The National LOGO Exchange

ISSN 0734-1717

Volume 1 Number 8 FORWARD 100! April 1983

Tessalate Your NLX ABC's

by Jim Fry

After receiving NLX’s great set of procedures for drawing the alphabet, I immediately typed them in, debugged, and saved. I then proceeded to try them out. "WOW!" is all I can say! They worked fantastically! My community education advanced LOGO class was going to be starting in two weeks and I knew the kids would love using the alphabet.

I saved the alphabet procedures on the 14 disks I use for my class. (I am very lucky to have a computer lab with 14 Apple II’s in one room at our high school for my LOGO classes.) When the students arrived, I introduced them to the ABC’s and off they went!

Of course, the first thing everyone did was to spell out his or her name. Then, some of them started to have the turtle repeat the same letter over again to make a pattern.

Just the week before, I had been reading the School House Apple article on tessalations. (SOFTALK January 1983, page 228, by Jim Muller of the Young Peoples’ LOGO Association.) I had been trying to think of a way to do tessellatons with my students. Of course! Let’s use the NLX Alphabet!

I talked to my students about patterns and repeating the same shape or set of shapes over and over again. They had already started doing tessellations with the alphabet while they were playing with the letters. We decided to set up a procedure called PATTERN.

```
TO PATTERN
  PU SETX -125
  FD 70 PD
  REPEAT 20 [PD PC 2 M PD PC 4 K]
  END
```

This was our first procedure. The students picked the letters and colors they wanted. We used all the PENDOWNs because a lot of the letters leave the pen up after they are drawn. (Ed. note: This shortcoming has been corrected in the newly revised version now available.) The students also experimented with different size letters and different amounts of repeating.

continued on page 2

Ed Emberley and the Turtle

by Sharon Burrowes
assisted by David Burrowes

This year, I have found myself rather suddenly involved in helping teachers find ways of using LOGO in their classrooms. I am always on the lookout for new ideas to suggest from whatever source I can find.

Last year, as I was just beginning to experiment with LOGO procedures, my husband remarked that the ability to combine predefined shapes reminded him of some drawing books our children had used when they were small. I agreed, filed it under "things to think about someday," and went back to trying to learn LOGO!

Then, just recently, I received a copy of the NLX ABC’s. After keying them in and having a delightful time fiddling with them, I remembered my husband’s remark of last year, and set off to scrounge for the drawing books.

To my delight, I found them exactly where I remembered them! The books, ED EMBERLEY’S DRAWING BOOK OF ANIMALS and ED EMBERLEY’S DRAWING BOOK, MAKE A WORLD (Little, Brown Co. Boston), each suggest a "vocabulary" of letters, numbers, shapes, and "things" (such as curlicues) that can be used to create any of the delightful drawings in the book.

Since I already had the NLX letters, I set out to complete the "vocabulary" necessary for the DRAWING BOOK OF ANIMALS. I made some modifications to the NLX letters, and added most of the numbers, shapes, and "things" that Mr. Emberley suggests. Before I had finished, my high school aged son, David (who loved the books when he was younger), got involved in the project and began creating procedures to do some of the drawings in the book.

I heard many mutterings of "Almost!" and "Darn!" over the next several hours. Later, he commented that doing these drawings was not easy ... and I would agree.

The books are aimed at free hand drawing, but they are definitely procedural in concept. Judicious choices of drawings from the books could result in many challenging LOGO project ideas on which children could build.

continued on page 2

The National LOGO Exchange, copyright 1983 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the USA. Single copy price $3. The opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress St., Salem, MA 01970. Address all other correspondence to The National LOGO Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717/83 $2.00

Editor........................Tom Lough

The Turtle Elite

During the Middle Ages, one of the sources of power for the Church was literacy. The ability to read and write was guarded carefully, and was transmitted only to selected persons. Such was the power of the written word.

With the arrival of computers, the potential exists for a similar elite class to arise. If computer literate educators are not willing to share with others, there is cause for concern.

