

Twenty Powerful Ideas

by
Molly Watt

When teachers know of the powerful ideas embedded in Logo, they can teach more effectively. Powerful ideas influence how they introduce Logo, how they decide which examples to show, and how they establish the context and culture for their students. When teachers know and demonstrate more than just a knowledge of the Logo language and its syntax, students are able to use Logo more powerfully and intentionally.

I hope that this set of twenty ideas will give you some things to consider as you begin a new school year.

1. The Turtle. Everyone knows how to move their own bodies. By using that knowledge and identifying with the turtle, everyone can tell the turtle how to move across the graphics screen. By teaching the turtle, we learn how to program in Logo.

2. Playing Turtle and Playing Computer. Pretend to be the turtle and walk the design you want the turtle to draw on the screen. Observe carefully your steps, your directions, your turns. Translate your steps and turns into Logo commands. This is a good programming tool and an excellent debugging tool.

3. The Total Turtle Trip Theorem. Whenever the turtle has drawn a figure and returned to its starting location and heading, it has turned a total of 360 degrees (or some multiple) during the Total Turtle Trip. This idea contains information needed to draw any regular shape, such as a square (4 times 90) or a triangle (3 times 120).

4. Drawing. Almost everyone uses drawing as a natural way to express ideas. With Logo, you are using the turtle to draw and to express your ideas.

5. Exploration and Discovery. Every Logo user is an explorer and can make genuine discoveries of how to draw a shape or solve a problem in its rich mathematical environment. A successful explorer must be willing to try ideas and see what happens.

6. Theory Building and Problem Solving. Logo learners make up theories about everything, especially about turtle moves. Taking note of their specific expectations and of the differences between them and what actually happened invites students to develop workable problem solving strategies.

7. Describing and Defining. All Logo procedures are descriptions of something you want Logo to do. You can save these descriptions as procedures and use them over and over. Also, describing and defining a problem is useful in debugging or solving that problem.

8. Debugging. Trapping and eliminating bugs is a great deal of what Logo programming is about. (Ed. note: See NLXual Challenges elsewhere in this issue.) Hardly anyone gets a Logo program to work the way they want the first time through. This is the problem solving "curriculum" built into Logo. Logo error messages are helpful in finding bugs.

9. Procedures. When doing routine jobs, most of us follow a series of small procedures. A procedure does one small task in life, as well as in Logo. Procedures are Logo's helpers. They can be combined to work together with a superprocedure as a boss. Procedure trees show how procedures work together and in what order they are called.

Powerful Ideas continued

10. Naming. Everyone loves to name things. In Logo, you name anything you want to save or use again. Naming helps you remember what you did and allows you to use something again by calling its name. Also, naming is an important part of problem solving.

11. Extensibility. You can teach Logo how to do something by defining a procedure. This extends the language.

12. Variables. A variable is a piece of information. It could be thought of as a place to put a number (or other object) as input to a command or operation. Numerical variables allow you to create the same shapes in many different sizes on the screen, thus giving you a great deal of power in using small building blocks.

13. Repeating. Logo's REPEAT command allows you to repeat a procedure or any list of instructions a number of times. With REPEAT, you can explore patterns, create tessellations, and discover mathematical relationships between numbers or variables.

14. Recursion. A Logo procedure can call another procedure by the same name (a copy of itself) as a way to get a job done simply. The idea of recursion is one of the most powerful ideas embedded in Logo, and it is not easy to grasp. Using recursive procedures to describe actions in real life is helpful to students. For example, let's ask Dan to clear the dinner table and wash the dishes. Using recursive thinking, Dan would check to see if there were any dirty dishes on the table. If not, his job is done. If there were, he would clear his own place and wash his own dishes, then call a helper with the same set of instructions. With such examples, students can grow into the more complex versions of recursion and come to use Logo's full power as an artificial intelligence language.

15. Conditionals. Logo can stop procedures when certain conditions are met. For example, IF HEADING = 0 [STOP] might be used to stop a recursive procedure. Of course, conditionals have many other uses as well.

16. Interaction. Logo as a process is interactive, and invites students to learn at the computer by trying things out. More advanced programming allows users to input information into programs with commands such as READCHAR, REQUEST, and READLIST.

17. Information Processing. Logo stores information as words and lists. Naming files and procedures and using REPEAT and PRINT are simple examples of processing words and lists. Logo can also take information input as words and lists, operate on it, and give the result as output.

18. Thinking About Learning and Learning About Thinking. Part of learning Logo is observing your own thought processes made visible on the screen, often in the form of turtle tracks. By watching the turtle's heading, you can see your own thought process in

estimating angles. By describing it in words, you can refine your thinking about angles.

19. Microworlds. Logo makes it possible to establish a safe, defined learning environment where students can do some real learning through open ended exploration. Teachers become guides and challengers, rather than lecturers.

20. Modes of the Computer. Logo has two main modes: direct (or command) mode, and edit mode. When you are working in the direct mode and you type an instruction and press the "do it" key (sometimes called <return> or <enter>), something happens. In the edit mode, however, instructions are stored to be used later. Nothing happens when "do it" is pressed. Understanding the effect of each mode and being able to shift from one mode to the other is essential to learning to use Logo effectively.

(Ed. note: This set of ideas is the product of Molly and Dan Watt's thinking and current work in Logo. More detailed versions of this article appear in Molly's Welcome to Logo! series from D. C. Heath, and Molly and Dan's Teaching with Logo from Addison-Wesley.)

Molly Watt is a teacher, lecturer, and author in the field of educational computing.

Imagewriter Overprinting?

Are you frustrated by the Apple Imagewriter when trying to print listings of Logo procedures which have long lines? Glen Bull is actively seeking a way to prevent the overprinting which results. So far, no solution has been discovered. If you have ideas or suggestions, send them to Glen in care of the NLX. Meanwhile, here is a "clunky" fix.

1. Load in the file to be printed.
2. Bring the procedures into the editor. (Use EDIT ALL with MIT Logo or EDIT [SQUARE TRI MOVEetc.] with LCSi.)
3. Locate any instruction lines which take up more than a single line in the editor.
4. Move the cursor to the middle of the second editor line of the long instruction line, type ! and press <return>. This has the effect of "breaking" the line, as far as the printer is concerned.
5. Repeat step 4 as often as necessary, until all long lines are "broken."
6. Define the procedures by leaving the editor appropriately.
7. Print out the file. The Imagewriter will print the "broken" lines with an exclamation point at the end of each to indicate that they are actually continued.
8. Erase the file. It will not run properly with the "broken" lines. If you want to run procedures in the file, load them from the disk once more.

