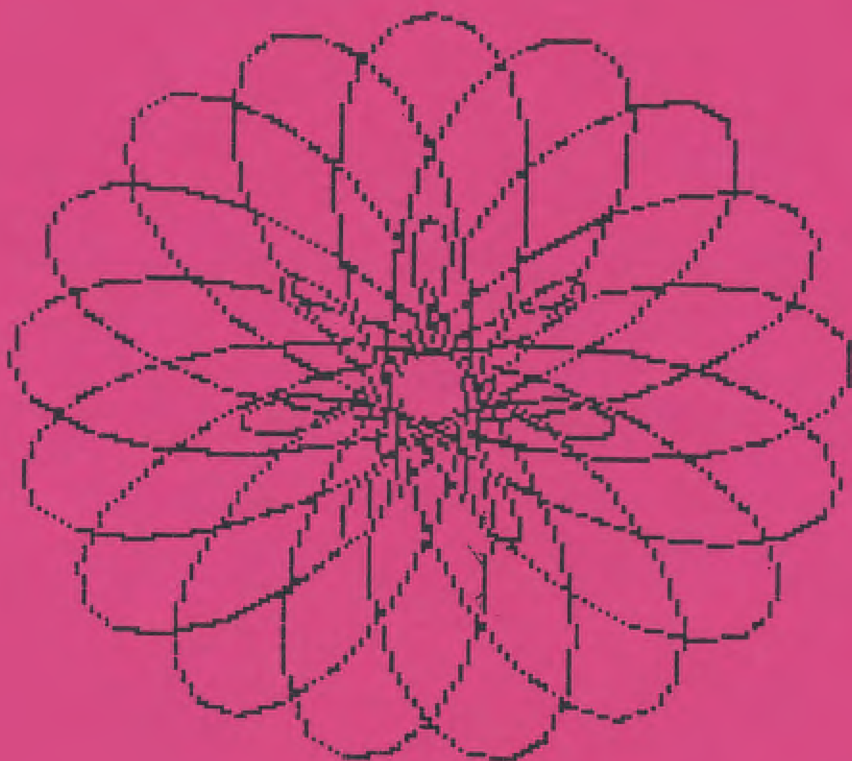
Journal of the ISTE Special Interest Group for Logo-Using Educators

LOGO EXCHANGE

October 1990

Volume 9 Number 2



International Society for Technology in Education

Publications

LOGO EXCHANGE

Volume 9 Number 2 Journal of the ISTE Special Interest Group for Logo-Using Educators October 1990

Logo Exchange is the journal of the International Society for Technology in Education Special Interest Group for Logo-using Educators (SIGLogo), published monthly September through May by ISTE, University of Oregon, 1787 Agate Street, Eugene, OR 97403-9905, USA; 503/346-4414. This publication was produced using Aldus *PageMaker*®.

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

Contents

ISTE Membership

	U.S.	Non-U.S.
	36.00	43.00

SIGLogo Membership (includes *The Logo Exchange*)

	U.S.	Non-U.S.
ISTE Member Price	25.00	30.00
Non-ISTE Member Price	30.00	35.00

Send membership dues to ISTE. Add $2.50 for processing if payment does not accompany your dues. VISA and Mastercard accepted. Add $18.00 for airmail shipping.

From the (Guest) Editor

How Soon They Forget
by David Moursund

In this issue of *LX*, Dennis Harper provides a brief report on Logo at the World Educational Computing Conference that was recently held in Australia. He reports:

> During one crowded session, Seymour Papert made some statements, filling in for one of the panelists. An Australian teacher asked SIGLogo president Gary Stager, "Who was that guy?" When Gary told her it was Seymour Papert she replied, "How do you spell Papert?"

This statement really struck a note with me. It suggests that the Australian teacher had enough interest to attend a Logo session, but was not familiar with the work of Seymour Papert and his seminal work in helping to create Logo. While it is painful for long-time Logo users to admit this possibility, it does reflect a maturation of Logo. Logo is an idea that transcends its creators and early disciples.

Logo has come a long way in the past 20 years. It has made significant contributions to the overall field of computers in education, has been vastly changed and improved through a process of continual formative and summative evaluation, and has spread worldwide. Logo's ability to captivate novice computer using teachers remains as strong as ever. It is obvious that Logo will continue to be a major force in computer education for the foreseeable future.

I would have liked to have had the chance to talk to the Australian teacher who was not familiar with Seymour Papert. I would have asked her about how she felt Logo could be used to help her students. I wonder if she would display understanding of some of Seymour Papert's powerful ideas on creating learning environments in which students get better at problem solving?

This past summer I taught a course entitled Frontiers of Computers in Education. One of the topics we covered was roles of computers in problem solving. The class got into a rather heated discussion about "pure" discovery-based learning versus "guided" discovery versus direct instruction. One of my doctoral students had just finished a dissertation on the issue of transfer of learning in problem solving, so I was somewhat familiar with the current research and underlying ideas. Many Logo-using educators seem to feel that if students are placed in a Logo environment, they will discover for themselves powerful ideas that will help them to become better problem solvers both within the Logo environment and in other areas.

Unfortunately, that is just not true. The research says that such discovery of powerful ideas and transfer to other areas does not occur without the help of well qualified teachers and without an instructional environment explicitly designed to help increase transfer. (See Doug Clements' column, Search and Research, in this issue.) In recent years, Seymour Papert has repeatedly emphasized this point in the talks he has presented at conferences. Logo cannot stand by itself. Its success in education depends on well qualified and motivated teachers who have a good understanding of what they are working to accomplish as they place their students in a Logo environment.

About two years ago, Sharon Yoder and I published a series of articles in *LX* on Logo and problem solving. We emphasized that Logo provides a rich environment for the explicit teaching of many important problem solving ideas. We have continued to work on those ideas in the ensuing months. Most recently we completed two short books on these ideas: *Introduction to LogoWriter and Problem Solving for Educators*, and *Introduction to Logo Plus and Problem Solving for Educators* The concepts needed to get started using Logo and the explicit teaching of problem solving are woven together throughout the books.

These books are designed to be used in a two to three week introduction to Logo. The problem solving ideas in both books are the same, and are independent of any particular version of Logo. If you are a teacher of teachers, ISTE would be happy to provide you with a free examination copy of one of these books. Write to Sharon Yoder at ISTE and indicate which one you would like to receive. Let her know that you read about this free offer in *LX*!

Dave Moursund is Executive Officer of ISTE, supervises masters and doctoral programs in computers in education at the University of Oregon, and has written numerous books and articles in the area of computers in education.

Dave Moursund
ISTE
1787 Agate Street
Eugene, OR 97403
BITNET: ISTE@Oregon

Sharon Yoder
ISTE
BITNET: YODER@Oregon

Monthly Musing

Should the Turtle Go to High School?
by Tom Lough

Logo? Ah, yes, the computer language for children.

Yes, Logo is the computer language for children. But it is also the computer language for many other people, too. And I'm tired of it getting a bum rap!

During the past few months, I have been making conference presentations about the potential of Logo use in secondary education. In particular, I discussed Logo possibilities in secondary mathematics topics at the annual meeting of the National Council of Teachers of Mathematics in Salt Lake City, and spoke on Logo use in secondary science and mathematics at the World Conference on Computers in Education in Sydney, Australia.

In both cases, the presentations were well attended, and included those who were already using Logo as well as many who were not familiar with it. In both presentations, the attendees gave a remarkably enthusiastic reception to the ideas and examples presented. When given an appropriate introduction to Logo with relevant examples, these secondary teachers appeared to be more than wiling to give it a try.

I do not believe that my presentation was particularly innovative. I simply reported my professional experiences and my assessment of Logo use at the higher grade levels. I described a few of my concerns, but concluded that the language had a lot of potential for more than just elementary school students. The responses indicated to me that something could happen.

But nothing is really going to happen, is it? Even though Seymour Papert spent an entire chapter in *Mindstorms* showing how Logo could be used in physics and Hal Abelson and Andy diSessa's *Turtle Geometry* described many sophisticated mathematical projects, not many people have paid attention, have they? Aren't Pascal and BASIC still the computer languages of secondary schools?

I'd be discouraged if I had not seen some signs of life: the continuation and growth of the Council for Logo in Mathematics Education (CLIME); the expanding series of more sophisticated Logo books from the MIT Press; the development and introduction of more powerful versions of Logo appropriate for high school use by Logo Computer Systems Inc. and Terrapin Inc.; and, perhaps most important of all, the steady movement up through the grade levels of a group of students with Logo experience who have some idea what the language can really do. I really feel frustrated for those students, because they are in a position for some very special learning experiences that will never come to pass if they finish high school without ever invoking their Logo skills.

I hope you are as tired as I am of Logo not getting the "respect" it deserves. What can and should we do about that? Why not write to CLIME for information about their organization? (10 Bogert Avenue, White Plains, NY 10606.) Get one or more of the MIT Logo books and show them to students and secondary teachers. Learn something about LogoWriter Secondary or Logo PLUS. Try to find some way to encourage high school students and their teachers. Students might be surprised at how much high school learning they could accomplish. High school teachers will most likely be flabbergasted!

