM (:PITCH + :TONIC) ->
    :PITCHLIST (:DURATION * :TEMPO)
END
  
```

```

TO GRAPH :PITCH :DURATION
  DOT LIST (XCOR + 1) (:PITCH * :SCALE)
  SETPOS LIST (XCOR + :DURATION-1)->
    (:PITCH * :SCALE)
END
  
```

```

TO RESET.GRAPH
  HT
  CG
  PU
  SETPOS [ -115 0 ]
END
  
```

"PLAYING WITH MUSIC" — CONTINUED

```
TO STARTUP
  SET.SCALE 1
END
```

```
TO SET.SCALE :NUMBER
  MAKE "SCALE :NUMBER
END
```

```
TO DOT :POS
  PU
  SETPOS :POS
  PD
  FORWARD 0
END
```

```
TO XCOR
  OUTPUT FIRST POS
END
```

SET.SCALE creates a variable, SCALE, which specifies the vertical scale of the graph. Also, recall that the name PITCHLIST refers to the list of notes in your melody, as explained in last month's column.

HAVE FUN!!!

Gary S. Stager, 12 Locust Place, Wayne, NJ 07470
CompuServe: 73306,2446

ICCE and MMI Develop Independent Study Courses

The International Council for Computers in Education and Media MicroWorlds are cooperating in the development of independent study courses for educators. ICCE is developing the course guides and credit arrangements, and MMI is producing videotapes and accompanying materials.

The first course to take advantage of the combined expertise of ICCE and MMI is Introduction to Logo Using *LogoWriter*. Sharon Burrowes Yoder designed the course lesson guide and materials. MMI developed a series of videotapes, called ON LOGO, which features Seymour Papert. In the ON LOGO tapes Papert shows how to use *LogoWriters* or traditional Logo to teach problem solving and thinking skills. In this course educators view the ON LOGO videotapes, do Logo programming, read and report on course materials, design lessons for their students, and correspond with the instructor by mail.

Student can take the course from virtually anywhere and start at any time. It is approved by the University of Oregon and it carries four quarter-hours of graduate credit. Additional course information is available from ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905; ph. 503/686-4414.

Information regarding ON LOGO videotapes is available from Media Microworlds, Inc., 12 Clayton Terrace, St. Louis, MO 63131.

"Logo LinX"

'sNo Mystery

by Judi Harris

No doubt many of you have seen the first (last?) snowfall of the school year by now. The luckiest among you have even had an unexpected vacation day due to this natural wonder. For some, the inexplicable peace and beauty of snow outweigh the inconvenience that it can cause. But did you ever suspect that a snowflake could both postpone AND inspire math lessons?

The substance of each snowflake is ice; frozen H₂O that is born in the clouds.

"Ah! From mountaintop!" said the King of Siam.

"From SKY, Your Majesty," corrected Anna.

"From sky TO mountaintop," revised the King.

"That's right, Your Majesty."

—Rogers and Hammerstein,
"The King and I"

Snow is a unique example of a single chemical compound that crystallizes with great variation. The ancient Greeks first used the word "crystal" to mean "ice." Now, the word crystal refers to the ways that molecules pile up in regular patterns to form solid matter.

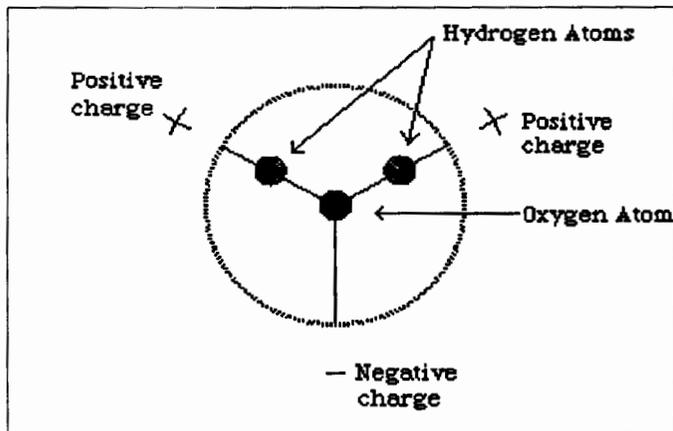
There are seven types of snow crystals: six-rayed stars, hexagonal plates, columns, bullets, needles, and irregular forms. Snow crystals clump together on their journey down from the clouds, forming snowflakes. Snow crystals can range in diameter from diamond dust size to 3/16 inch.