Let's face it. The computer using teacher will be able to accomplish much more than a "regular" teacher. Using a word processor to create handouts and other lesson materials can save countless hours, for example.

Similarly, a LOGO teacher will be able to create within his/her classes an excitement difficult to match with traditional methods. Children will be learning in new ways with a powerful tool of today's electronic age. Thus, access to the computer and knowledge of its capabilities become key points of concern.

I can imagine nothing more stultifying than a clique of LOGO teachers who dare anyone to join them, or to try to arrange for access to "their" computers. As we learn more and more about this incredible language and its potential for learning, let us all resolve to reject the idea of a LOGO elite.

If we are not willing to share our knowledge and facilities with others, the entire profession suffers. Even more importantly, the children suffer.

OUTWARD 100!

Tom

Emberley continued

So, if you are looking for some ideas to suggest to your children using LOGO, why not obtain Ed Emberley's wonderful drawing books and a copy of the procedures for the NLX ABC's, and try some experimentation yourself?

(Ed. note: To get your copy of the NLX ABC's, send a stamped self-addressed envelope to NEW NLX ABC's, PO Box 5341, Charlottesville, VA 22905. Several of the "thing" procedures referenced above are included with the listing. Also see the April 1983 issue of The Computing Teacher.)

Sharon Burrowes is a consultant and a computer science teacher in the Wooster (Ohio) City School system.

Tessalate continued

They then started having the turtle draw the PATTERN procedure all over the screen by typing PATTERN many times. We then made a procedure called DESIGN.

```
TO DESIGN
REPEAT 10 [ PATTERN ]
END
```

By playing with the REPEAT input number, the students were able to fill up the entire screen with some very interesting designs.

The students found out very quickly that letters with any type of curve took a long time to draw, so they used only letters like A, E, H, K, M, etc.

I have been working with the alphabet tesselations and have come up with some other procedures that I plan to show the students and let them experiment with.

My students have had a great deal of fun with the alphabet and I thank Tom Lough and the NLX for putting them together and sharing them with all of us LOGO fans.

FD 100! Jim Fry

Jim Fry teaches math, reading, and LOGO to elementary school students in Novi, Michigan. He teaches a series of summer computer courses also.

EdCom Postponed

EdCom is postponed! According to its organizers, the EdCom Conference scheduled for late April in Washington, DC, has been rescheduled for October 15, due to a "conflict with another major conference."

TIPPS for TEACHERS

by

Steve Tipps

Variables: More than one way to tame a turtle

Achieving power with a computer is letting the computer do things which relieve humans of drudgery and allow creativity. Repetition is one powerful idea in computing which repeats and recurs. LOGO's meaningful names for procedures, variables, and files are another example of using the computer's abilities rather than humans' to remember things like A1 or X4 or N$.

Variability is another way that the computer reduces work. Actually, variability has been a part of a student's LOGO experience since the first time a decision was made about how far to move or to turn the turtle. The words CHANGE, VARY, and INPUT should be commonplace. Typing PENCOLOR without any numbers reminds the user that PENCOLOR NEEDS MORE INPUTS. (It needs a number to determine which color to use when drawing.) When children have explored and can use economical REPEAT statements, create and embed procedures, and save their work on file disks, they should be ready for making procedures with variable inputs.

ONE WAY AND MORE

Different sizes of the same shape are often needed to construct a project in LOGO. A house may need squares or rectangles for the frame, doors, and windows. A cylinder can be made of circles and circleparts. One modular way is to describe each part of the design in a specific procedure that is unique to that project. HOUSE is composed of FRAME, DOOR, WINDOW, and TRIANGLE. All of the procedures are sized for only that house.

Basic shapes can also be labeled in some relative way. Procedures can be written and named BIG.SQUARE, MEDIUM.SQUARE, LITTLE.SQUARE, or GIANT.CIRCLE, MEDIUM.HALF.CIRCLE, ITSY.CIRCLE.PART. These names are partially descriptive; however, they will proliferate and are not fully descriptive. What is the relationship between BIG.SQUARE and BIG.TRIANGLE, for example?