Then something just might begin to happen. And it's about time.

Until then,

FD 100!

Tom Lough
Founding Editor
PO Box 394
Simsbury, CT 06070

Beginner's Corner

What's Cookin'?

by Dorothy Fitch

```
HALLOWEEN DELIGHT - SERVES 8

3 PINTS OF DRAGON WINGS
10 PINTS OF DRAGON WINGS
8 CUPS OF BAT FEET
4 PINTS OF LEMONADE
9 CUPS OF VINEGAR
MIX ALL 6 INGREDIENTS
THEN BROIL FOR 6 HOURS
SERVE WITH PEAS AND LEMONADE
```

Here's a program that will introduce your kids to Logo words and lists, just in time for Halloween. They will learn to distinguish between words and lists and to edit procedures, and their creative juices will surely flow. This program creates recipes from randomly selected words in different categories. The main procedure is RECIPE. (These procedures listed in this column will work with almost every version of Logo. The instructions on the Copy-Me page are designed for use with Terrapin Logo for the Apple or Logo PLUS, but with minor modifications, they can be used with other versions of Logo.)

```
TO RECIPE
CLEARTEXT
(PRINT TITLE [ - SERVES] NUMBER)
PRINT [] ;this prints a blank line
REPEAT NUMBER [(PRINT NUMBER MEASURE
   [OF] ANIMAL PARTS)]
PRINT []
REPEAT PICK [1 2 3] [(PRINT NUMBER
   MEASURE [OF] LIQUID)]
PRINT []
(PRINT NONCOOK [ALL] NUMBER
   [INGREDIENTS])
(PRINT [THEN] COOK [FOR] NUMBER TIME)
PRINT []
(PRINT [SERVE WITH] VEGETABLE [AND]
   LIQUID)
PRINT []
END
```

There are actually only a few Logo primitives in this procedure: CLEARTEXT, PRINT and REPEAT. Text in square brackets will always be printed as is (except for the lists that are given as input to the PICK procedure). All the rest of the words are names of other Logo procedures (NUMBER, MEASURE, ANIMAL, PARTS, etc.), which report back information to RECIPE.

Here are the procedures you will need to use the RECIPE procedure (its subprocedures):

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

PICK is a very important procedure, as it provides a means of picking randomly from the lists stored in the other procedures. It takes a list (or word) as input and reports an item selected at random. The input can have any number of items, as PICK uses COUNT to find out how many are available and picks a random number based on that count to select a random item. You can see how this works by typing:

```
PRINT PICK [BUNNY MOUSE TURTLE]
```

The rest of the procedures contain lists in various categories. PICK reports one item from the list to RECIPE, where it is printed as part of a phrase. Here are the procedures you can start with:

```
TO TITLE
OUTPUT PICK [[HALLOWEEN DELIGHT]
   [MYSTERY MEAL]]
END

TO NUMBER
OUTPUT PICK [2 3 4 5 6 7 8 9 10 11
   12 20 50]
END

TO MEASURE
OUTPUT PICK [CUPS PINTS]
END

TO ANIMAL
OUTPUT PICK [BATS DRAGON]
END

TO PARTS
OUTPUT PICK [WINGS FEET]
END

TO LIQUID
OUTPUT PICK [VINEGAR LEMONADE]
END

TO NONCOOK
OUTPUT PICK [STIR MIX]
END
```

```
TO COOK
OUTPUT PICK [BAKE BROIL]
END

TO TIME
OUTPUT PICK [HOURS MINUTES]
END

TO VEGETABLE
OUTPUT PICK [PEAS [BRUSSEL SPROUTS]]
END
```

Notice that when a list contains a item with more than one word (as in Brussel Sprouts), the entire item is placed in square brackets, creating a list within a list. This tells Logo to keep the words together when it picks an item at random. The vegetable list, therefore, contains 2 items: the word peas and the list [brussel sprouts]. The Copy-Me page that follows can help your students get started. It assumes that you have entered these procedures and saved them on a disk with the filename RECIPE. Then your students will have fun customizing the program with their own words.

What else might your kids be learning?

Spelling:

- they will learn to spell names of their favorite (or not so favorite) animals and vegetables

Math:

- they can calculate how much volume their recipe will produce (Is it enough or too much food for the number the recipe says it serves?)

Science:

- they will investigate the anatomy of various animal forms
- they will learn and use terms relating to units of measure and time

Practical Arts:

- they will learn cooking terms (and perhaps be inspired to experiment with culinary arts at home!)
- they can think about whether the set of ingredients from their randomly created recipe is more like a soup, a thick stew, or a casserole that's a bit on the dry side!

A former education and computer consultant, Dorothy Fitch has been the Director of Product Development at Terrapin since 1987. She can be reached at:

Terrapin Software, Inc.
Riverside Street
Portland, ME 04103
(207) 878-8200
CompuServe: 71760,366

CELEBRATING TEN YEARS OF EDUCATIONAL COMPUTING

The Great Lakes/East Coast Logo Conference will be combined with the annual Educational Computer Fair sponsored by ECCO, the Educational Computer Consortium of Ohio, on Thursday and Friday, October 11-12, 1990, at the Cleveland Marriott East Hotel in Cleveland, Ohio. These two conferences were combined to celebrate ECCO's tenth anniversary

The theme of the conference is CELEBRATING TEN YEARS OF EDUCATIONAL COMPUTING. There will be two strands of Logo sessions during the two days of the conference. Featured speakers include Sharon Yoder, Gary Stager, Dorothy Fitch , and Michael Tempel.

Two day-long post-conference workshops are scheduled for Saturday, October 13. They are Logo Connections taught by Sharon Yoder, and Lego-Logo taught by Gary Stager. A gala "Party in the Flats" is planned for Thursday night.

The cost of the conference is $30 per day for ECCO members, and $35 per day for non-members. Post conference workshops cost $40 for ECCO members and $50 for non-members.

For further information, or to request the registration form, contact ECCO, 1123 S.O.M. Center Road, Cleveland, Ohio 44124, or call 216/461-0800.

What's Cookin'?

Here's a Logo program that will create delicious recipes, perfect for your next party! Load the RECIPE file from the disk by typing:

```
READ "RECIPE
```

Type

```
RECIPE
```

Type

```
RECIPE
```

again to get a different recipe. (Can you buy dragon feet or bat wings in your supermarket?)

These recipes get boring because they always use the same ingredients. That is because the words for the recipes are picked at random, and there aren't very many words to choose from right now. But you can add words to the program to make more interesting meals! There are ten different word lists you can change or add to. The words in capital letters are the names of the Logo procedures that store these lists.

The title for your recipe	TITLE
Types of animals to put in your recipe	ANIMAL
A list of animal parts	PARTS
Words used to measure things	MEASURE
Liquids for cooking and	LIQUID
Words that tell how to cook food	COOK
Other things you do to food	NONCOOK
Kinds of vegetables	VEGETABLE
A set of numbers	NUMBER
Units of of time	TIME

To change or add to these lists of words, you will edit the procedures that store them. To add more animals, for example, type:

```
EDIT ANIMAL
```

This instruction takes you into the editor. You will see the ANIMAL procedure that contains a list of animals.

Here are tips for adding new animal.words:

- Use the arrow keys to move up, down, left or right. Nothing on the screen will change when you use these keys.
- To add a word, move to a list and type the new letters inside the square brackets.
- To erase letters, move the cursor to the right of the letter and press the Delete key.
- Be sure to leave a space between words.
- Place square brackets around any pair of words that you want to keep together, like this: [DINOSAUR [GILA MONSTER] CROCODILE [GIANT SQUID]].
- Do not erase the words OUTPUT or PICK; the program needs them!
- Do not press the Return key in the editor; long lists will wrap around to the next line by themselves.
- When you are through, press Control-C to leave the editor and store your changes.

Now type

```
RECIPE
```

to see another recipe. Words are picked at random from your lists, so you may not see your new words right away!

You can save the RECIPE file with your new words by typing:

```
SAVE "RECIPE.DLM
```

Hint: Add your initials to the filename so that the new version doesn't erase the original file or get confused with other RECIPE files on the disk.

Challenge

Look at the RECIPE procedure and see if you can figure out how it works. Try making changes to it and then testing it. Remember that no matter what happens with this set of procedures, you can start fresh by typing GOODBYE and reading the original RECIPE file from the disk.

Questions Please?

by Frank Corley

This column of the *Logo Exchange* is particularly important to those of you who are just beginning to use Logo. As I write this, it is still July, so obviously I have not yet had any answers to last month's questions. I am waiting to see some of those soon, so get to work! These questions were taken from a collection sent to me after NECC in July from a small but dedicated, band of Logo users—some very experienced and some new to the language. Because this is the beginning of the year, I have tried to take questions from among that collection that might be of importance to teachers and students who do not necessarily have a lot of experience with Logo.