Tipps continued

Investigate which part of the pattern is drawn and which part is "skipped." Some children may have a hard time making this distinction. You can help by posing questions about the line and the skips. Which part makes the dash? Which part makes the space?

The social interaction, discussion, and argument between children in matching lines with REPEAT patterns are excellent means for learning that the turtle has two different functions. Seeing the turtle as a drawing tool and as a position mover is important as children gain mastery in Logo. It is quite possible that your children will define the space as the DASH, because the space interrupts the line and makes it dashed. Eventually this may get in the way of DASH projects, but let them work through that assumption.

Another observation may be that the line made by certain combinations of PENUP (PU), PENDOWN (PD), and FORWARD (FD) are not "accurate." For instance, PU FD 10 PD FD 10 makes a line which may not be even. The dash made by FD 10 may be longer than the space made by FD 10. This is a result of what I call "pixel capture." The lighted pixels of picture elements on the monitor sometimes capture nearby pixels and turn them on as well. While this is not a very technical explanation, most children will be familiar with the idea that a light bulb in a dark room also lights up the area around it.

DASHING COMMANDS

Rather than making a procedure of an entire line or repeat statement, use a single movement as the basis of a procedure. Writing a DASH procedure is an easy but important task, when familiar with PU and PD. After working with the dashed line, children may decide that a dash is about 5 turtle steps long. They can capture this idea in a DASH procedure. In fact, DASH may be a very good procedure to introduce the concept of creating your own commands. It is simple, functional, and modular. By making DASH the drawing part of a pattern, many lines are possible.

```
TO DASH
PENDOWN FD 5 PENUP
END
```

```
REPEAT 10 [ DASH FD 20]
REPEAT 10 [ DASH DASH DASH FD 10]
REPEAT 10 [ DASH FD 10 DASH DASH FD 10]
```

A length of 5 may be too big for some tastes, so that 2 or 3 is preferred. Different dashes may be better for different projects. The variability of the dash size is something which children can see. A variable DASH may be just what the turtle ordered. If children invent the need for variability, the variable form is amazingly simple for them to grasp, and incredibly versatile as well.

```
TO DASH2 :LENGTH
PENOWN FD :LENGTH PENUP
END
```

Now REPEAT statements can be written with the plain DASH or a variable DASH.

```
REPEAT 10 [ DASH DASH FD 10 DASH FD 20 ]
or variably
REPEAT 10 [ DASH2 10 FD 10 DASH2 5 FD 20]
```

Some projects don't work quite as well if the displacement of the turtle is 2 or 3 turtle steps. A third DASH procedure returns the turtle to its original position.

```
TO DASH3 :LENGTH
PENDOWN FD :LENGTH BK :LENGTH PENUP
END
```

The projects which children invent determine what type of DASH is most useful. I found that DASH, DASH2, and DASH3 were equally useful to me and allowed me to do different things. Multiple DASH procedures extend the understanding of procedural thinking. Executing several commands by creating and naming a procedure is powerful; using this power is one key to unlocking Logo.

DASHING SHAPES

Dashes can be used with familiar shapes. In fact, children who already "know" square, triangle, and circle may have to rethink their ideas if they use DASH to complete the figures.

Think about the differences between

```
REPEAT 4 [ DASH FORWARD 50 RIGHT 90]
and
REPEAT 4 [ FORWARD 50 RIGHT 90 DASH2 5]
and
REPEAT 4 [ FORWARD 50 DASH3 5 RIGHT 90]
```

Part of the answer lies with the kind of DASH which is defined. But part also lies with the order of

Logo Teacher Feature by Rebecca Poplin

Featuring: Robs Muir

Challenges are a vital part of Robs Muir's life. NLX readers are familiar with some of the Logo challenges which Robs incorporates into his instructional environment. These NLXual Challenges are a natural outgrowth of one realm of activities in which Robs is involved. Robs and his wife Candi teach fifth and sixth graders in a small elementary school in Claremont, California. Their students are actively involved in solving structured problems with many possible solutions.

A critical phase of the problem-solving is sharing the solutions with others. Robs has placed increasing emphasis on the sharing phase of problem-solving as he has worked more with Logo and has realized the benefits of this process.

Two of the most exciting problems last year produced a Logo program which created a word search puzzle and several versions of "Stationmaster," a favorite indoor game. (Ed. note: See NLXual Challenges in the February 1985 NLX.) Problems and solutions are shared within the classroom and with NLX readers all over the United States and in other countries.

Robs has recently taken on a new challenge. He has become the coordinator of a project supported by the Apple Education Foundation, in which the Logo Conference Tree (a public telecommunications system for sharing information and ideas about Logo) was established. Ten school districts in southern California are linked through this project. Students may use the service from their schools or homes. NLX readers are welcome to contact the Logo Conference Tree (300 baud) at (714) 621-2310. No identification or password is needed. Most of the NLXual Challenges are available through this service.

Robs also teaches at the Claremont Graduate School, where Logo is used extensively in teacher education and for teaching data structures in graduate computer science programs. He is also participating in planning the West Coast Logo Conference, scheduled for November 21-23, 1985.

In the coming school year, Robs is looking forward to the new challenges he has set for himself and his students. He expects to involve students more extensively in telecommunications. A persistent problem has been how to fit Logo into a time schedule in which certain allocations are mandated for various curriculum areas. As a result, Robs is interested in greater integration of Logo and curriculum areas, particularly science.

Teaching with Logo is a growing and learning process. Robs Muir conveys a sense of developing perspective, as he has modified his approach and use of Logo over time. In the past, Robs felt that every student should program in Logo, and his approach to computer instruction and use was more prescriptive than it is today. Challenge problems are shared, and many students participate in solving and creating them, but the computer is used in a variety of ways in addition to Logo.

Effective Logo teachers are constantly learning, thinking, and growing. They probe the environment for ways to stimulate creativity and growth with all the tools available, including the Logo language. Each month, this column will profile a teacher who is using Logo in his or her classroom on a regular basis. Readers are invited to suggest names of teachers who have developed an interesting or unique perspective on Logo. Please contact: Rebecca Poplin, 2421 Fain Street, Wichita Falls, Texas, 76308, or call (817) 766-5323.