Frozen Solids

Although there is an infinite variety of snow types, all are based upon a pyramidal molecular structure in which atoms bond to form hexagonal molecule clusters. The water molecule is comprised of two atoms of hydrogen and one atom of oxygen. The oxygen atom is positioned at the center, with the

'SNO STORM — CONTINUED

hydrogen atoms arranged away from it at 120-degree angles. Two positive electrical charges are released in the directions of the hydrogen atoms, and a negative charge is exerted at the third 120-degree angle of the sphere:



When molecule activity is slowed down by freezing temperatures, the weak electrical charges attract one another. Then the water molecules behave like magnets, as the negative poles are drawn to the positive poles of neighboring molecules. Ice thereby forms a pyramidal structure. Since the atoms are always separated by angles of 120 degrees, snow crystal components must form in threes (the ice pyramids) and sixes. The simplest possible ice crystal would be a collection of 60 water molecules, but this would be far too small to see with the unaided eye. A snow crystal of average size contains 100 million or more ice molecules.

Not-So-Softly Falling

Snow crystal structures reflect the air temperature and water saturation level in the clouds where they form. Warm water vapor has a greater concentration of water molecules, which are more active than cold water vapor molecules. Increased molecule activity speeds up the crystallization process; colder temperatures slow down molecule activity, and therefore ice crystal production.

Needle crystals form in clouds heavy with water vapor, ranging in temperature from 25 to 28 degrees

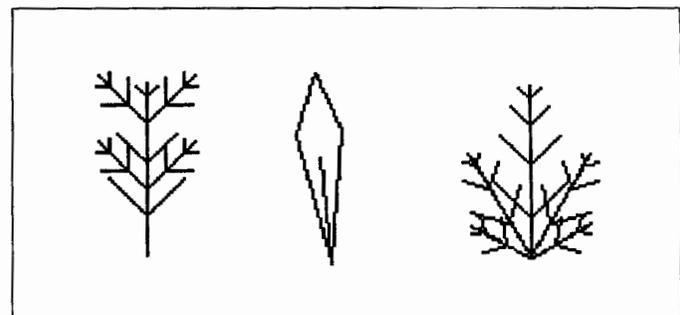
Fahrenheit. Plate crystals form in air temperatures of 14 to -4 degrees. Column crystals are manufactured in clouds with few water molecules, 13-14 degrees Fahrenheit. The familiar dendritic shapes (stars and ferns) develop in clouds with moderate amounts of water vapor, where the temperatures vary between 0 to 7 degrees.

The snow crystal comes into the world as a microscopic ice germ. Water vapor molecules attach themselves to a speck of dust, and condense and multiply for about 15 minutes. For the next 15-30 minutes, the snow crystal has a rocky ride as it falls through the storm cloud. Passing temperature and humidity conditions help to form the crystal's unique size, shape and design. Soon the ice crystal becomes too heavy to be supported by the cloud, and it joins the other acrobats as they tumble to the earth.

Snowgo

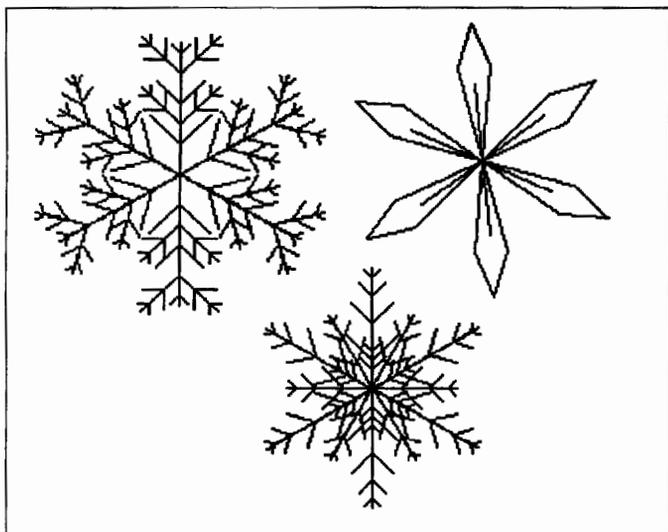
Snow crystals take their forms from their environments. How often have you seen paper snowflakes blanketing the windows of elementary classrooms? These special crystals owe their structure to the scissors and imaginations of their young creators. Logo environments can yield snow crystals, also.