Next month, I plan on presenting a similar collection of "big questions," questions that will be hard to answer in just a few lines, I hope. Until then, I am looking forward to receiving some correspondence from the readers of this column. I need this correspondence for several reasons. First of all, these questions are very important to the people who have asked them and I would like to see them answered in a timely fashion. Secondly, my supply of good, thought-provoking, appropriate questions is not bottomless, so I need a steady stream of mail to provide the column with material, both questions and answers, both beginner-level and advanced. So, for my sake, but most of all for the sake of other *LX* readers: **ask! answer!**

1. In LogoWriter, how can I override the PRINTTEXT80 command to print 10 characters per inch and fill the page leaving one inch margins on both sides of the paper?

2. Please explain in simple language the LPUT command, how it works and how it is used.

   Editor's Note: I'm not that familiar with LogoWriter; in Terrapin, LPUT is analagous to FPUT, but the item is put at the back of the list. In standard LISP, there is no LPUT. Does this imply that in Logo, lists are implemented as doubly-linked lists instead of linked lists, as I believe they are in LISP. This still does not answer the question of good uses for LPUT. Are there natural applications of LPUT versus FPUT?

3. How does one adapt material for one version of Logo to materials for different version of Logo? Are there a few easy translation rules?

   Editor's Note: The biggest difference I've encountered among the several versions is the syntax of the IF-THEN and the IF-THEN-ELSE command. Many other commands are simply different keywords which have essentially the same meaning. A list of these "translations" would be helpful to a lot of people. I believe that there are much deeper, semantic differences in the versions of Logo—can anyone provide some information in this regard?

4. How do I make a full disk work?

5. In what sequence should the commands and concepts of Logo be taught to introduce programming in the language?

6. Are there some activities that are particularly well-suited for beginners to the language: both beginning teachers and teachers teaching to beginning students?

7. Can someone give a typical page of a Logo lesson plan, including goals and objectives?

8. Are there Logo institutes around the country for teachers and supervisors just beginning in Logo? Could this be a regular section of *LX*?

Until next month, I hope that all of your questions are answered, but those that aren't, sent them to me!

Frank J. Corley
St. Louis Priory School
500 South Mason Road
St. Louis MO 63141

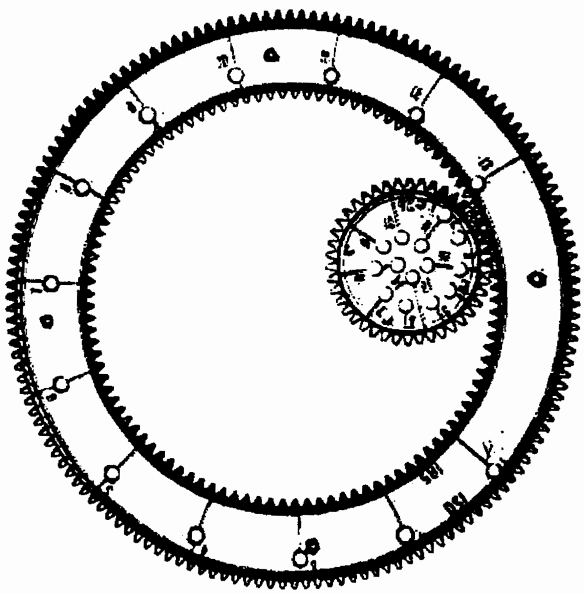
About the Cover

This cover this month is from the work of
Paula Sarver and her students.
See the article on the next page.

Logo Spirograph

by Paula G. Sarver

Children love to create beautiful pictures and designs, especially with unusual tools. Spirograph, an easily available toy produced by Kenner, is one such tool with which children can draw. The Spirograph set comes with four colored pens, two large rings, and many small wheels. By placing a wheel inside one of the rings and inserting the point of a pen into one of a series of holes in the wheel, the drawer moves the wheel around the ring creating loops and other colorful designs.



Spirograph uses the mathematical concept of the hypocycloid. A hypocycloid is the rotation of a small ring in contact with and within a larger ring. The drawing line is the locus or path of any point on the smaller ring as it rotates. Teachers can use Spirograph and Logo to help students intuitively explore hypocycloids, circles, least common multiples, and other mathematical concepts.

Introducing Logo Spirograph to Students

Students need to begin their exploration on a concrete level, especially for the difficult concepts introduced in this project. Therefore, I began by introducing Kenner's Spirograph to my third grade students. Many of my students were familiar with the Spirograph and the others quickly learned how to use it. A booklet is included in the package that shows a number of designs and combinations of ring, wheel, and hole position needed to recreate the design pictured. Most of the time, however, the students preferred to create their own. Once they gained experience with the tools, they were able to predict with some degree of accuracy what combination of ring, wheel, and hole they needed to create the design they visualized.

After a few weeks and several refills of the colored pens, I introduced them to my Logo version of Spirograph. There are some differences between Kenner's Spirograph and my Logo Spirograph but the students were able to transfer their knowledge with little trouble.

After running through the program together a few times, I let the students explore with Logo Spirograph until they were comfortable. Students quickly discovered the need for good recordkeeping. Without careful records, the students were unable to reproduce their favorite designs. Therefore, we developed the following chart for recording data:

	FIRST DESIGN	SECOND DESIGN
COLOR		
OUTER RING		
WHEEL		
NAME OF DESIGN		

After all students had several opportunities to create designs using Logo Spirograph, they were ready to draw some conclusions based on a study of designs and records of data. I placed the students in cooperative groups and together, they created and saved a design and made a record of their data using the above chart. After all groups were finished, I printed a copy of the group's design—one copy for each group member. On each copy, the students wrote the color, outer ring, wheel, and name of the design. I redistributed the copies so that each group had several different designs.

I asked each group to compare the designs and the data for patterns. One pattern that quickly emerged was the larger the outer ring number, the larger the design. At this point, I introduced the word "diameter" and we tested the students' observation by using rulers to measure the diameters of designs with different outer ring numbers and comparing the measurements. They then were able to begin predicting which numbers would create small and large designs and which numbers to choose if they wanted their second design to fit inside their first design.

Next, we tried to determine the relationship between the numbers for the outer ring and the wheel. Why were some designs so open and others almost filled? This is actually based on the least common multiple (LCM). The larger the LCM of the ring and wheel numbers, the more rotations the turtle has to make to draw a closed figure, and so the design is more filled. My third grade students could not quite reach this conclusion since they are just beginning to master their basic multiplication facts. However, they intuitively saw that the results were based on how well the numbers "matched." Some pointed out that numbers ending in zero were similar and so drew only "loops that were not filled." More observations were made, though many hypotheses were disproved by other different designs drawn by the turtle. I believe older students could have eventually seen the relationship between the outer ring and wheel numbers.

Older students may also be able to develop a basic understanding of how the designs are drawn by the rotation of the smaller wheel inside the larger ring– the concept of the hypocycloid. Like the younger students, it would probably be necessary to use the concrete tools of Kenner's Spirograph to act out the path that the turtle takes.

Whatever age the students are, Logo Spirograph gives them a means to intuitively explore the relationships between numbers, practice keeping careful records, and begin predicting the effects of numbers on the path of the turtle.

The Mechanics of Using Logo Spirograph

To use Logo Spirograph, the teacher must first enter and save the procedures found at the end of this article. The procedures are written in Terrapin Logo but can be adjusted for other versions of Logo. After typing in the procedures, create a startup variable by typing

```
MAKE "STARTUP [WELCOME]
```

and then save all of your workspace by typing

```
SAVE "SPIROGRAPH
```

The startup variable is saved along with all of the procedures, and when the file SPIROGRAPH is called by the students, the procedure WELCOME is automatically executed. This keeps to a minimum the amount of typing the students must do to begin the program.

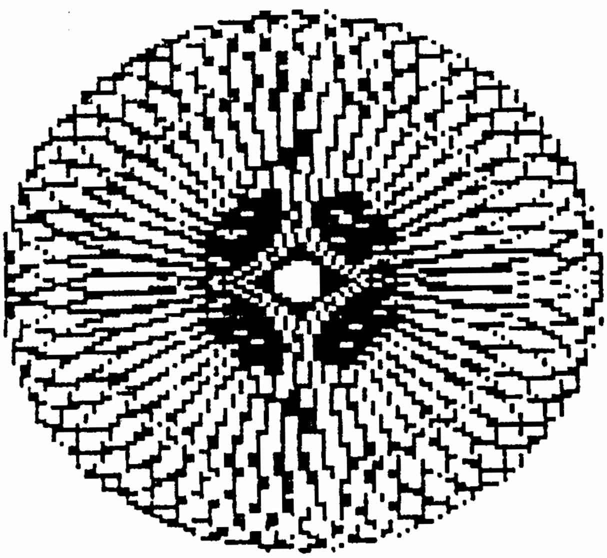
The teacher should also format several data disks in advance to store the students' designs. Since the designs are saved as pictures, which use more storage space than procedures, a class may use four or more data disks easily.