Children should be able to see that a square can be changed in size by editing the input number for FORWARD. The size of the square is determined solely by the FORWARD input.

```
TO BIG.SQUARE
REPEAT 4 [FORWARD 99 RIGHT 90]
END

TO MEDIUM.SQUARE
REPEAT 4 [FORWARD 47 RIGHT 90]
END

TO SMALL.SQUARE
REPEAT 4 [FORWARD 13 RIGHT 90]
END
```

Comparison of the procedures makes it plain that the size is related to the FORWARD number.

I have suggested to students that they rename their procedures so that they can tell LOGO directly which square they want, rather than trying to remember how big a BIG.SQUARE is. The method I suggest is that they rename BIG.SQUARE as SQUARE.99, MEDIUM.SQUARE as SQUARE.47, and the like.

```
TO SQUARE.99
REPEAT 4 [FORWARD 99 RIGHT 90]
END
```

From this point, children can use a family of squares, circles, triangles, or hexagons to create nested, adjacent or radiating master procedures.

```
TO NEST.TRIANGLES
TRIANGLE.10
TRIANGLE.20
TRIANGLE.30
TRIANGLE.40
END

TO FAN.TRIANGLES
TRIANGLE.10 RT 10
TRIANGLE.20 RT 10
TRIANGLE.30 RT 10
TRIANGLE.40 RT 10
END
```

Projects using procedures named with an indication of the size demonstrates good understanding of the concepts. Although the issue of descriptiveness is solved, two problems remain.

a. The names get long. Children soon solve this by changing SQUARE.100 to SQ.100.

b. Many square procedures are being developed which differ only in one respect: the input number for FORWARD. This diverts the user from good economical programming.

Moving to names with numbers in them serves as a bridge in syntax and provides a lot of experience with variability of a procedure. LOGO teachers will recognize that this naming system is only a stepping stone toward the introduction of variable names and inputs.

continued on page 4

Tipps for Teachers continued

CHANGING WAYS

One or more of the students may be used to asking with regularity, "Isn't there an easier way to do this?". The answer with computers is usually YES. Finding the right time to move on to that easier way is the teacher's dilemma with LOGO. Waiting until children have enough experience to understand the concept is important before introducing the method of the easier way.

SQUARE.99, SQUARE.28, and SQUARE.47 are procedures which differ in size, stated in the procedure name as 99, 28, and 47. By now, children may be able to recognize that all of the SQUARE.xx procedures can be reduced to a single procedure by substituting a variable rather than a number as input to FORWARD.

```
     TO SQUARE :SIZE
     REPEAT  4  [ FORWARD :SIZE RIGHT
90 ]
     END
```

Any meaningful name may be used to refer to the changing value. Typical choices are :SIZE, :LENGTH, or :SIDE. The colon, called "dots" in LOGO lingo, is a signal that a value is expected in order for the procedure to work.

The :SIZE which you add in the naming line of the procedure allows you to input that value directly to the procedure. Adding :SIZE to the name line of the procedure is like a contract or promise to the procedure. Every time you call the procedure, you must give it a value to use for the variable.

Review what happens when you type FORWARD without a number.

```
     FORWARD
     FORWARD  NEEDS  MORE  INPUTS (or
some such message)
     FORWARD 10
```

With the last entry, a familiar result is obtained: the turtle makes a line 10 turtle steps long. Now try to type SQUARE without a number, having defined SQUARE :SIZE as above.

```
     SQUARE
     SQUARE NEEDS MORE INPUTS
     SQUARE 10
     SQUARE 20
     SQUARE 52
```

With the last entries, you have drawn squares of size 10, 20, and 52, all with a single procedure which has had a variable substituted for a specific number.

Two major ideas are explored in this seemingly simple way. First, the idea of variability and establishing a variable microworld is heady stuff. Second, the abstraction of signifiers being used for value is basic to logical thought.