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:

*NO STORM — CONTINUED



All of these stellar crystals were formed with the STELLAR superprocedure:

```
TO STELLAR
  MAKE "BRANCH.CHOICE PICK BRANCHLIST
  REPEAT 6 [RUN (LIST:BRANCH.CHOICE)→
    RIGHT 60]
END
TO PICK :LIST
  OUTPUT ITEM (1 + RANDOM (COUNT :LIST))→
    :LIST
END
TO BRANCHLIST
  OUTPUT [BRANCH1 BRANCH2 BRANCH3]
END
```

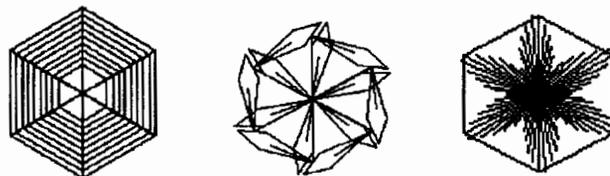
The PLATE superprocedure is similarly structured.

```
TO PLATE
  MAKE "SECTION.CHOICE PICK SECTLIST
  REPEAT 6 [RUN :SECTION.CHOICE RIGHT 60]
END
TO SECTLIST
  OUTPUT [[SECTION1 20] [SECTION2 40]→
    [SECTION3]]
END
```

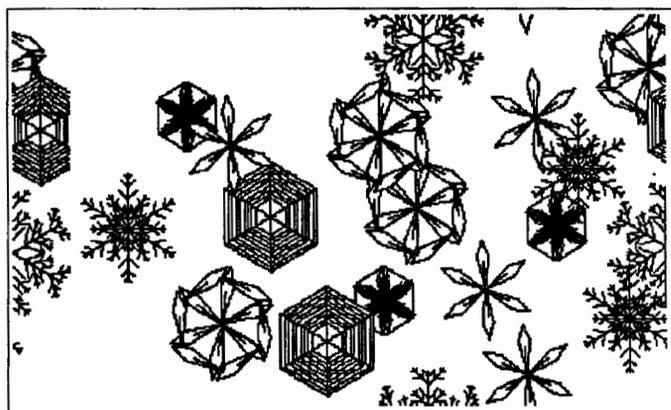
Pictures of hexagonal plate snow crystals can be built from triangular sections that are embellished in a variety of ways.



When combined, six equilateral triangles arranged around a common point can produce a variety of frosty effects.

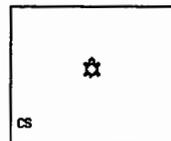


One might even want to create blizzard conditions on the CRT.



```
TO SNOWSTORM
  IF KEYP [STOP]
  RUN PICK [[STELLAR] [PLATE]]
  PU
  RIGHT RANDOM 360
  FORWARD RANDOM 200
  PD
  SNOWSTORM
  CT
  PRINT [BLIZZARD CONDITIONS]
END
```

And unlike the "real stuff," this snow can be cleared with one swift CS (computer shovel).



*"A snowdrift is a beautiful thing—
if it doesn't lie across the path you have to shovel
or block the road that leads to your destination."
—Hal Borland, 1964*

Judi Harris, 621-F Madison Ave., Charlottesville,
VA 22903; CompuServe: 75116,1207

LXIONARY

by Bill Craig

This month's column continues the response of the academic community to Seymour Papert's paper on technocentric thinking. The responses come from Roy Pea and Henry Becker. Much of Papert's paper on technocentric thinking and his assertions on the ways in which Logo uses should be evaluated were in response to the results of Pea's research findings on Logo use at the Bank Street School. This research failed to find transfer of problem solving skills from Logo to non-programming problems. Henry Becker is well known for his work with instructional computing in elementary and secondary schools.