Once the procedures and startup variable are saved, Logo can be used to create new Spirograph designs. The students must first load the Terrapin Logo Language Disk and then insert the disk on which the teacher has saved the Spirograph procedures. After the students type

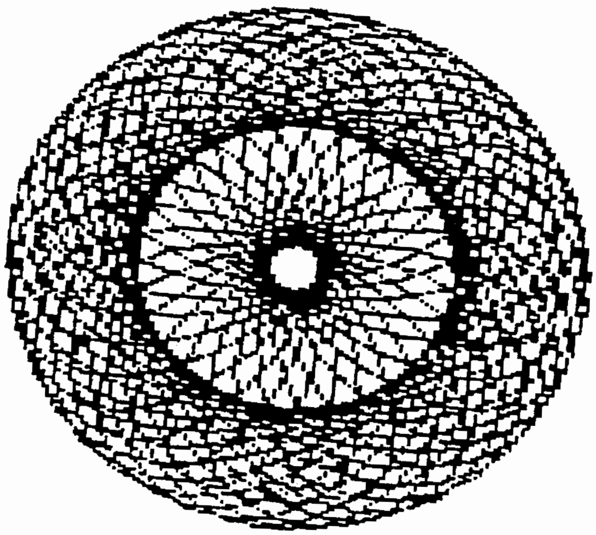
```
READ "SPIROGRAPH
```

the program is loaded and begins automatically. After the students have created their design, they are given the opportunity to save it. It is important for the teacher to tell the students that Logo does not recognize spaces in the names of saved pictures. The students may divide words using periods if they wish, but not spaces.

The listing on the next page is of a *streamlined* version of my program. If you are interested in receiving a copy of my original program (which is more user friendly), instead of retyping this shortened version, send a blank, formatted disk and $3.00 to cover mailing costs to the address at the end of the article.

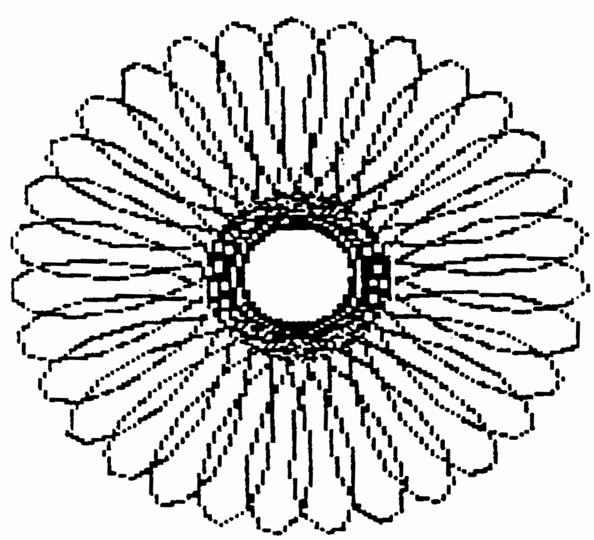


Logo Spirograph - continued



Procedures for Logo Spirograph

```
TO WELCOME
COLOR
RING
WHEEL
PRINT [WHEN YOU ARE READY TO SEE YOUR]
PRINT [DESIGN, TYPE D]
INTERPRET RC
PATTERN :RRING :RWHEEL
CLEARTEXT SPLITSCREEN
CURSOR 0 20
PRINT [WOULD YOU LIKE TO DRAW ANOTHER DE-
   SIGN?]
PRINT [TYPE YES OR NO.]
TEST [YES] = RQ
IFTRUE DESIGN2
IFFALSE SAVE.PICT
END

TO PATTERN :RRING :RWHEEL
HIDETURTLE
PENUP
SETXY 0 :RRING
PENDOWN
MAKE "CENTR.ANGLE 0
MAKE "POINT.ANGLE 0
MAKE "RATIO :RRING / :RWHEEL
MAKE "R.ANNULUS :RRING - :RWHEEL
MAKE "COUNTER 0
MAKE "NUMBER 0
LCM
MAKE "MAXTIME :FACTOR * 36
ROUTINE
PENUP
HOME
END

TO LCM
MAKE "NUMBER :NUMBER + 1
MAKE "MULTIPLE :RRING * :NUMBER
TEST REMAINDER :MULTIPLE :RWHEEL = 0
IFFALSE LCM
IFTRUE MAKE "FACTOR :NUMBER
END

TO READNUMBER
OUTPUT FIRST REQUEST
END

TO RING
PRINT [THE SIZE OF YOUR RING WILL DETERMINE]
PRINT [THE OUTER LIMITS OF YOUR DESIGN.]
PRINT [PRESS ANY NUMBER BETWEEN 30 AND 110]
MAKE "RRING READNUMBER
TEST :RRING < 30
IFTRUE ERROR2
TEST :RRING > 110
IFTRUE ERROR2
END

TO WHEEL
PRINT [TYPE IN THE NUMBER YOU CHOOSE AS]
PRINT [THE SIZE OF YOUR WHEEL.  THIS]
PRINT [NUMBER MUST BE SMALLER THAN THE]
PRINT [SIZE OF YOUR RING.]
MAKE "RWHEEL READNUMBER
TEST :RWHEEL > :RRING - 1
IFTRUE ERROR
END
```



```
TO INTERPRET :CHAR
IF :CHAR = "B PENCOLOR 5
IF :CHAR = "O PENCOLOR 4
IF :CHAR = "V PENCOLOR 3
IF :CHAR = "G PENCOLOR 2
IF :CHAR = "D FULLSCREEN
IF :CHAR = "N CLEARSCREEN TEXTSCREEN
END

TO GOODBY
TOPLEVEL STOP
END

TO ERROR
PRINT [YOU NEED TO CHOOSE A SMALLER]
PRINT [NUMBER FOR YOUR WHEEL.]
WHEEL
END

TO COLOR
PRINT [YOU CAN CHOOSE ONE OF FOUR]
PRINT [PENCOLORS.  PRESS THE LETTER]
PRINT [OF THE COLOR YOU WISH.]
PRINT [B=BLUE V=VIOLET O=ORANGE G=GREEN]
INTERPRET RC
END

TO ROUTINE
IF :COUNTER > :MAXTIME STOP
MAKE "POINT.ANGLE 360 - :CENTR.ANGLE *
    :RATIO
MAKE "X.CENTR :R.ANNULUS * SIN (
    :CENTR.ANGLE )
MAKE "Y.CENTR :R.ANNULUS * COS (
    :CENTR.ANGLE )
MAKE "X.POINT :X.CENTR + :RWHEEL * SIN (
    :POINT.ANGLE )
MAKE "Y.POINT :Y.CENTR + :RWHEEL * COS (
    :POINT.ANGLE )
SETXY :X.POINT ( :Y.POINT )
MAKE "CENTR.ANGLE :CENTR.ANGLE + 10
MAKE "COUNTER :COUNTER + 1
ROUTINE
END

TO ERROR2
PRINT [YOU NEED TO CHOOSE A NUMBER BETWEEN]
PRINT [30 AND 110]
RING
END

TO DESIGN2
CLEARTEXT
TEXTSCREEN
PRINT [YOU CAN NOW CHOOSE DIFFERENT]
PRINT [PENCOLOR, RING, AND WHEEL SIZES.]
PRINT [IF YOU WISH TO BEGIN A BRAND NEW]
PRINT [PICTURE, TYPE N.  IF YOU WANT TO]
PRINT [DRAW YOUR NEW DESIGN ON TOP OF]
PRINT [YOUR OLD DESIGN, PRESS RETURN.]
INTERPRET RC
COLOR
RING
WHEEL
FULLSCREEN
PATTERN :RRING :RWHEEL
SAVE.PICT
END

TO SAVE.PICT
CLEARTEXT
SPLITSCREEN
CURSOR 0 20
PRINT [DO YOU WISH TO SAVE THIS PICTURE?]
PRINT [TYPE YES OR NO.]
TEST [YES] = REQUEST
IFFALSE GOODBY
IFTRUE PRINT [INSERT THE LOGO DATA DISK,]
PRINT [TYPE A NAME FOR THIS PICTURE,]
PRINT [AND THEN PRESS <RETURN>.]
MAKE "NAME FIRST REQUEST
IF WORD? :NAME THEN SAVEPICT :NAME ELSE
    SAVE.PICT
GOODBY
END

MAKE "STARTUP [WELCOME]
```

Paula Sarver has been teaching for eight years in the Archdiocese of Louisville in Kentucky. She has been teaching third grade for the last five years and is the computer coordinator of her school, Saint Francis of Assisi.

Paula Sarver
St. Francis of Assisi School
1938 Alfresco Place
Louisville, Kentucky 40205

Logo Ideas

The Name's the Project, POS(ition) Is the Game

by Eadie Adamson

LogoWriter's turtle-move keys are great for beginners. Beginners use these keys so easily that they can be a nice lead-in to learning some important Logo commands: show, pos and **setpos**. As I watched an eager beginner last year working on his project to announce himself on screen, I observed that I could teach him some new commands and thereby enhance his project enormously.

A Neat Project Begins

Jerry (not his real name) had already created shapes for the letters of his name, plus shapes to say "I AM." His idea was to make a page that said "I AM JERRY"—writ large, of course, by changing the turtle's shape and sometimes color, moving it with the turtle move keys, and then stamping each letter in place.