Coming to grips with variability and signifiers in the turtle's world should prepare children to work with ideas in algebra, geometry, social studies, literature, etc. You may say that algebra and geometry are obvious, but why social studies and literature? I ask you, aren't metaphors and personification, economic theories and cultural rituals rich examples of variability and abstact signifiers?

MANY WAYS

The use of variable inputs to variable procedures may soon generalize to the substitution of variable names for all numbers. One seventh grade class I worked with insisted on changing all the numbers in SQUARE:

```
     TO    SQUARE    :SIZE    :TURN
:NUMBER.OF.TIMES
     REPEAT :NUMBER.OF.TIMES [FORWARD
:SIZE RIGHT :TURN]
     END
```

They discovered that the procedure worked only if three numbers were provided as input. But, they also learned that, suddenly, the procedure name SQUARE was no longer very descriptive!

When challenged to figure out how to make the turtle return to its starting place, they rediscovered the Total Turtle Trip Theorem as a special relationship between :TURN and :NUMBER.OF.TIMES. A new generalized procedure was made up to account for these discoveries.

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

In the new variable microworld, children also found that there was a relationship between triangles, squares, hexagons, octagons, twelve-agons, ... and circles, based on varying :NUMBER.

Working with variables is important, one of the most important steps in programming. Using variables in procedures is a good start, but it is not the end.

Using assignment statements for global variables, initialization of variables, passing parameters between procedures, reading values for variables from the keyboard and paddles in interactive procedures -- all these provide many ways to explore and extend variability. Almost all of the interesting things with list processing in LOGO demand a good grasp of variables. When you help your children discover many ways to tame the variable turtle, you have given them much power in thinking and programming.

Steve Tipps is a professor at the University of Virginia, and has presented LOGO workshops for teachers throughout the eastern United States.

MICROWORLDS

by
Glen Bull

Teaching Grammar to a Computer

One of the best ways to learn is to teach someone else. The foundation of LOGO is based on the child teaching the computer, rather than use of the computer to program the child. Rules for formation of plurals provide an illustration of one way this might be done.

AN ENGLISH DESCRIPTION

The first step in teaching a computer to form plurals involves a thinking of a description of how plurals are formed. A child's first attempt might be:

To form a plural, ADD S TO THE END OF THE WORD.

This English description is the most important part of the process of teaching the computer to form plurals.

TRANSLATING THE DESCRIPTION INTO LOGO

The next question is, "How can this be translated into a LOGO procedure?" As it happens, this is not a difficult problem. The key to learning focuses on the initial description, not minor details of writing the short LOGO procedure. The LOGO instruction WORD can be used to combine words:

PRINT WORD "FIRE "TRUCK results in FIRETRUCK.

The same process can be employed to add S to the end of any item:

```
TO PLURAL :ITEM
OUTPUT WORD :ITEM "S
END
```

This procedure states, "Return as output a word consisting of an item plus the letter S attached to the end." For example,

```
PRINT PLURAL "BOY
BOYS
```

In this instance, LOGO provides a means of verifying the rule. Inevitably, another child will use this procedure in the following way:

```
PRINT PLURAL "BOX
BOXS
```

DEBUGGING THE RULE

The most important learning occurs because the procedure does not work, rather than because it does. In this case, the old rule of adding S must be modified to account for other possibilities. This leads to the discovery that, in general, if a word ends in X, the plural is formed by adding ES rather than S. This rule may be expressed in English in the following way:

To form a plural, IF THE LAST LETTER IN A WORD IS X, ADD ES; OTHERWISE, ADD S TO THE END OF THE WORD.

The last letter in a word can be examined using the LOGO command LAST.

```
PRINT LAST "BOY
Y
PRINT LAST "BOX
X
```

This instruction provides a means of determining whether a word ends in X.

```
TO PLURAL :ITEM
TEST LAST :ITEM = "X
IFTRUE OUTPUT WORD :ITEM "ES
OUTPUT WORD :ITEM "S
END
```

In this modification of the procedure, the last element of the item (LAST :ITEM) would be Y in the instance of BOY and X in the case of BOX. In the first case, the test would fail to obtain a match, and the plural would be formed by adding S. In the second case, the test would produce a TRUE result, and the plural would be formed by adding ES. (Note that the procedure is terminated when _either_ of the OUTPUT commands is executed.)