Rebecca Poplin uses Logo to teach computing and mathematics at a junior high school in Wichita Falls, TX.

NLXionary A Lectionary of Selected Logo Readings with Commentary and Opinion by Griff Wigley

"Disorganized Students Do Better with Computers, Says Logo Creator." Conference Report, Education Computer News, June 19, 1985.

Seymour Papert sparked more controversy this summer at a Tufts University symposium on using computers in school science labs. The symposium was sponsored by the Technical Education Research Center (TERC) based in Cambridge, MA.

Papert reportedly argued that a systematic step-by-step approach to programming is not best for all students; yes, even that some of the best programmers are non-planners who fiddle with a program until it works. Papert believes that "messing about" is particularly beneficial in science classes because students learn more through exploration than through being forced to prove theories. He added that writing without using an outline may not be always bad either.

There you go again, Seymour. Taking pot shots at the established methods of teaching that have a solid history of alienating generations of students. Haven't you been keeping up on all the latest in problem solving? What about all those thinking skill hierarchies,

NLXionary continued

matricies, and taxonomies that educators everywhere are transforming into handy-dandy scopes and sequences? If you insist, "messing about" can be covered under creative thinking, which is taught only after the problem solving strategy of working backwards is mastered (usually by the 4th grade in progressive districts).

How can a teacher allow a student to "mess about" with a science concept for days, weeks, or as long as it takes to "sink into the subsoil of the mind?" What about chapters? Units? Tests? Quarters / Semesters / Trimesters?

Seymour, you sound as if you've been reading "In Search of Excellence," as if Managment By Wandering Around (MBWA) should become Learning By Wandering Around (LBWA). Most schools / corporations want their students / employees focused on the grades / profits for this quarter so the parents / shareholders will be happy. "Messing about" / "Wandering around " might make it in the free enterprise system where a company has incentives and the customer has some power. But never in the monopoly of public education where incentives are few and student / parent customers have relatively little power. Unless...

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

Logo Disserts: Dissertations Dealing with Logo by Barbara Elias

Portuguese schools are now beginning to use microcomputers. Some preparatory and secondary schools now provide extra-curricular programming courses on a limited basis and a few pilot projects have been started. Recently, the Portuguese Ministry of Education formed a committee to work on ways to introduce computers into the schools. One important step in assuring an effective program is to prepare teachers to use this technology. A study has been conducted which presents the design of a Logo unit for prospective elementary teachers. It is offered as one means of helping to prepare teachers to teach Logo.

The study includes a literature review of Logo research and teacher computer education. Also included is a Logo course and an implementation and evaluation plan.

The literature summarized in this study suggests that (a) instruction in a programming language is an important part of course work which prepares teachers to deal with computer education, (b) Logo can provide valuable learning experiences for children, but the claims regarding logical thinking and problem solving skills remain inconclusive, and (c) Logo research in the area of teacher preparation is sparse and inconclusive.

The unit design includes a description of the learners who will use this unit, the content of the course syllabus, and resources and conditions desirable for implementation.

Learners expected to use this unit are first year college students, 18 to 21 years old. Most of them are female, and come from a variety of backgrounds. It is assumed that they are capable students who recognize the importance of computer education. The proposed program goals are the development of (1) Logo programming mastery, (2) positive attitudes towards mathematics and the teaching of mathematics, and (3) problem solving skills.

The proposed syllabus contains a wealth of information. Section I includes an introduction to Logo, turtle graphics, and workspace and file management. Section II presents advanced programming concepts, such as recursion, and a brief introduction to list processing.

Learning activities and teaching strategies are included in lesson plans organized around the program goals. Though fairly structured, they are intended for use with freedom and flexibility in a manner which helps to create a Logo learning environment.

Formative and summative evaluation measures are included. The researcher commends this study for consideration by Portuguese officials as one possible way of carrying out two important tasks now needed: curriculum development and research.

>>>Moreira, Candida Maria De Almeida Pauperio Queiroz (1984). Logo: Design of a Unit for a Course in Computers in Education for Portuguese Student Teachers of Escolas Superiores De Educacao. (Unpublished thesis, Master of Education, Boston University, 1984.)

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Listful Thinking

by Glen Bull and Paula Cochran

Blending Logo and Language Arts

Welcome back! For new readers who are joining us, there are two tenets which underlie this column:

(1) We are more interested in using Logo to teach than in teaching Logo. Therefore the focus of the column is on content areas such as English rather than on Logo programming.

(2) We think Logo tools are essential for allowing the teacher to focus on content rather than programming. A Logo tool is a procedure which can be used by the teacher and by students as if it were built into the language. (Logo is one of the few programming languages which allows this to be done easily.)

At the beginning of school, an essay on "What I Did Last Summer" is almost obligatory. This summer we spent some time working with teachers at Punahou School on the island of Oahu in Hawaii.

We had fun exploring Oahu by car. Quickly we noticed that the street names all seemed to begin with K or H, and were about 40 syllables long with names like "Kanehamenamehahahakelike".

We tried different strategies. The navigator would tell the driver, "In about two blocks, turn to the right at 'Kao .. oo??' ." The driver would respond, "What?". Then the spelling strategy was invoked: "The first four letters are 'K A N E' and the last letter is also an E". We began to wonder, "Why are all the names so long?"

THE TOOL PHILOSOPHY

It turns out that the Hawaiian language has an interesting property. It has fewer sounds than most other languages. Since there are fewer sounds in Hawaiian, fewer combinations are possible. To create new words using the same sounds, the words must be longer. We wondered if we could write a Logo procedure that generates Hawaiian-like words. This led eventually to some explorations of English.

There are only six or seven consonants in Hawaiian. Here are the ones we could remember.

```
TO CONSONANTS
OUTPUT [P L K M N H W]
END
```

Now when we wanted to see what the Hawaiian consonants were, we simply typed:

```
?PRINT CONSONANTS
P L K M N H W
```

(Throughout the column, the part you type is shown in boldface.)

The vowels in Hawaiian are the same as English.

```
TO VOWELS
OUTPUT [A E I O U]
END
```

We can look at the vowels in the same way.

```
?PRINT VOWELS
A E I O U
```

We wanted to create a Logo procedure that would pick Hawaiian consonants and put them together with Hawaiian vowels to make Hawaiian syllables. Readers from last year will remember a common Logo tool that picks items from a list.