Here was an opportunity to teach several things in a single context that matched beautifully with Jerry's own agenda. Jerry's project was an excellent lead-in to using screen coordinates for placing the turtle and then writing procedures that could eventually do this all with a single command. Thus there was the potential to learn about several new primitives, learn about writing modular procedures, and to learn about using superprocedures as well.

It is important for teachers to observe children's work carefully. Teachers can assist learning greatly by observing ways to advance each student's programming skills. However, teachers must be careful not to impose their own ideas on the project, but rather to allow the student to maintain ownership of the work.

Laying the Groundwork

Jerry had already begun by creating his shapes and making a key list of shape numbers and their corresponding letters. He had also (wonderful perception!) realized that he needed to make only one "r" shape in order to stamp his name on the screen. This is a trick students often miss when they try similar projects: the turtle can change it's "costume" multiple times and so a single shape can be used and reused even by four turtles at once.

Extending Knowledge

We began with the letter "I."

Task one: Using turtle-move keys, put the turtle wearing the letter "I" in the position on the screen where "I" is to be. Press Esc.

Task two: Get LogoWriter to show where the turtle is. Type: **show pos. Show** is a command that displays information in the Command Center. **Pos** is the reporter that gives the current position of the turtle. It reports the current coordinates for the turtle as a list of two numbers, the first representing the x-direction, the second the y-direction. Typing **show pos** in the Command Center and pressing return will display [0 0] in the Command Center when the turtle is in the home position at the center of the page. When the turtle is moved to other places on the screen and **show pos** is typed again, the numbers will be different, reflecting just where the turtle is in the x and in the y directions. Jerry jotted down his numbers for the position for "I."

Task three: Write a procedure for the letter "I" that changes the costume the turtle is wearing, changes color if necessary, picks the pen up, moves to the right position on the screen, puts the pen down, and stamps the letter in place. Here, the command **setpos** is used with the list of numbers reported in Task Two, above. Example:

```
to i
pu
setc 2
setsh 2
setpos [-100 80]
pd
stamp
end
```

Following the same process, Jerry decided on the places for the word "AM" and wrote a procedure, **am**, for the two shape changes and moves. He did one other clever thing, however: Jerry used the Copy keys to copy the body of the "I" procedure. Using the Paste keys, he pasted it twice in the "AM" procedure. Then Jerry simply changed the shape numbers, the colors, and the positions. Already the project was taking shape!

Jerry could now type "I am" in the Command Center and the two words would pop up on the screen! Now he was eager to finish the project and quickly found the other positions for the letters to spell his name. He could still paste the parts of the "I" procedure on the flip side of the page, so he quickly put together a **jerry** procedure and made the necessary adjustments.

Adding Some Dynamics

Here I asked a question: "Now that you can type "I AM JERRY" and get that to appear on the screen, can you write a procedure so that all that happens when you type a single word, perhaps just **me**?" Jerry thought a bit and remembered all that talk I had done earlier about superprocedures. It's a big word but Jerry liked big words.

"A superprocedure?" he asked. "I guess that might work." He set to work, and indeed it did work.

We looked at the display.

"How," I asked, "could you add some suspense to that? So that all the words don't seem to appear at once?"

"**Wait!**" cried Jerry. "I'll add **wait** in between!"

Jerry hit a snag. An error message popped up: "**wait needs more inputs in I.**" **Wait** is a command which requires an input for the length of time. With Apple IIe or GS running on slow speed, a **wait** of 20 equals one second. Jerry had forgotten to tell LogoWriter how long to **wait**, but this was easily fixed. He realized, of course, that he had also left out the inputs in all the other procedures too.

Now came the interesting part. It became clear that it was possible to play around with time in this project. The first word could appear, there could be a pause, the second word could appear slowly, there could be a longer pause, and then the name could appear ... sound of trumpets to accompany? Well, not quite, but you could add sound!

Jerry's final project was fun to see and turned out to be an excellent way to create a classroom expert on **show pos**, **setpos**, and **wait** as well!

This seems to be a great project to try with a new class. Suppose, as a teacher, you don't know everyone's names yet. What a neat way to teach some important concepts and give you a chance to get to know the students' names too!

Extensions for the adept

- Make the name flash. With a four-letter name, it's easy to have the turtles show and hide. With more letters, write a procedure to have the shapes to erase, then reappear.
- Copy the procedure that stamps the name. Change its name and then change the procedure so that the turtle uses **pe** (for penerase) rather than **pd**. Decide how to combine them, how long to **wait**, whether to change colors each time, and whether to finish with the name on the screen or with a black screen (**rg ht**).
- Can you make your display work like an LCD sign? (Look carefully at a calculator or digital clock display.) Try using multiple turtles to stamp the appropriate piece of the symbol in sequence, clear, then move over and stamp in sequence again.
- Can you use the above idea with multiple turtles simply appearing and disappearing in the appropriate place? Use the idea to create great commercials or titles for interactive reports.

Share your inventions with us!

Eadie Adamson is a Project Collaborator and Consultant for The New Laboratory for Teaching and Learning at The Dalton School, where she teaches Middle School computer courses.

Eadie Adamson
The Dalton School
108 East 89th Street
New York, NY 10128

Logo LinX

Turtling Around in Circles
by Judi Harris

"There are 360 degrees in a circle."

Did you have to memorize this in elementary or junior high school? Did you have any idea how exactly 360 of anything would fit into a circle of any size? Did you question your well-meaning teacher, only to become more confused by the well-intentioned reply? And later, in high school, did you "understand" several other geometric theorems that were presented to you, based upon what you already "knew": that a circle has 360 degrees?

I did. Fortunately, my mother, who was a math teacher, had the good sense to help me to explore this confusing idea (even before Logo was a gleam in Papert's eye) by thinking about constructing a circle as a series of line segments and angles of uniform size. In that way, I also was able to see relationships of regular polygons to circles.

Many of my adult students don't have an opportunity to *feel* this powerful set of related ideas until they begin experimenting with Logo graphics. Their reactions are usually predictably delightful. "So *that's* what the 360 means!" "Wait'll I let my kids loose on this!"

As you probably already know, Papert calls this the *Total Turtle Trip Theorem:*

> If a Turtle takes a trip around the boundary of any area and ends up in the state in which it started, then the sum of all turns will be 360 degrees. (Papert, p. 76)

Just what makes teachers' eyes light up about this particular set of geometric ideas? First is probably their own personal reaction; a long-awaited "aha!" of comprehension for something that they probably memorized and probably didn't feel that they could fully explain in operational terms to students. But second is the intuitive recognition of a truly powerful idea: as Papert suggests, the students can actually use it, it is more general (it applies to squares and curves, as well as to triangles), and it is intelligible (its proof is easy to grasp). Children that experiment with this idea in the Logo environment will, as Papert says, "come to appreciate how certain ideas can be used as tools to think with over a lifetime." (Papert, p. 76)

The TTTT Applied

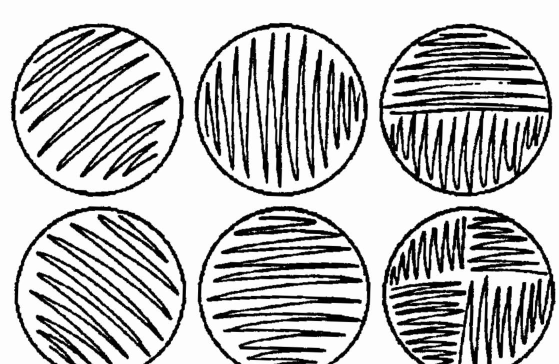
What follows is a particularly popular example of one genre of Total Turtle Trip projects explored with children aged 9 - 12. While taking one of my frequent jaunts through a local bookstore, I happened upon one of the more recent of Ed Emberley's drawing books, *Picture Pie: A Circle Drawing Book.* Prolific Mr. Emberley has produced more than a dozen "how to draw" books still in print that are excellent Logo idea sources for catalyzing procedural code writing. He breaks each drawing down into a logically sequenced series of "picture parts," showing each step in the process in comic strip format. So, too, did he display the how-to's of a magnificent array of designs in his circle book.

The central idea here is to create all images literally using pieces of circles of different sizes. Once the circles themselves are cut from different colors of paper, the only other cuts that should be made are along the radii of the circles, so that each piece looks like a slice of "fraction pie." It's amazing how many different images can be created by varying circle size, portion size, color, material, and placement.

Turtled Circumambience

Picture Pie suggests that students use scissors and colored paper to help them create circle creatures; my third-grade Logo students went one step further. After planning their designs with colored paper pasted to grids (which represented each quadrant of the SETPOS screen), they went about creating their circle sections designs with the turtle. They then decided to create a cooperative class "circles quilt" with the completed designs.