"The Aims of Software Criticism: Reply to Professor Papert." by Roy D. Pea, Educational Researcher, June-July, 1987.

Pea supports Papert's intent to develop a discipline of computer criticism with goals similar to literary criticism and is optimistic that the field holds the potential for useful products. But that is about as much support as Papert gets from Pea. Papert's description of computer criticism fails to provide criteria for evaluating the growth of a computer culture. Pea describes other inconsistencies between the discipline of literary criticism and Papert's description of computer criticism. He even injects a doubt that very many people think in the technocentric manner that Papert is attacking. While rejecting computer criticism, Pea defends the use of experimental research as at least a means of defining the limits of Logo's utility to education.

Pea also uses this paper to defend the methods of the Bank Street Logo research. Contrary to popular perception, Pea claims that the teachers involved in the study were well trained and committed to using Logo with their students. The environment at the Bank Street School is consistent with many of the ideas expressed in *Mindstorms*. At the time of the study, there was a relatively large number of computers in the school. Pea also points out that the research did not end with this one study.

Pea concludes by claiming that no educational technology deserves "the same free reins that Papert recommends for Logo...Responsibility for designing the future of education cannot rest in any single vision."

"The Importance of a Methodology That Maximizes Feasibility: Its Applicability to Research About Logo." by Henry Jay Becker, Educational Researcher, June-July, 1987.

Becker begins by summarizing the characteristics of Logo users and finds that most users are teachers of students in kindergarten through fifth grade. These teachers are using Logo primarily for enrichment and with high ability students. Becker also found that "most classrooms that use Logo appear to be rather traditional instructional environments."

Becker believes that computer criticism as described by Papert has three steps: observing students while they use Logo, asking them questions about what they did, and making inferences about their intellectual growth. He then describes a design for research that he believes would provide a fair measure of the effects of Logo's use.

Becker concedes that there will not be one uniform effect for any teaching practice on all students. But he defends the necessity to "generalize about consequences for a range of students under a range of circumstances." If a practice is going to be effective in only unique situations, it is important that educators and policymakers know that.

Becker and Pea, and many others, are asking the Logo community to substitute empirical data for the anecdotal data it has been providing. The fact that this same request is coming from many corners is indication that the use of Logo has not yet achieved the credibility many of us hoped it would or assumed it had.

Bill Craig, Hening Elementary School, 5230 Chicora Drive, Richmond, VA 23234

Global News

Edited by:
Dennis Harper

Working Group: "Educational Informatics"
State University, Ghent, Belgium

The working group EDucational InFormatics is operating within the Seminar and Laboratory for Experimental, Psychological and Social Pedagogic of the State University of Ghent. The organization: "Computers in Primary Education" is the service organization of EDIF.

Target group

On one hand researchers in the field of the applied pedagogical-didactical research.

On the other hand, teachers in primary education, principals and their staff, inspectors, psychologists and pedagogues.

Aims

The aim is twofold, depending on the distinct target groups:

- (1) The activation, setting up and coordination of running research projects.
- (2) "Translation" of computer applications into practice in education context of primary school.

Activities

The working group was founded in January 1983 as a result of changes in the focus of researchers of the department. In 1977, collaborators of the department were already involved in research on pocket calculators in primary and secondary education. Since 1982, attention shifted towards the introduction of new information technologies in education. Activities have been diversifying at a growing rate. Considering the present situation, we can classify all former and running activities under one of the following headings:

- (1) scientific research
- (2) initiatives for in-service training
- (3) training of students of the R.U.G. (State University of Ghent)
- (4) collaboration with trade and industry
- (5) international contacts

In relation to all these activities, emphases is put on the educational use of Logo. Reports on the specific themes/topics are available.

(1) *Scientific research*

- 1.1. Logo as a learning environment for solving problems; long term research project with 8-12 year old pupils.