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

(In most versions of Logo, ITEM and COUNT are built-in. If you use version 1 of Terrapin Logo, you will need to use the procedures at the end of the column.)

Reflecting our tool philosophy, you don't have to understand what happens inside PICK. Try using it this way:

```
?PRINT PICK [X Y Z]
Y
```

```
?PRINT PICK [X Y Z]
Z
```

The procedure PICK chooses random items from a list, so you may have gotten different letters.

Now we can pick a Hawaiian consonant and a vowel.

```
?PRINT PICK CONSONANTS
```

```
M
?PRINT PICK VOWELS
U
```

Listful Thinking continued

?PRINT PICK CONSONANTS

H

?PRINT PICK VOWELS

I

If we put the letters that we picked together, it creates a Hawaiian word of sorts: MUHI.

This probably isn't a real Hawaiian word, but it looks and sounds like one. This word has a pattern. It consists of a consonant followed by a vowel followed by another consonant followed by another vowel. Linguists have a convention in which they use the letter C to stand for a consonant and the letter V to stand for any vowel. The pattern could be represented in this way.

Consonant Vowel Consonant Vowel

C V C V

M U H I

We are going to represent a Hawaiian consonant with C and a vowel with V. We can do this with Logo procedures that work like this.

```
TO C
OUTPUT PICK CONSONANTS
END
```

```
TO V
OUTPUT PICK VOWELS
END
```

Now we can print a consonant or a vowel by typing:

```
?PRINT C
H
?PRINT V
A
?PRINT C
M
?PRINT V
A
```

This makes another word that looks like it might be Hawaiian: HAMA.

Until now, we have had to put the separate letters together to see what the word looks like. There is a primitive procedure (WORD) built into Logo which will put the letters together for you.

```
?PRINT ( WORD C V C V )
KIHU
```

Now that we have a debugged Hawaiian word generator, we can turn it loose!

```
?PRINT ( WORD C V C V C V C V )
HAMA HANOHI
```

SPELLING LESSON

Thinking about Hawaiian and its permissible sound combinations made us start thinking about sound combinations in English. For example, English words, unlike Hawaiian, frequently begin with consonant blends or clusters.

Examples of blends which occur frequently in English (and therefore in spelling lessons) are BR, SL, ST, PL, etc. Can we adapt our Hawaiian word "generator" for English?

First, let's change the consonants to include sounds that often occur in English blends (especially R, L, and S). Edit your CONSONANTS procedure to look like this:

```
TO CONSONANTS
OUTPUT [ P L R S T B D W ]
END
```

To print consonant blends, try the following:

```
?PRINT ( WORD C C V )
BTA
```

```
?PRINT ( WORD C C V )
STO
```

AGH! Some of these combinations are not possible in English!

Even without formally studying the syllable structures and sound combinations of English, native speakers have knowledge about such things. You know which of the samples above are well-formed without reference to explicit rules. An exercise which emphasizes the consonant blends of English can take advantage of the intuitions your students also have about language. The goal is to make them more aware of their intuitive knowledge of English. This can make "rules" seem more obvious and easy to use.

Your students could test some randomly generated CCV combinations, using their own intuitions and a worksheet from you for guidance:

COMBINATION	ENGLISH BLEND?
_____	YES or NO.
_____	YES or NO

Listful Thinking continued

They should type:

```
?PRINT ( WORD C C V )
```

to get combinations to write on the worksheet. (Be sure to tell them about the control key for your version of Logo that retypes the line, so they only need to type the line the first time.)

Younger children may have trouble typing PRINT (WORD C C V) without errors. You can make a procedure for them if you wish.

```
TO BLEND
PRINT ( WORD C C V )
END
```

There is both an advantage and a disadvantage to creating a BLEND procedure. The advantage is that the teacher doesn't have to discuss the convention of using C and V to represent consonants and vowels. However, if students are given a BLEND procedure, they are less likely to experiment with other combinations such as:

```
?PRINT ( WORD C C C V )
or
?PRINT ( WORD C C V C )
```

Students can choose a couple of combinations that could be words in English. They can create definitions for the ones that sound interesting. For example, PRU: The sand left in the bottom of your sandals after a trip to the beach.

After several samples are generated, you might issue a challenge. Can your students change the letters in the CONSONANTS procedure so that no blends are produced? For example, we tried changing the CONSONANTS to:

```
TO CONSONANTS
OUTPUT [P T B N X Q]
END
```

This change caused combinations like this to be generated:

```
?PRINT ( WORD C C V )
XQA
?PRINT ( WORD C C V )
BNE
```

Can your class observe a rule about consonant blends in English? What sounds occur most frequently in blends? What is the maximum number of consecutive

consonants which can begin a word? What will those consonants be? Try a P or a T in the blank spots: S _ L or S _ R.

We hope you will find that Logo blends beautifully with language arts and the exploration of language. Aloha!