Again, it must be noted that the mechanics of how the rule is stated in LOGO are not as important as the discovery of the rule itself. LOGO is only being used as a _tool_ to verify the hypothesis.

The role of the teacher is not to write a program which forms plurals for the child. Instead, it is to guide the process into productive areas and away from sterile dead ends, and to ensure that irrelevant mechanical details do not interfere with the learning process.

By this time, there may be some competition among the students to determine who can produce the smartest spelling program. Some teachers may have students compete in teams of two or three, depending on the teaching environment and philosophy.

Microworlds continued

At this point, energies will be equally divided between efforts to improve proprietary PLURAL programs, and efforts to discover words which will fool competing programs. Efforts to discover weaknesses in other programs are just as valuable as improvement of a team's own program, since those efforts also lead to extensions of the rule system. For example, a team might discover that PRINT PLURAL "GRASS produces GRASSS! This presents a problem which could be relieved in the following way:

```
     TO PLURAL :ITEM
     TEST LAST :ITEM = "X
     IFTRUE OUTPUT WORD :ITEM "ES
     TEST LAST :ITEM = "S
     IFTRUE OUTPUT WORD :ITEM "ES
     OUTPUT WORD :ITEM "S
     END
```

DETERMINATION OF GROUP MEMBERSHIP

When there are several items to be tested, the command MEMBER? (MIT) or MEMBERP (LCSI) can be used to evaluate the possibilities. This command is used to determine whether an item is an element of a list. This instruction could be used to test whether the last letter of the word were one of S, X, or Z. For example,

```
     TEST MEMBER? LAST "BOX [ S X Z ]
```

would produce a result of TRUE because the last letter X is a member of the list (S X Z). This instruction is a primitive (built-in) command in some versions of LOGO. For those in which it is not (TI LOGO, for example), it is possible to construct this command.

```
     TO MEMBER? :ITEM :LIST
     IF :LIST = [ ] OUTPUT "FALSE
     IF :ITEM  =  FIRST  :LIST OUTPUT
"TRUE
     OUTPUT  MEMBER?  :ITEM  BUTFIRST
:LIST
     END
```

It is not necessary to follow the strategy employed in this procedure in order to use it. The first element in the list is checked to determine if it matches the item in question. If it does, the procedure returns a value of "TRUE. If not, the first element of the list is removed and the process is repeated until a match is obtained or the list is empty. If all the items have been checked without obtaining a match, the procedure returns a value of FALSE.

MEMBER? provides an economical means of testing several possible conditions which would require the addition of ES rather than the S to form the plural.

```
     TO PLURAL :ITEM
     TEST MEMBER? LAST :ITEM [S X Z]
     IFTRUE OUTPUT WORD :ITEM "ES
     OUTPUT WORD :ITEM "S
     END
```

This version of PLURAL does not encompass all possible methods of forming plurals. There are the cases of suffixes such as CH and SH and of words such as BABY (Readers of last month's NLX will recognize the significance of this word!). The particular methods used to express the rules for plural formation in LOGO are less important than the student's attempts to discover those rules. LOGO simply provides a dynamic means of testing those rules.

CLASSROOM USE

Most of the students who might attempt to describe the rules for plural formation will have had exposure to these rules in previous grades. There is a discrepancy between the level of the content material learned (formation of plurals) and the level of understanding of LOGO required to teach this content to the computer.

LOGO provides a means of exploring these spelling rules in greater depth, even though they may have been learned in earlier grades. In school, the way a rule works is often learned well before the need for a rule is understood. The strange appearance of the word GRASSS on the screen and the difficulty in pronouncing such a word create for the student a reason to find an alternative to the rule ADD S TO THE END OF THE WORD.

In this case, the knowledge of the particular rules may be less important than the sudden insight into why the rules operate in the manner in which they do. At the same time, mechanical LOGO skills are being developed which can be used to attack more sophisticated syntactic and linguistic concepts. Some of these possibilities will be discussed in next month's column.