- 1.2. Identification of interventions by teachers to stimulate problem-solving thinking.
- 1.3. Integration of Logo into the mathematics curriculum in primary education; development of resource materials, a handbook and the evaluation of its use in a large scale evaluation project.
- 1.4. Word processors and primary school
- 1.5. Possible roles of the computer in spare-time corporate life.
- 1.6. Investigation into the actual situation of computer use in state primary schools in the provinces of East and West Flanders.
- 1.7. Description and evaluation of courseware.
- 1.8. Development of courseware packages; history simulations, learning environments (e.g. Logo and the teaching of writing skills)

(2) *Initiatives for inservice training*

- 2.1. Upon request, inservice training sessions are organized for small groups (at the University or residential).
- 2.2. Inservice training in the educational use of computers for pedagogues and psychologists.
- 2.3. Inservice training for teachers, lecturers, members of inspection and school principals in "Logo as a learning environment"

Terrapin Publication Aids Probability Learning

Terrapin's new Logo Ware publication titled *Logo Probability* offers hands-on Logo activities for teaching probability in middle school or high school mathematics classrooms. *Logo Probability* is designed for use with Terrapin Logo for the Apple or Commodore Logo. Its 12 activities allow students to explore counting techniques, sample sizes, dependent and independent events, combinations and permutations, fairness, graphing, and estimation. *Logo Probability* comes complete with a Student Booklet, Teacher's Guide, and Probability Disk. The Teacher's Guide includes reviews of both Logo and probability, teaching ideas and challenges for each activity, reproducible worksheets, and resource bibliography. The Probability Disk contains over 30 programs. The Student Booklet introduces each activity, provides helpful tips, and poses open-ended questions for investigation and discussion. A single package sells for \$23.95; a mini pak with eight Student Booklets sells for \$34.95 and a class pak with 25 Student Booklets sells for \$59.95. Please add \$2.50 for shipping and handling (add 5% sales tax if ordering from Massachusetts.). To order, contact Terrapin Inc., 376 Washington St., Malden, MA 02148; ph. 617/322-4800.

Global News — CONTINUED

- 2.4. Organization of instructional days and afternoon meetings covering specific subjects.
- (3) *Training of students R. U. G. (State University of Ghent)*
- 3.1. Students pedagogics: in relation to the course "Specific Methodology of Primary Education", there is a series of sessions about educational computer use.
 - 3.2. Students R. U. G.: course "Data processing on computer"
- (4) *Collaboration with trade and industry*
- 4.1. Adaptation of existing software: MSX Logo, LogoWriter, etc.
 - 4.2. Developments of software: course packages, simulations, programming environments, etc.
- (5) *Present international contacts*
- 5.1. Leadership of the Logo-project of the European Community (in which 10 member states are participating).
 - 5.2. Organization of the first, second, third and fourth Summer University of the European communities regarding new information technologies in education.
 - 5.3. Organization of IFIP working conferences in relation to the use of computers in primary schools.
 - 5.4. Co-organizer or organizer of conferences in Logo, e.g., the 1989 EUROLOGO conference

People interested in former or future activities of the working group can write to

Leader: Professor Dr. G. Schuyten
LOGO Department: Dr. M. Valcke
Address: State University of Ghent
 Department of Education
 H. Dunantlaan 1
 B9000 Ghent
 Belgium
 Phone: 91 25 41 00 ext. 354
 Telex: 12754 rugent b

Announcing...

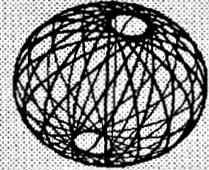
Summer Institutes and Logo Conferences

University of Virginia; Charlottesville, VA

Logo Connections:

A Two-week Retreat will be offered at the University of Virginia in Charlottesville, VA August 1-12. The intent of the retreat is to bring together teachers across the country who are interested in connecting with each other and learning how to create "Logo Connections."

From its beginning, Logo was designed as a language which could be connected to objects in the outside world, such as switches, motors, lights, and sensors. In addition, Logo work has always emphasized the importance of interpersonal learning through connections with colleagues and friends. Both of these aspects of connections will be emphasized at the retreat in an informal, hands-on environment.