Procedures for Version 1
of Terrapin Logo

```
TO COUNT :LIST
OP COUNT.LOOP :LIST 0
END
```

```
TO COUNT.LOOP :LIST NUMBER
IF :LIST = [ ] OP :NUMBER
OP COUNT.LOOP BF :LIST :NUMBER + 1
END
```

```
TO ITEM :NUMBER :LIST
OP ITEM.LOOP :NUMBER :LIST 1
END
```

```
TO ITEM.LOOP :NUMBER :LIST :COUNT
IF :LIST = [ ] OP [ ]
IF :NUMBER = :COUNT OP FIRST :LIST
OP ITEM.LOOP :NUMBER BF :LIST :COUNT + 1
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

ILX Announced

Posy Publications announces the establishment of the International Logo Exchange (ILX) newsletter! Scheduled to begin publication in January 1986, the ILX is designed for those who want to keep abreast of Logo activities from an international perspective. Continental field editors will write columns for each ILX issue, highlighting recent Logo activity in their areas. Each issue will also contain notices of conferences, trips, and other activities of international scope. Editor of the bimonthly publication will be Dennis Harper, of the University of California at Santa Barbara.

A subscription to the ILX costs \$19.50 per year for addresses in the USA, Canada, and Mexico, and \$24.50 per year elsewhere. The International Logo Exchange, PO Box 5686, Charlottesville, VA 22905.

Turtle Tips

by
Jane Toth and
Donna Lanyi

Breathe fire into the Logo activities in your classroom as the year begins, and get your students off to a "hot start" in programming! Here are more Logo activities, ideas, and procedures to help motivate your students.

The Turtle Teacher

Perhaps your students worked with Logo last year. But how much do they remember? As in all subject areas, student retention tends to slip over the summer months. We have found that the beginning of the school year is an excellent time to use pre-computer activities with students to help their recall and gain further mastery of concepts. We work many of these activities into other subject areas, too.

Often, computer lab instructors wish they had more time for these activities. Why not a partnership between classroom teacher and computer teacher to accomplish this? Design appropriate Logo activities, games, or tasks that can be done during math, reading, or other subjects. After all, teachers have always been expert at creating games to enhance a concept. Here are some ideas based on other popular games you probably have used before.

Turtle Hunt is a variation of Treasure Hunt. Have students work in small groups to design turtle maps with directions to secret locations given in Logo commands only. After the maps are completed, have groups trade maps and try to locate the hidden "turtle treasure" (prizes or treats).

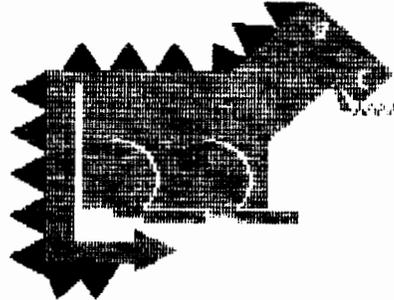
Bingo, Concentration, and popular board games can be revised to include Logo commands. Turtle Bingo cards containing Logo commands could be used with turtle shaped markers, for example.

The Yuppies have us all intrigued with the game of Trivia Pursuit. How about a Turtle Challenge board game along the same line? When presented with questions about Logo, students accumulate points or move turtle markers according to their answers, trying to "go home."

Procedure of the Month

Use the dragon's fire to ignite the imagination of your students! They enjoy designing and drawing all sorts of fantastic creatures with Logo, learning as they go. This dragon "starter" picture was submitted by Pat

Arnold, a first grade teacher here in Wooster. If you are interested in a listing of this program, please send us a self addressed stamped envelope.



Last year, we were overwhelmed by your excitement over the "Procedures of the Month" and all the possible applications. We thank you for your responses, and invite you and your students to send us ideas and procedures for use in this column.

Jane Toth and Donna Lanyi, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Jane Toth and Donna Lanyi use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Teacher to Teacher A Review of New Logo Materials by Anne Cairns Federlein

Fall is here and, with the new school year, Logo teachers discover they have a greater choice of publications to use as supplements to their Logo curricula. There are still the plethora of books addressing the needs of beginning students. However, some authors have developed texts that take advanced students and teachers into new and unexplored territories.

In order to assist teachers with all levels of students, this column will review a wide range of publications each month and define their strengths. These new and exciting materials offer a guide for facilitating learning through discovery in the classroom and the establishment and nurturing of exciting learning environments.

Teacher to Teacher continued

Beginning Students

The Workbook for Learning Logo, by Annie Kwok, Robert Sones, and Leon Hochglaube, is published by Turtle Publishing, a division of Art-I-Zan Specialities, Inc., 4810 Mariette Avenue, Montreal, Canada H4V 2G1.

Target Audience: elementary age children

Content: The workbook includes chapters on Getting Started, Meeting the Turtle, Screen Modes, Colors, and Procedures. It is designed to be used with a workbook diskette available from the publisher for \$15 (which, by the way, includes several utility procedures in addition to Logo programs for the book).

Strengths: The text is written in an easy to understand style and encourages both early and later elementary children to make discoveries on their own. The illustrations and text size are very appealing to both adult and child, and offer a change from written instructions. The sequential format allows the teacher to add creative classroom activities at each discovery in Logo. There are also blank pages and wide margins on each page to encourage the children and teacher to take notes on their work in each chapter. A cardboard turtle and turtle board game are included to help introduce the concept of turns and degrees.

Advanced Students

Beyond Mindstorms: Teaching with IBM Logo, by Joyce Tobias, Jerry Short, Sharon Burrowes, and Tom Lough, published by CBS College Publishing, Holt, Rinehart and Winston, 383 Madison Avenue, New York, NY 10017.

Target Audience: Logo teachers

Content: The title indicates that this book goes beyond Seymour Papert's Mindstorms "not in terms of ideas, but in terms of applications. It presents an approach designed to help teachers learn to use Logo and then to teach it to their students and to use it to enhance the curricula." The three divisions of the text include:

- (1) Welcome to Logo - an introduction and tour of IBM Logo,
- (2) IBM Programming - discusses graphics, non-graphics, the editor, pictures and words, saving work, subprocedures, random events, recursion, inputs to procedures, and global variables, and

(3) Beyond Programming - addresses creating a Logo environment, the Logo curriculum, teaching with Logo, working with special students, and the future of Logo.

Strengths: The value of this book is that it takes both teacher and student to a higher level of learning in Logo. Many of the publications on the market today only address the immediate technical and curricular concerns of the beginning student and teacher. These authors discuss the practical concerns of beginners and then proceed to integrating Logo into the school curriculum in different subject areas and grades, not just in the computer classes.

Once the practical aspects of Logo have been concretely presented, the authors take us to the philosophical level and outline the future of Logo. The Logo microworld offers important models for other educational environments and emphasizes the enjoyment of learning, the sharing of ideas, the importance of the gradual accumulation of fragments of knowledge, the discovery method of learning, and the acceptance of the outcomes during discovery learning.

Teachers could learn from their students the importance of "playing around" with ideas and learning from their mistakes. These authors should be commended for pushing us into a new era of teaching Logo ... one that affects the total learning environment of the child.

Do you know of any new Logo materials and books? Did you find them useful in teaching? If so, we invite you to send information to The Logo Information for Teachers (LIFT) Group, PO Box 5396, Plymouth, MI 48170.

Anne Cairns Federlein is a professor in the Early Childhood Area, School of Human and Educational Services, Oakland University, Rochester, MI.

Q and A

by
Jim McCauley

(Ed. note: Jim McCauley is in the process of moving to Eugene, OR, where he will pursue his Ph.D. under the supervision of David Moursund. His regular NLX column will resume with the October issue. In the meantime, you may send your Logo questions to him at 55 Sunset Drive, Eugene, OR 97403.)

Logo in the United States: A Survey by Joyce Tobias

During the past winter, survey forms were sent to members of the National Logo Exchange and the Logo 84 conference participants to learn more about the Logo community. The results of only the 366 respondents from the United States have been included in this report.

As in the results of the survey conducted by the National Logo Exchange in 1984 (Ed. note: see NLX September 1984, p. 10, or "Gradations of Logo" in The Computing Teacher, December 1984 / January 1985, p. 56), a wide majority of respondents use some form of Logo on the Apple computer (Terrapin, Krell, Apple, or Sprite Logo). Both Commodore and IBM have shown considerable gains since last year. Logo use with the Texas Instruments and Coleco computers is expected to diminish since those computers are no longer being manufactured.

Computers Used With Logo

77%	Apple
7%	Commodore
7%	IBM
5%	Atari
2%	Texas Instruments
2%	Coleco

The table below summarizes the indications of where Logo is being used. Not surprisingly, the Logo use is concentrated in the elementary schools, both public and private, with universities / colleges, and high schools gaining from last year. Logo use in the homes is also gaining in significance. A growing use of Logo in clubs is evident.

Responses in the "Other" category below include programs for the gifted and talented, special needs programs, summer camps, adult education, recreation departments, church groups, museums, hospitals, rehabilitation centers, and correctional institutions.