They told me that one of the reasons why this particular structure was so appealing was that they had been playing with the "other part" of circles (the circumference) for so long. As an introduction to the programming part of this project, I drew several circumferences on chart paper for them, and asked how they might color in the circles if they were the turtle and had no FILL command to use. They generated many different methods, such as:



...until one student thought of starting in the center of the circle, rather than along the circumference. Much in the same way that we as teachers immediately recognized the power of the TTTT, a collective gasp, then clamor of voices testified that this was the most powerful (simple, generally applicable) way of coloring in a circle without the FILL tool. The procedure almost wrote itself:

```
TO FILL.CIRCLE.OF :RADIUS
REPEAT 360 [FORWARD :RADIUS BACK
   :RADIUS RIGHT 1]
END
```

(Some of the students were already familiar with the words *circumference* and *radius*. I suggested the syntax for naming the procedure.)

How, then, to draw the portions of circles used so cleverly in Ed Emberley's picture pie designs? It was almost immediately apparent:

```
TO FILL.HALF.CIRCLE.OF :RADIUS
REPEAT 180 [FORWARD :RADIUS BACK
   :RADIUS RIGHT 1]
END

TO FILL.QUARTER.CIRCLE.OF :RADIUS
REPEAT 90 [FORWARD :RADIUS BACK
   :RADIUS RIGHT 1]
END
```

...and so on.

Yes, of *course* someone suggested a "generic" FILL.CIRCLE procedure (after some circle pie experimenting, naturally):

```
TO FILL.CIRCLE :SLICE :RADIUS
REPEAT :SLICE [FORWARD :RADIUS BACK
   :RADIUS RIGHT 1]
END
```

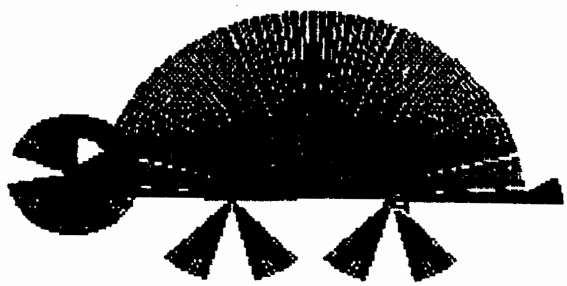
And someone else decided to make a randomly generated circle slice procedure:

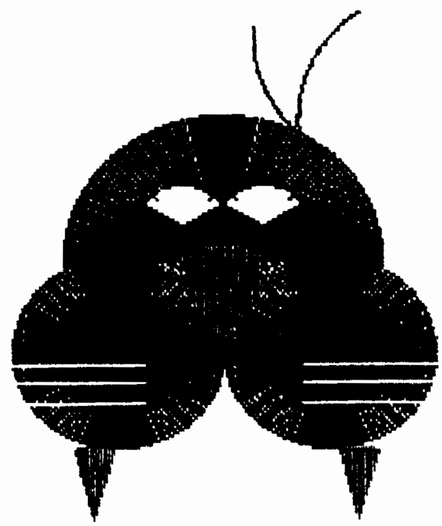
```
TO DRAW.RANDOM.CIRCLE
SETH (RANDOM 360)
PU FD (RANDOM 100) PD
SETC (1 + RANDOM 5)
FILL.CIRCLE (RANDOM 360) (RANDOM 120)
END
```

Sample Use:

```
REPEAT 30 [DRAW.RANDOM.CIRCLE]
```

Others preferred not to leave the results of their circle composition efforts to chance.





Powerful Resources

Other how-to-draw books by Ed and Michael Emberley are terrific sources for Logo project ideas, since all sketches are presented first as completed images, then in modular, sequenced steps to show how they were created. Here is a partial list of these helpful procedural graphics resources:

Drawing Book of Animals
Big Green Drawing Book
The Orange Drawing Book
Big Purple Drawing Book
Ed Emberley's Drawing Book: Make a World
Ed Emberley's Drawing Book of Faces
Ed Emberley's Great Thumbprint Drawing Book

Logo Linx - continued

Ed Emberley's Book of Crazy, Mixed-Up Faces
Ed Emberley's Little Drawing Book of Birds
Ed Emberley's Little Drawing Book of Farms
Ed Emberley's Little Drawing Book of Trains
Ed Emberley's Little Drawing Book of Weirdoes
Dinosaurs!: A Drawing Book
More Dinosaurs! and Other Prehistoric Beasts: A Drawing Book

Enough *circum*locution! Now it's time for you and your students to help the turtle to *circum*navigate the screen in new ways!

(An earlier version of this column appeared in a 1984 issue of the Montgomery County, PA Intermediate Unit's newsletter, *Interface*).

References

Emberley, E. (1984). *Picture pie: A circle drawing book.* Boston: Little, Brown and Company.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books, Inc.

Judi Harris works in the Department of Teacher Education at the University of Nebraska, Omaha as an assistant professor of educational technology. Her teaching, research, and service interests include Logo (of course), computer-mediated educational telecommunications, hypermedia, multimedia, and children's computer-assisted artwork.

Judi can be reached at:
CIS: 75116,1207

Thinking Like the Turtle

by Bert Eliason

How should we teach Logo?

For the past six years, I have used Logo with students in middle school. I have always *told* them how to make designs and how to make improvements in their work. The class usually progressed at a fairly rapid clip, but *I* was the one telling them how to make circles or how they could write procedures. Actually, I was telling them how to do just about everything! The more I read about Seymour Papert and his work with Piaget, the more I came to realize that I wasn't really asking my students to use their brains—to think for themselves.

This term I decided that I would try encouraging my students to *really* think. My original hope was that I would be able to discern enough of a difference to decide which style I would continue to use in the future. What I found was unexpectedly exciting and has convinced me to change my teaching style when I introduce and use Logo in the future. I hope that this article renews your faith in what you are doing or encourages you to consider asking students to "think like the turtle."

Background

The discussion that follows includes my thoughts about and anecdotal recordings of students who were in the process of being introduced to Logo. The students ranged from sixth through eighth grade and had varying ability levels and behavior patterns. The class is an elective that meets during the last period of the school day. The use of Logo described follows a three-day introduction to computers and a five-day course on BASIC programming. This is the first time most of these students have used Logo, though a few students have had some previous experience. We use a lab of 24 Commodore 64's, which means I do a lot of running around to answer questions, relieve anxieties, and share in student successes.

Day 1: Thinking Like the Turtle

I began by introducing Logo as a programming language that would enable the students to make things on the computer. I told about the beginning of Logo, the use of the floor turtle, and the eventual use of the turtle cursor. I told them that students of all ages have been very successful at using the turtle to draw and solve problems like the ones I would be posing. I told them that we would have fun, but I would expect them to think. In fact, I would ask them to "think like the turtle." This caused some giggles. I smiled and added, "You'll see what I mean by that later. Right now, let's check out Logo."

We began exploring. They saw how to get started and were introduced to the long and short versions of the following commands: FORWARD, BACK, RIGHT, LEFT, PENUP, PENDOWN, PEN, and BACKGROUND. As I introduced each of these, I turned the students loose for short periods of time to explore on their own. They were excited by the ability to produce something on the screen of their own making. Many became especially excited about the use of color and "wrapping" to create impressive designs.

It became apparent that some students had at least some prior Logo experience. Their designs quickly became the object of adulation from the less experienced members in class. There was much sharing taking place. When one of the experienced students asked how to make commands "continue," I asked him what he knew that brought the idea to mind. He responded that he remembered a command that made things go a "bunch of times." He just couldn't remember how it worked. I explained the REPEAT command to him. This caught the attention of several other students. Soon, several were writing REPEAT statements. The hook was set. These kids were excited about what they were doing. In letting the students play, I had been fishing to see what questions would arise. I was excited when a big one hit.

Susan came up to me and said, "Mr. E., how do I make it go in a circle?" My big chance—I had to be careful not to blow my cool and let her know how excited I was, lest she shy away.

"Susan, this is what I was talking about when I said I would ask you to 'think like the turtle.' I want you to think this problem through as if you were the Logo turtle. That means you would be creating a line whenever you move."

She gave me one of those, "OK, here goes," smiles. I asked her to bear with me and we would think this one out together.

"Show me how you walk in a circle, Susan." In a few steps she made a circle. "That's good! Now tell me what you are doing as you make those movements."

"I walked in a circle," she answered unsure of what I was after.

"Right! Now tell me how you did that, but you can only use turtle commands to tell me."

"All right." She looked excited and continued, "let's see, I...went forward a little, ...back some, and ...then forward again."

"Great," I remarked, "will you walk through it now as I repeat what you have told me?"

As soon as I said, "...back some," a look of great discovery appeared on her face.

"Oh... that won't work!" she blurted, looking up quizzically.

"No, but it was a pretty good try. Let's do it again," I answered.

She started, "I went forward and turned... and...turned and turned," she spoke her voice speeding up as if she had figured it out now.

"OK, let's try that one." She tried to keep walking as I repeated the second "...turn..." Expecting this, I said, "Whoa!"

Before I could get another word out of my mouth, Susan looked up at me and tried to fix what was wrong. "I went forward and turned...and then went forward...and then turned again, huh?"

"That's right!"

"So its forward, turn, forward, turn, forward, turn...?" she asked.

"You got it, Susan! I'm impressed!" I was excited, and students who were watching saw my excitement. So did Susan.