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

New ABC's Announced

Many of you have received your copy of the NLX ABC procedures for the Apple LOGO versions. Here is some important news for you. There is now a "new improved" set of ABC's available! Many subscribers have helped point out ways to make the ABC procedures even better. If you received your ABC listing before March 12, then you probably have the "old" version. To get a copy of the new, send a stamped self-addressed envelope to NEW NLX ABC's, Box 5341, Charlottesville, VA 22905.

Creating a LOGO Environment

by Tim Riordon

(Editor's note: This is the final section of a 2-part reprinting of Tim Riordon's significant article which originally appeared in the November 1982 issue of The Computing Teacher. Reprinted with permission.)

Throughout the day, the teacher is seen consulting a list with eight student names. When a few minutes are available, the teacher looks at the list and selects one student for an impromptu 5-minute conference. Tomorrow, another eight names will be added to the list.

The first question in these conferences always seems to be, "What are you doing with LOGO this week?" Some students have elaborate plans for games or procedures. Other students, exhibiting a very different manner of working, explain that they are "messing around" with one procedure. By "messing around," they apparently mean to work with one procedure, make a change in it, run it, make antoher change, run it again, etc.

Evidently, the teacher does not try to coerce students into adopting another method. "Messing around" with LOGO procedures is valued. The teacher is sensitive to and accepting of different intellectual styles.

The second set of questions is usually, "Have you had any unexpected results? Do you have a theory about what caused this? What did you try in attempting to solve the problem?" The teacher is using questions to help students focus upon their own theorizing, and is promoting learning by having students theorize, test and revise, instead of always having them learn by direct instruction.

The central question in these short conferences always seems to be, "Is there anything you are having trouble with on your current LOGO project?" If the answer is yes, the student is asked to draw what happens and if s/he has any ideas about what might be causing the problem.

Frequently, the problem is complex enough that the student and teacher agree that they need to watch what happens on the computer. Somehow, a few minutes at recess, lunch, or after school are found so that the problem can be demonstrated.

Sometimes the teacher provides the student with an answer. Other times the teacher will give hints and encourage the student to try another approach to solving the problem. Often the teacher will tell the student to use the "masking tape strategy." Using this strategy, a problematic design is examined and then taped on the floor. The student plays turtle and walks through the procedure.

Using LOGO language, the student describes everything s/he is doing to walk the design. Sometimes the teacher or another student copies down each statement. The result is usually the LOGO procedure needed to create the design in question.

Sometimes students will say that they are done with a project or are bored with it. With some students, the teacher's resources for project ideas are offered as possibilities: "LOGO for the Apple II" by Abelson, and "Final Report Brookline LOGO Project, Parts II and III" by the MIT LOGO Group. With other students, the teacher encourages and occasionally directs that something specific be done - usually a project that the teacher believes will open up new LOGO vistas for that particular student.

"Bill, I'd like you to do a little project that makes the computer draw umbrellas of increasing size all over the screen. Do you have an idea about how you would do that?" (Bill understands how to use variables and recursion, but has not used the two ideas together; the teacher makes sure that Bill has some idea about how to begin so he will not experience great frustration.)

"Katie, would you rewrite this procedure that always draws 9-pointed stars the same size? Make it so that you can input both how big and what color the 9-pointed star is. Do you have some ideas about where you would change this procedure?"

"Matthew, I think that you will be able to solve your problem if you try the project on the wall that asks you to use a circle to make a slinky design. If you do that project and still can't figure out how to do what you want in your current project, come on back and see me."

After watching and listening to these 5-minute interactions throughout the course of a week, one can draw several conclusions. The teacher:

-is sensitive to whether or not students are headed for frustration.

-has in mind a sequence of LOGO programming concepts and a large repertoire of LOGO project ideas.

-often considers his/her teaching role not as a repository of answers but as a midwife helping students discover answers by theorizing about problems and unexpected results.