Where Logo Is Used

42%	Elementary Schools
17%	University / College
16%	Homes
14%	Secondary Schools
4%	Clubs
7%	Other

The table below indicates the response to the question, "How is Logo being used?" The "Other" category includes recreation, preservice and inservice training, logic, and artificial intelligence.

Logo Use

72%	Education
10%	Research
10%	Hobby
1%	Business
7%	Other

Of the respondents indicating educational use, 86% checked programming and 52% checked use as a tool to teach other subjects. (Many respondents checked both boxes.)

Subjects in which Logo is used as a tool included 62% math, 26% language arts, 20% problem solving, 11% critical thinking, 9% science, 8% social studies, 20% other (computer literacy, music, art, special education, and design). (Many respondents checked more than one subject.)

Some of the most difficult concepts to learn were reported to include list processing, recursion, integrating Logo into the curriculum, variables, file and workspace management, output, music, and language syntax.

When asked for the areas in which additional help was needed, the respondents included only list processing and integrating Logo into the curriculum from the difficult concepts above. Other topics included creating microworlds and applications, music, sprites, "everything!", and ideas for using Logo with adults.

Particularly helpful Logo materials cited included Learning with Logo (Dan Watt, McGraw-Hill), Apple Logo (Harold Abelson, McGraw-Hill), self made materials, and the National Logo Exchange newsletter. Others mentioned included Logo manuals, The Turtle's Sourcebook (Donna Bearden et al., Reston Publishing), Nudges (Steve Tipps et al., Holt, Rinehart and Winston), and Turtle Geometry (Harold Abelson and Andrea diSessa, MIT Press).

The survey raised a question for me. As I tabulated the responses, I noticed that many people mentioned "problem solving," others wrote down "critical thinking," and still others mentioned both. What is meant by these terms? How is Logo enhancing these skills? I would be interested in your comments and ideas on this topic. Please write to Joyce Tobias, 38 Maple Avenue, Lexington, MA 02173.

Joyce Tobias is an educational consultant and works with Microworlds Learning, Inc., in Cambridge, MA.

Logo 85 Report

The Logo 85 conference was held in late July at MIT. Copies of the Logo 85 Preproceedings and Theoretical Papers are available for \$15 each. Send a check (payable to MIT Logo Group) to Logo 85 Preproceedings, MIT AMT, E15-309, 20 Ames Street, Cambridge, MA 02139.

At the conclusion of the conference, Seymour Papert suggested that reflection on the subjective side of Logo is important and has not yet found an adequate place in the literature. He asked that teachers and researchers using Logo collect anecdotes and incidents and send them to him (at the above address) for a study of the range and diversity of human experience. Letters, outlines, plans, free associations, all are welcome.

Here are some comments on the conference from those who attended.

Kathleen Nutting:

Logo 85 provided an exciting opportunity to learn from a diversity of individual perspectives about Logo. I discovered that Logo's richness supercedes any one language as I interacted in an environment of individuals from all cultures. The people truly furthered a learning experience not only with Logo, the language, but with the process of change and growth in education, with honesty and risks, with issues in teacher training, with encouraging cultures and microworlds, and with enabling learning to be a process.

Logo, much as with any language, is a process of combining disjointed segments, words, and commands, to create a picture of reality. Seymour Papert challenged us to move the turtle from the flat screen into a practical reality that applies to life situations. He spoke of using Logo as a dynamic tool to settle questions that arise in classrooms and using Lego™ motors and kits in conjunction with the computer and Logo to provide a discovery-rich environment. Students may be guided in duplicating realities such as circles, and in creating new realities through their own visions. Horacio Reggini demonstrated such a vision through his presentation of a three-dimensional reality.

Logo 85 provided me with a broader perspective of the possibilities with Logo, new issues to ponder, and ideas and insights to share.

Ihor Charischak:

Though Logo 85 was a three day event, it was actually a three week experience for me. During the two weeks prior to the conference, I had the opportunity to participate in Seymour Papert's improvisational "Samba School." He invited inexperienced teachers from a local school, veteran teacher trainers from various parts of the world, elementary school students, and MIT faculty and students.

My environment consisted of people who presented ideas formally, my colleagues and friends who discussed the ideas with me informally, and the MIT culture. I soon found myself thinking about what it takes to make learning happen in places called schools, and how Logo can make a difference. Each new idea gave me insights into a new way of looking at this old problem. It became exciting! I thought about what was happening to me. Why was I enjoying this so much? I delighted in the freedom! I felt like a pioneer exploring in the realm of ideas!

Many people came to Logo 85 looking for ways to calm the chaos in their schools and in their own minds. I heard someone say, "My kids love Logo, but what are they learning? They're having too much fun. We need more structure." Logo critics would applaud this. They want proof that Logo teaches this or that. They are overlooking Logo's real power: it can make children fall in love with learning. But how does one measure that?

I fell back in love with learning during my preconference time. But, did I become a better planner? Can I now do math faster? Am I smarter? Probably not. But I do know that I'm happier and more confident as a result of my experience. And who knows what I can accomplish when I feel this way?!

Paul Lorentzen and Arun Keshap:

Logo 85 was a veritable collage of excitement, a plethora of activity, uplifting for the newcomers and rejuvenating for the old. It was an excellent blend of national and international involvement.

It is a wonder that there are still academicians who do not realize Logo's multiplicity of uses. If they had attended the conference, they would have seen the range of Logo extend from physics and meteorological forecasting to adventure games and all levels of teaching.

They would have enjoyed the open, casual atmosphere and the chance to share experiences with new-found friends. They might have even gotten a glimpse into a teaching method of the future - Lego™ and Logo, a combination of two simple but limitless building blocks of knowledge. To top it off, they could have mingled with the "visionaries" in the exhibition hall to discuss the latest Logo ideas and products.

Logo 85 was an enjoyable experience organized by remarkable people. They are to be commended for all events. We can't wait for Logo 86!

Gayle Lawrence:

It was evident from the conference that Logo is blooming at a time when educators worldwide are troubled by the scarcity of such stimulating environments for students. It is an unhappy realization that the educational community is failing to produce students who possess well developed problem solving and critical thinking skills which they can use in meaningful and productive ways.

As evidenced by presentations such as Ann Thompson's (of Iowa State University), Logo gives us a chance to set aside some of the low risk "pat answer" curriculum materials and to explore a world of dynamic, interactive, and (at times) unpredictable learning.

Molly Watt's demonstration of a model for recursion illustrated that teachers and learners alike are free to create their own models for the powerful ideas embedded in the Logo language. Dan Watt presented microworlds created by teachers using Logo, and suggested how these might be used to accomplish learning goals. Clearly, from David Thornburg's informative and entertaining presentation of the use of ExperLogo by Stanford University students, we are limited only by our own imaginations and creativity.

Kathleen Nutting is a Resource Teacher in the Computer Science Department of Jefferson County (CO) School District. Ihor Charischak conducts workshops for Logo Computer Systems, Inc. Paul Lorentzen is a student of nuclear engineering at the Georgia Institute of Technology. Arun Keshap is majoring in mechanical engineering at the University of Virginia. Both students use Logo in their studies. Gayle Lawrence is a computer teacher for grades 1 - 5 in Troy, NY.

NLXual Challenges

by Robs Muir

ENTOMOLOGY: The study of insects, i.e., bugs.

The word "entomology" is derived from a Greek root meaning to cut. The following NLXual Challenge constitutes an fascinating invitation from E. Paul Goldenberg to go on a "bug hunt" within an interesting Logo project. How good a Logo entomologist have you become? Can you "cut the bug?"

Bug Hunting

by E. Paul Goldenberg

Normally, I don't go out of my way to ask people not to be creative, but I have a rather subtle bug for you to find. This bug's residence is in a locale oft referred to as "the garden path." The flowers along this path are some of the sweetest and most seductive on the face of the globe.

To keep you moving, I must ask you to indulge me and allow me to lead you down the path. I promise that I will not mislead, but I cannot promise that I will point out where you may be stung.

In any event, stay close by me and, if you would like to see this most subtle of bugs, do not try any experiments I have not asked you to try just yet. Type my procedures exactly as I show them.

(If you have an MIT syntax Logo like Terrapin Logo, you will have to change the syntax of the lines containing IF and use a substitute for EMPTY, but make no other changes for fear of shooing the bug away before you see it.)

TO BUG :WHERE

Here is a procedure that increments by putting into a box named :COUNTER the value of 1 plus whatever had been previously inside that box.