"Yeah, I did!" The bell rang. Susan was still working her discovery through in her head. Standing there looking stunned, a smile started to cover her face. You could see the wheels still turning as she looked up. "I get it!" she exclaimed.

"You better go, but tomorrow I'll work on that circle with you." She left smiling.

I had never been as excited about Logo as I was at that moment. Later, I felt that same excitement when I replayed the scene for another math teacher. Susan had "debugged" her own circle program. She had found what didn't work and had gone about fixing it. As she walked the circle, she could see that the commands had to be given in a specific order. It was logical to her sense of space, her sense of being. Susan's debugging sequence had given me my best understanding of Seymour Papert's use of the term syntonic learning.

Thinking Like the Turtle - continued

Papert talks about "syntonic learning" as learning that goes together with that which the child already knows. He asserts that children come to school with a knowledge of "body geometry." They know how to move their bodies through space, how to turn, move forward and backwards, and more. This problem was body syntonic. Susan was able to debug the command sequence, because she could see what made a circle when she walked in a circular path. It went together with and was related to that which she already accepted and understood about her own body's movement through space.

Use of the turtle is also ego syntonic. It correlates with a child's sense of self as a person with desires, goals, and values. It was Susan who had posed the question about the circle. There was value to her in debugging the problem. It was satisfying to her because it was something she wanted to know and it was something that she had worked to figure out.

Papert adds, "Every act of syntonic learning is learning to keep the new structures in touch with the old ones." (Taylor, p. 209). If students have learned based on something that is already part of their everyday life, something to which they can relate and do without thinking, it follows that they might have a deeper understanding of this newly learned material. It also makes sense that if a student has a need to learn something or values something that they have learned, they are likely to retain it for a longer period of time.

What I saw this first day was much different than what I have seen when I used Logo in the past. It was exciting and powerful. I liked what I saw.

Day 2: The Power of REPEAT

I reviewed the Logo primitives we had covered on the first day and then proceeded to introduce the REPEAT command to those who had not already learned it. I quickly found myself wanting to take much more control than I had originally planned to take during these lessons. A majority of the students wanted to continue the pursuit of wrapping to make more elaborate random designs. I stopped class long enough to have them understand that I wanted to see them using the REPEAT command. I assured them that using this command would then allow them to make even better designs than they were currently producing. As they returned to work, I realized that they either weren't comfortable with the use of REPEAT or they weren't listening. Many continued to produce line wraps of varying colors. I decided that it may be that they just weren't ready to move on. With this in mind, I backed off and allowed them to continue at their own pace.

One student, who was working on using REPEAT, asked if it mattered what went into the brackets. "Of course it matters," I thought, but said, "Well that depends on what you want to make." The student made it clear that all he wanted to know was whether you could put anything inside and still have it work. "Try it! Let's see what happens." His first creation took some big moves back and forth and even had some turns written consecutively, but it made a rather dynamic design that won some acclamation from his bank of fellow explorers. Soon, others were interested in knowing how he did this. At that point, I recall thinking the word "empowering."

The ability to create something that others can enjoy is a powerful tool. Throughout the ages, we have revered those who could use their skills to create that which is pleasing to our fellow man. Children are no different. They want others to notice their work, but as they begin to create, they notice or someone notices for them that their creations don't actually "look" the way they "should." Often they become hesitant to share, having lost confidence in their ability to create. They begin to seek, from those they can trust, an affirmation that their work is acceptable, thus giving up on whether or not it is acceptable to themselves. I have an art class where students will continually ask whether or not their current work is "good." I think working at the computer can separate the student from the idea of good and bad. Perhaps they view their product as the computer's work, and they are somehow distanced from from the right or wrong of how it "should look." At the keyboard, the child has an opportunity to create. Sometimes these creations become the center of their peer's attention. When this happens, and the child is asked, "How did you do that?", it is both an exciting and powerful experience for the creator. In their control is the ability to share something that others may not have yet discovered.

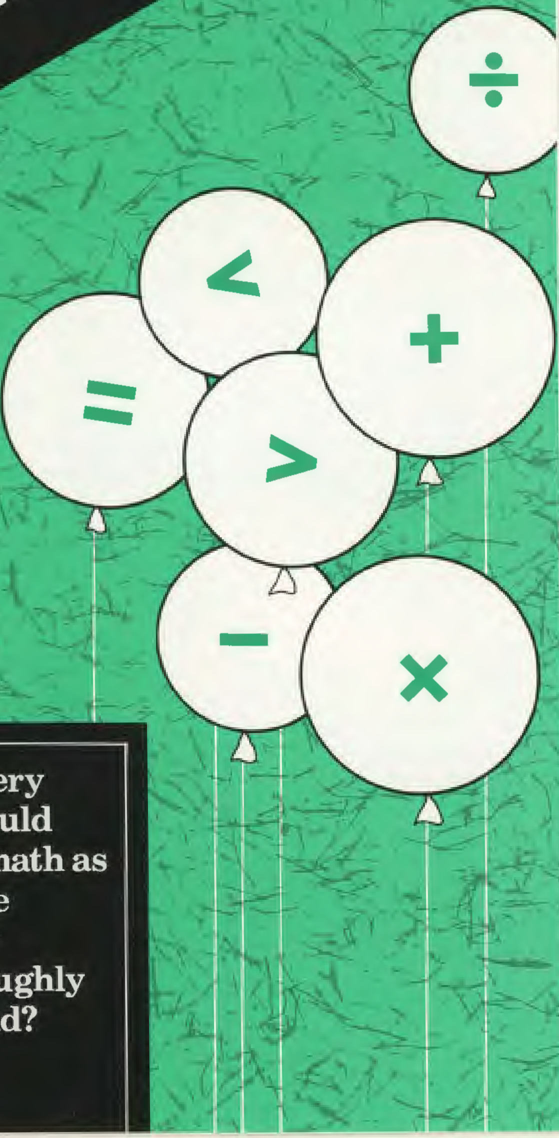
I approached Susan about working on the circle, but having seen the commotion caused by the student using the REPEAT command, she preferred to try random designs. I felt it was best that she experience what was important to her at that time. I knew that eventually I would be able to help her put the series of "forward and turn" commands into a REPEAT statement to complete her circle.

"Mr. E, can you help me?" Ann asked. "I want to make a triangle?" I looked at the clock and realized we would run over if I helped her. I asked if she could stay after school, but added that if I was going to help her, she had to be able to think like a turtle. She laughed and said she would stay.

We walked through a square routine, so that she could identify the four sides and four turns. We did some 180° turns

It's programming,
It's spelling,
It's...

FUNdamentallyMATH™



What if every student could embrace math as a language they enjoy and thoroughly understand?

A unique method of teaching programming

The programming manual has the student examine the effect of a programming technique by looking at and often changing parts of a pre-existing math procedure.

Learning is constantly related to doing!

How short and long division may be taught by the same method!

A revolutionary technique permits the teaching of both short and long division with the same program. The consistent use of the same step-by-step instructions permits an easy adjustment to long division.

The problem is: 5672 ÷ 85

Estimated by

90 | 5672

↑ ↑

Look at the number as far as the arrow.

The problem is: 5672 ÷ 8

Estimated by

8 | 5672

↑ ↑

Look at the number as far as the arrow.

Apple versions:

FUNdamentallyMATH™ works on an Apple IIE, Apple IIC, Apple IIC Plus or an Apple IIGS with at least 128K (uses Apple LogoWriter 2.0). The 128K version has a limited spelling tester, but is substantially identical to the other versions in every other procedure. A version without any such limitation just for the Apple IIGS requires 700K (uses Apple LogoWriter GS).

IBM version:

The IBM version of ***FUNdamentallyMATH***™ requires an IBM or compatible computer with at least 320K and a color graphics adapter. It has a full spelling tester. This version uses IBM LogoWriter 2.0.

Partial list of procedures

Primarily for grades 3 to 8, ***FUNdamentallyMATH***™ is being used in high schools for review and remedial work.