The workshop faculty of Glen Bull, Gina Bull, Judi Harris, Tom Lough and Cheryl Wissock have a wide range of interests and expertise, ranging from pendula in physics to switch inputs for handicapped users. Among them, they have written numerous books and several hundred articles on Logo.

Registration for the two-week retreat will be limited to a maximum of twenty participants. Air conditioned dormitory rooms will be available at Summer rates for those who require housing. Three hours of graduate credit will also be available through the University, on an optional basis. A wide variety of social experiences throughout the two-week period will emphasize the importance of personal relationships in teaching with Logo.

Participants must have intermediate Logo programming skills prior to the retreat. For an application, write to: Logo Connections; c/o Tom Lough; Curry School of Education; University of Virginia; 405 Emmet Street; Charlottesville, VA 22903 or call Peggy Marshall at 804-924-7471.

Lesley College; Cambridge, Massachusetts

What are your plans for professional renewal this summer?

Consider The Logo Institute, August 7-21, 1988:

- Experience a two-week immersion in a learning environment for Logo-using educators;
- Use the full scope and power of Logo and its educational philosophy to create programming, research, teacher education and curriculum projects; and
- Receive three graduate credits from Lesley College, with three additional credits optional.

Participate with An Outstanding Faculty:

- Dan Watt and Molly Watt, *Codirectors*;
- Plus: Alison Birch, Sharon Burrowes Yoder, Ricky Carter, Dorothy Fitch, and Michael Tempel.
- Faculty will offer lectures and demonstrations, lead short courses, provide technical assistance and coaching, and support participants in completing their own creative projects.

Attend The Logo Institute Conference, August 20-21, 1988

- An intimate weekend conference to which the broader education community and participants in previous Logo Institutes are invited.
- Keynote speaker: Steve Ocko, codeveloper of Logo-Logo: Tracing the History of Technology Through Clocks, Automobiles and Robots.
- Speakers and workshop leaders include Logo Institute faculty and participants and leading Logo-using educators.
- Conference participants may register for one optional graduate credit.

For further information and registration materials, contact: Prof. Ricky Carter, The Logo Institute, Lesley College, 29 Everett St., Cambridge, MA 02138-2790; ph. 617/868-9600.

**University of Michigan
Ann Arbor, Michigan**

Co-Sponsored by
University of Michigan's Professional Development Office and
the Michigan Association for Computer Users in Learning (MACUL) SIGLOGO

June 27-30, 1988

Here's a description of the sessions:

June 27 Morning -- Introduction to Logo for Novices
Afternoon -- Presentations by Michigan-area teachers on using Logo in their classrooms.

June 29 More Logo Writer (on using LogoWriter music, animation, and linking multiple scrapbook pages together in one project -- some Logo-Writer experience required)

June 28 Introduction to LogoWriter (no Logo experience necessary)

June 30 Morning -- Introduction to Logo TC Logo
Afternoon -- The Phantom Fish Tank and Other Logo Ideas

The prices for the sessions are:

Half Day	\$20
One Day	\$40
Two Days	\$70
Three Days	\$90
Four Days	\$100

For more information, Contact :

Christine Canning; Professional Development Office; 1225 School of Education; The University of Michigan; Ann Arbor, Michigan 48109-1259; 313/763-9497

British Logo User Group Conference 1988

This year's conference of the British Logo User Group will be held in the College of Saint Paul and Saint Mary, Cheltenham, from Friday 2 September to Sunday 4 September, 1988.

This conference is intended for primary and secondary school teachers, advisory teachers, teacher trainers, researchers in education and anyone else interested in the use of micromcomputers in schools, in particular in the Logo programming language.

This conference will include a series of activities for Beginners in Logo, and a chance to hear about teachers' experiences in their classrooms. There will be a trade exhibition.