```
TO INCREMENT :COUNTER
MAKE :COUNTER (1 + THING :COUNTER)
END
```

Try it out this way.

```
PONS (to show all the names you have - type PO
NAMES with MIT Logos)
MAKE "MY.SCORE 0
MAKE "YOUR.SCORE 0
INCREMENT "MY.SCORE
```

```
PONS (to show how the names have changed)
INCREMENT "MY.SCORE
PONS
```

Now, using INCREMENT, create a procedure TALLY that counts how many times each letter occurs in a word.

```
TO TALLY :WORD
IF (EMPTY :WORD) [STOP]
INCREMENT FIRST :WORD
TALLY BUTFIRST :WORD
END
```

To try out TALLY, you first need to initialize the counts of the letters you are about to count. So, try this sequence.

```
MAKE "A 0
MAKE "E 0
MAKE "I 0
MAKE "R 0
PONS (you should have MY.SCORE being 2, and 5
other names all set to 0)
TALLY "AIRIER
PONS (A and E should both be 1, while R and I should
both be 2.)
TALLY "EERIER
PONS (now the counts should reflect the totals in both
words)
```

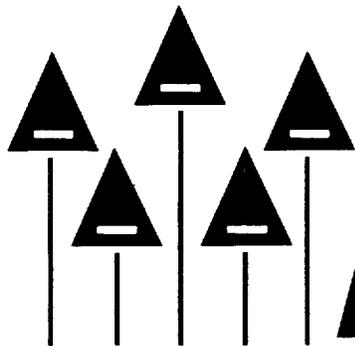
That, of course, suggests the idea of counting the frequency of occurrences of letters in an entire sentence by tallying each word sequentially.

```
TO TALLYSENT :SE
IF (EMPTY :SE) [STOP]
TALLY FIRST :SE
TALLYSENT BUTFIRST :SE
END
```

Of course, we must go through a lot of initialization before using TALLYSENT each time, so we have now developed a need for an initializer. This procedure uses a subprocedure that follows the same kind of logic that TALLY and TALLYSENT followed.