Using all of this knowledge, the teacher exercises the art of teaching in quickly discerning how to respond to each student as an individual. Some need an answer and some just a hint; some need a practice project; and many need nothing but more computer time.

continued on page 8

LOGO Environment continued

A LOGO environment has many of the attributes of a democratic classroom. Authority is distributed; sharing and cooperation are promoted; students look to their classmates as legitimate sources of information; because students can make project choices, variety rather than uniformity is the norm; rewards are intrinsic; differences in working style are valued. There is a sense of shared learning - students may hear their teacher respsond, "I'm not sure how to make the computer do that. I'll work on it at home tonight."

WHAT FEATURES ARE CONSPICUOUSLY ABSENT FROM A LOGO ENVIRONEMENT?

There is no evaluation of student learning prepared by the teacher and given to students or parents. There are very few required projects which students are expected to do. Many of those that are required are selected specifically for an individual student rather than for the whole class.

While there is no formal evaluation of a child's LOGO learning, parents are kept abreast of what their kids do, sometimes through a parent night, where students demonstrate their programs. LOGO lends itself to class projects, booklets and calendars with LOGO graphic designs by each class member. It should also be noted that LOGO excites students enough to go home and volunteer information to parents on " what we did in school today."

WHY SHOULD SUCH AN ENVIRONMENT BE SET UP IN A CLASSROOM?

Why should classroom instructional time be used by students in an activity which is not formally evaluated? The most complete answer to these questions is to be found in Seymour Papert's book, MINDSTORMS: Children, Computers and Powerful Ideas. LOGO was created because of a fundamental belief that children can learn many more important things without formal instruction.

Just as children learn verbal language without formal instruction - a vocabulary of thousands of words and a set of complex syntactical rules for putting words together meaningfully - so, too, the language of mathematics can be learned. No matter what one's theory about how language can be learned, it is a fact that children learn language when they (interact with) a language-rich environment.

A LOGO environment is a mathematics-rich environment, a context including not only a computer and walls and floors with project ideas but also how students and the teacher in the classroom interact with each other. In an environment where mathematical objects and ideas are joyfully shared, played with, discussed and encouraged, mathematical intuitions and language will be learned without excessive formal instruction.

What lies behind all of this is a Piagetian view of children who learn because they naturally make and revise theories about things they are interested in.

HOW DOES ONE BEGIN CREATING A LOGO ENVIRONMENT?

Once you begin learning LOGO yourself, introduce it in your classroom. This article was written with self-contained classroom teachers in mind. In middle schools, high schools and other situations that are not self-contained, teachers can invent other ways of organizing. Get LOGO going as a peripheral activity; that is, let your regular curriculum proceed but allow individual students to miss part of it. Students who miss instruction can be treated like students who were absent. With older students, it is possible to strike a bargain- a classroom agreement. A student working on the computer will be responsible for checking with a friend about what s/he has missed.

In the beginning, you will need to demonstrate a number of things to the whole class. The question of scope and sequence is problematic. LOGO inventors fear that if a scope and sequence is published, it will invite the belief that students should be accountable for learning LOGO programming concepts. This will inevitably lead to evaluation of student learning. This will lead to LOGO being a joyless, unnatural learning learning activity.

LOGO was not meant to be taught, and the kind of learning that occurs was not meant to be evaluated like other school learning!

It is in the spirit of LOGO that you should begin the project of creating a LOGO environment with the idea that you will try something, debug it and try again.

References

Abelson, Harold. "LOGO for the Apple II." New York: BYTE/McGraw Hill, 1982.

Adams, James L. "Conceptual Blockbusting." Stanford, CA: Stanford Alumni Association, 1974.

BYTE Magazine. Entire August, 1982, issue is devoted to LOGO.

MIT Logo Group. "Parts II and III Final Report Brookline LOGO Project." Cambridge, MA: Artificial Intelligence Publications, Massachusetts Institute of Technology, 1979.

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

Tim Riordon is a teacher and consultant in Eugene, Oregon, and is a regular contributor to The Computing Teacher magazine.