Conference fees:

	Resident	Non-resident	Sat. only
Members	£80	£55	£25
Non-members	£90	£65	£35

For bookings or further details write to:

The British Logo User Group; P.O. Box 79; WALSALL, West Midlands; WS5 3RW

Great Lakes/East Coast Logo Conference

Cleveland, OH
May 5-7, 1988

Educational Computer Consortium of Ohio (ECCO)

with Cleveland State University, University of Virginia
Curry School of Education, John Carroll University and
Kent State University.

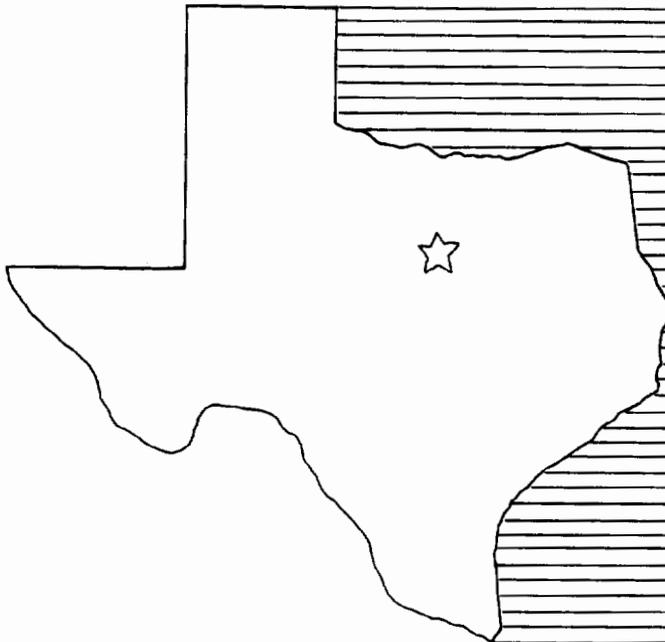
Three days of preconference workshops, panels and
break-out sessions. Join Mike Battista, Alison Birch, Sharon
Burrowes, Doug Clements, Brian Harvey, Tom Lough,
Steve Ocko, Seymour Papert, Brian Silverman, Michael
Tempel, Dan Watt, Molly Watt, Sylvia Weir, and many
more.

Contact ECCO, 1123 S.O.M. Center Road, Mayfield
Heights, Ohio 44124; ph. 216/461-0800.

**West Coast Logo Conference Presents
Hands on Technology**

Los Angeles Hilton Hotel, February 3-4, 1989

Mark your calendar now and look for more information
in future issues.



NECC '88

June 15-17, 1988
Loews Anatole Hotel
Dallas, Texas

8th National Educational Computing Conference

INVITES YOU TO ATTEND

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

For information contact:

**Jim Polrot
NECC '88
P. O. Box 5155
Denton, TX 76203-5155**

For information on Exhibits contact:

**Paul Katz
Continuation Center
University of Oregon
Eugene, OR 97403**

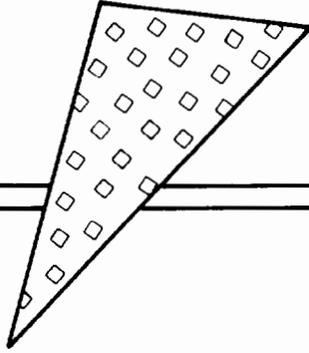
NECC '88 is sponsored by the International Council for Computers in Education (ICCE) in cooperation with member organizations of the NECC Steering Committee:

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

Join the Leading Professional Organization for Computer Educators!

Whether you are an administrator, curriculum specialist, classroom teacher, media specialist or special educator, you will benefit from membership in ICCE.

ICCE is the leading U.S. and international professional organization for computer educators. It is non-profit, supported by more than 50 organizations of computer-using educators worldwide.



Membership in ICCE includes a subscription to *The Computing Teacher* journal. The journal has long been respected as an important source of information for computer educators, providing accurate, responsible and innovative information. Nine times per year, its articles, departments and reviews keep you in touch with the constant changes taking place in educational technology.

Members will also receive discounts on all ICCE publications and SIG memberships, and information on ICCE committee activities.

As educational technology continues to grow and change, look to ICCE and *The Computing Teacher* journal to keep you on the leading edge.

Join today!

ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403.
Ph: 503/686-4414.