```
TO INIT :L
IF EMPTY :L [STOP]
MAKE (FIRST :L) 0
INIT BUTFIRST :L
END
```

Aren't the flowers beginning to smell particularly sweet here? Now, we are quite close, so be very still and don't run ahead! Try INIT exactly as I tell you to!



NLXtra

A Supplement to the National Logo Exchange Newsletter

September 1985

FORWARD 100!

NLXtra Page 1

IF: At First You Don't Succeed

by
Michael Tempel
and Mario Bourgoin

This is the last of four NLX articles about writing your own version of IF in Logo. After presenting the problem in the March issue, we gave a few solutions in April and extended the discussion in May.

The response to the articles has been strong. Many points of view came through, each expressing a different style of thought. All the user created IFs depended on the fact that Logo predicates output the literal words TRUE or FALSE and that these words may be the names of things in Logo.

MICHAEL'S.IF, shown in the April NLX, used the words TRUE and FALSE as LABELS for GO. (Similarly structured IFs were sent in by Sharon Burrowes and Eric Nagler.)

BRIAN'S.IFF and YVES'.IF use TRUE and FALSE as the names of lists which are then RUN. This technique was also used by Jan Bybee, Forrest Smith, Lorna Jinherson, and Patrick Salsich.

Sylvan Wallach submitted an IF that relies on the use of TRUE and FALSE as the names of procedures:

```
TO IF.THEN.ELSE :COND :IF.ACT :ELSE.ACT
RUN (SENTENCE :COND)
END
```

```
TO TRUE
RUN :IF.ACT
END
```

```
TO FALSE
RUN :ELSE.ACT
END
```

Solutions using Sylvan's general approach were also sent in by Robs Muir and Tom Lough.

Jan Bybee and Forrest Smith came up with an interesting variant:

```
TO IFELSE :CONDITION :TLIST :FLIST
DEFINE "TRUE [ [ ] [RUN :TLIST] ]
DEFINE "FALSE [ [ ] [RUN :FLIST] ]
RUN (LIST :CONDITION)
ER [TRUE FALSE]
END
```

Instead of writing the procedures TRUE and FALSE themselves, they have IFELSE create them using the primitive procedure DEFINE. One of these two procedures is run, and then both are erased.

IF OPERATIONP

In our April article, we raised the question of whether or not a user defined IF could be either a command or an operation like the primitive IF. Here's a solution written in Apple Logo II.

```
TO IF :COND :TRUE :FALSE
CATCH "ERROR [OUTPUT RUN THING :COND]
END
```

```
?IF 3=3 [PRINT "HI] [PRINT "BYE]
HI
?PRINT IF 3=3 ["HI] ["BYE]
HI
```

What's going on here? Well, the key is RUN THING :COND, which is the same as BRIAN'S.IF. Now, if we treat IF as a command, as in the first example, we get the desired result, but RUN (which may be a command or an operation) doesn't report (or "output") to OUTPUT. This error is caught by CATCH "ERROR so no complaint is printed and program execution continues.

If we treat IF as an operation, as in the second case, there is no error, so CATCH "ERROR has no effect. Unfortunately, all errors are caught.

IF LEARNINGP

Using a strategy similar to ours, Cynthia Solomon also wrote an IF that behaves either as a command or operation. But, in her letter, she raised an important issue: "In the case of developing a procedure for addition when there is no ADD primitive¹, I see an opportunity for learning a lot about computing and about addition, even if I end up with a limited ADD procedure. In this case, ... I am left wondering what you expect people to learn from the experience. What are the important ideas that this problem illustrates? Personally, I became reacquainted with the differences [between] Atari and Apple Logos."

In fact, there was no initial goal. Some of us simply became interested in the problem and pursued it. Partly, it was just the pleasure of the challenge ... the doing of something that first appeared impossible. Since we found it engaging, we figured other Logo people might also, so we presented it in these articles.

In hindsight, what has been the value of this exercise? What have we learned about Logo? What have we learned about other domains?

Along with Cynthia, one thing we have learned is that there are differences between Logos. These may appear minor at first, but they are really philosophical at the core. As with any high level language, Logo wants you to concentrate on the problem at hand, not on how Logo is implemented. Because you can create an IF, you should expect Logo's IF to behave as yours will. Those of you who have Apple Logo or an MIT Logo know that this is not the case². Should Logo support more English-like features? Part of Logo's power is that it asks you to say exactly what you mean. How can Logo be more intuitive and yet remain unambiguous?

Most versions of Logo support an IF that takes an optional input. This is possible because those who write Logo have access to features that are not available without machine language level tinkering. Optional inputs can make Logo programs hard to understand and make debugging difficult. You might be better off using Lisp. Its parentheses make clear which procedures get what inputs and from whom. This is called "parsing."

A more technical issue is the difficulty for Logo users to create new utilities that work correctly in all cases. Remember the problem of dealing with errors in user created versions of IF? This points out large holes in Logos. Many primitives are missing, such as those that allow more complete error handling or the generation of your own error messages. Try writing a STEP procedure. Does it handle errors properly?

On the positive side, we learned about the power of naming in Logo. All of the solutions rely on the fact that the words TRUE and FALSE may be the names of lists, procedures, property lists, and labels. Logo allows

you to name something to be done in all of these ways. This is because there is little difference in Logo between data and procedures. A list can be seen as a container of words or as something to do.

We could now produce an IF that includes a MAYBE case and move into the world of multi-valued logic. Such an IF would still work with Logo's predicates that return TRUE and FALSE, but we would add new predicates that return this name MAYBE in special circumstances.

IF DECISIONP

We have also found out something about how Logo makes decisions. The trick is deciding what to do in the TRUE case, and what to do in the FALSE case. A predicate is used to find out which case applies and we then RUN the action that this case names. (How much does this relate to how people make decisions?)

This idea can be used in other circumstances. For example, Yves Poiree wrote a straightforward way to change the penstate.

```
NAME [PENDOWN] "PENUP
NAME [PENUP] "PENDOWN
```

```
TO CHANGE.PEN
  RUN THING FIRST PEN
END
```

What's happening here? We name what to do if the pen is up and what to do if it is down. FIRST PEN tells us which is the case so it acts as the predicate. Then we find the thing named by the penstate and RUN it.

We hope you have enjoyed the IF articles. Please share your further explorations and discoveries with us. Who knows? Perhaps you will be the first one to write an OUTPUT procedure in Logo!!

1. See: Solomon, Cynthia J. Teaching the Computer to Add: An Example of Problem Solving in an Anthropomorphic Computer Culture. Logo Memo No. 41, 1976, MIT AI Lab.

2. See: Harvey, Brian. Computer Science Logo Style. MIT Press, 1985. Appendix A contains a thorough discussion of the differences between the various versions of Logo.

Michael Tempel is Director of Training with Logo Computer Systems, Inc., 555 West 57th Street, Suite 1236, New York, NY 10019. (212) 765-4780. Mario Bourgoin is a Research Assistant with the Learning Technologies Group, Arts and Media Technologies, Massachusetts Institute of Technology, Building E15-309, 20 Ames Street, Cambridge, MA 02139. (617) 253-8579.