NAME OF PROGRAM	EXPLAINS THIS TOPIC
ROOT	VARIABLES, LIMITS
GUESS	BINARY SEARCH
SR (**S**how **R**aise)	RECURSION
COUNTUP	RECURSION
UP. DOWN	RECURSION
SA (**S**how **A**mount)	RECURSION
RAISE	POWERS
AMOUNT	COUNTING
A1 (**A**lgebra 1)	Operation on an Equality
A2 (**A**lgebra 2)	Operation on an Equality
A3 (**A**lgebra 3)	Operation on an Equality
A4 (**A**lgebra 4)	INVERSE OPERATIONS
A5 (**A**lgebra 5)	INVERSE OPERATIONS
A6 (**A**lgebra 6)	INVERSE OPERATIONS
AREA	MULTIPLICATION, AREA OF A RECTANGLE
TRI	AREA OF A TRIANGLE
WHY.TRI	WHY AREA OF A TRIANGLE = ½bh
HIT	DEGREES
XY	COORDINATES
CIRCLES	DIAMETER, π
TANG (**TANG**ent)	RADIUS, TANGENT
PI . PARTY	π
CIRCLE	CIRCUMFERENCE
LETTER	BUSINESS LETTER
LD (**L**ong **D**ivision)	Short and Long Division
MATH	ORDER OF COMPUTATION
MN (**M**ixed **N**umbers)	ADDING FRACTIONS
X	MULTIPLICATION
M	MULTIPLICATION
DIV	DIVISION
BAT	DECIMALS
S1111 (**S**how **1111**)	NUMBER SYSTEM BASES
SD (**S**how **D**ivision)	DIVISION AS A CONCEPT
SN (**S**how **N**umber)	NUMBER SYSTEM
CMP (**C**o**MP**ound growth)	COMPOUND GROWTH
SMP (**S**i**MP**le Interest)	SIMPLE INTEREST
GS (**G**row **S**quare)	COMPOUND GROWTH
PRIME	PRIME NUMBERS
PRIME?	PRIME NUMBERS
METRIC	ENGLISH VERSUS METRIC
NL (**N**umber **L**ine)	NUMBER LINE
MENU	NUMBER LINE
CHOICE #1	ADDITION
CHOICE #2	ADDITION, SUBTRACTION
CHOICE #3	MULTIPLICATION
CHOICE #4	ADDITION, SUBTRACTION
CHOICE #5	MULTIPLICATION, DECIMALS
CHOICE #6	MULTIPLICATION, DECIMALS, DOUBLE NEGATIVES
LINES	COORDINATES, ALGEBRA OF LINES
OWN . LINE	ALGEBRA OF LINES
LCD	LEAST COMMON DENOMINATOR
SORT	SORTING
SORT. DICT	Sort the SPELL page dictionary
SPELL . HELP	SPELLING TESTER
P	POLYGONS
SEE	POLYGONS, VARIABLES
SLOW	VARIABLES
TRK	VARIABLES
MAZE	PROGRAMMING PROJECT
RECTANGLE	RECTANGLES
SQ	SQUARES
SPIRAL	VARIABLES
S (**S**how Spiral)	VARIABLES
FUNHOUSE	COMMUTATIVE PROPERTY, FACTORING

*To use **FUNdamentallyMATH**™ you also need LogoWriter®*

Schools may preview ***FUNdamentallyMATH***™ for 30 days without obligation.

For more information write or call:

1-800-MATH-MAP (1-800-6284-627)

Chip Publications, Inc.

SCHOOLS

Use *FUNdamentallyMATH*™ in your class for 30 days without obligation! Mail this postage-free card now to obtain your trial copy!

Yes, please send me a sample copy of ***FUNdamentallyMATH*™** that I can use in my class for 30 days without obligation. I understand that to keep ***FUNdamentallyMATH*™** I will pay the invoice.

☐ ***FUNdamentallyMATH*™** Building Site License* **$195**

☐ ***FUNdamentallyMATH*™** Single Computer License* **$89**

Name ______________________________
(please print clearly)

Title ______________________________

School ______________________________

Address ______________________________

City ____________________ State __________ Zip __________

Phone Number ______________________________

***Requires corresponding LogoWriter® license.**

Schools must contact Logo Computer Systems, Inc. to purchase a LogoWriter® school package. 1-800-321-LOGO

Shipping in US paid
Outside US $5
NC schools add 5% sales tax

HOME

Yes, I want to purchase a copy of ***FUNdamentallyMATH*™**.

☐ Home Version of ***FUNdamentallyMATH*™** **$89**
☐ LogoWriter Home Edition **$89**
☐ Both **$165**

Name ______________________________
(please print clearly)

Address ______________________________

City ____________________ State __________ Zip __________

Phone Number ______________________________

We sell the Home Version of LogoWriter.
If you are purchasing the LogoWriter Home Edition from us, check version and disk type desired.

☐ Apple LogoWriter 2.0 (128K) ☐ Apple LogoWriter GS (700K) ☐ IBM LogoWriter 2.0 (320K)

Please check desired format: ☐ Apple 5¼″ ☐ Apple 3½″ ☐ IBM 5¼″ ☐ IBM 3½″

NC Residents add 5% sales tax.
Shipping, handling & COD charge (if any) $5

Also from Chip
WIZARD Typing™ tutor software
Check box for information ☐

Chip Publications, Inc.

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 548 CHAPEL HILL, NC

POSTAGE WILL BE PAID BY ADDRESSEE

Chip Publications, Inc.
P.O. Box 2161
Chapel Hill, NC 27515

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

Chip Publications, Inc.
P.O. Box 2161
Chapel Hill, NC 27515

Non-Profit
Organization
U.S. POSTAGE
PAID
Permit No. 63
Eugene, OR

An Invitation To Explore The Synergy Between Math And Programming

because she understood the command "about face." We added two of these up and she remembered having heard that a full turn was 360°. I told her about the Total Trip Theorem. Every time the turtle makes a complete revolution, a total trip, the turns will add up to be 360°.

"Ann, if you made four turns and your total turning equalled 360°, then how many degrees would each turn be?"

"I don't know, but I think you'd divide to find out."

"Great! We have a computer here, let's see if we can make it tell us how far we'd turn." I showed her how to use the slash sign to divide 360 by 4 and how to place it in a command such as RIGHT 360/4. Then an interesting thing happened.

I said, "OK, let's use that in a REPEAT command to make a square." She produced this statement: REPEAT 4 [FORWARD 50 RIGHT 60].

"Like that?" she looked up at me.

"Try it!" I encouraged, and she did.

"Uhh..What happened?" she asked.

It took a while to debug her statement. We walked through the box shape, discussed the turns and the Total Trip Theorem, and reviewed how to have the computer divide 360 by 4. She looked at her command REPEAT 4[FORWARD 50 RIGHT 60]. Suddenly she said, "Oh 90—it's supposed to be 90°."

Try as I might, I couldn't figure where she got the 60 from originally, nor could she explain to me why it had become fixed in her mind. We went back to the computer where she promptly wrote REPEAT 4[FORWARD 50 RIGHT 90]. I love the look of success on a child's face. She just stared and smiled.

We moved on to the triangle. We walked through the figure until she realized that this time she would need to repeat three times: forward something and turn something. I asked her to type:

```
REPEAT 3[FORWARD 50 RIGHT 60]
```

Ann ran this and could see that it didn't work. She was anxious to debug it. By this time, she was as hooked on figuring this out as I was on helping her.

"Back to the box, Ann. How many sides in a box? How many turns? How many degrees in a full turn all the way around? How many sides in a triangle? How many turns in the triangle? How many degrees in a complete turn around? How many times would you have to repeat 'go forward something, turn something' to make a box? How many times would you repeat 'go forward and turn' if you were making a triangle?" All of these questions were met with correct responses. "Tell me again how you would find the number of degrees in four turns, if the turns equalled 360°?" Another correct response. "Ok, I want you to write a REPEAT statement to make a four-sided box, but this time I want you to show me you can make the computer do the dividing.

She wrote:

```
REPEAT 4[FORWARD 50 RIGHT 360/4]
```

"It works!" she was delighted, and honestly, so was I. "How come it works now?"

We talked about the pictures that had appeared previously. She could tell that the only thing she had changed in her command was the measurement of the angle. I asked her to show me what the 360/4 would be, but she said 90 without batting an eye.

"Oh 90! I get it, 90, because it turns 90°. What about the triangle then, would it be 90?" she asked, and then added, "Could you divide the 360 by 3?" As if she needed my permission to continue, she poised herself at the keyboard.

"Go for it!" I said. It was all that she needed.

```
REPEAT 3[FORWARD 50 RIGHT 360/3]
```

"Yes... it works!" she exclaimed, smiling as if she had just won a convincing victory over the computer.

She went on to make several polygons and found she could make them all using 360 divided by the number of sides in the given polygon. I never did figure out why she got stuck using the number 60 on both the square and the triangle. Instead I started thinking, "How do I get small classes and an opportunity to work on something like this daily?" Maybe someplace in the future, our middle school schedule will recognize the benefit of a coordinated full-time computer position. When that happens, I hope I'm somewhere nearby.

Day 3: Confidence and Self Esteem

We continued working on using the REPEAT statement. I wanted each student to attempt to make a square, a triangle, a pentagon, and an octagon. Those that were doing well could try a decagon and a duo-decagon, after which they were free to make designs, combine figures, or explore new realms.

Thinking Like the Turtle - continued

Students were doing OK, and I was trying to help those that were having a hard time getting started. Most preferred creating designs by repeating patterns a large amount of times. This usually produced doughnut shaped objects with the outside edges being spread out in fan shape fashion. These were obviously not the polygons I'd asked for, but there is something to be said for creating from the desire to create as opposed to creating to perform for someone else.

We should remember here that ultimately Logo serves as a tool to be used by students to solve problems. If the problem is to make a decagon, then a student who knows what a decagon is and knows how to use Logo to some degree should be able to make a decagon. In this discussion, we are talking about a class that had only two 45-minute sessions with Logo. These students were still quite excited about exploring. We are also talking about a teacher who was a little bit anxious to see these kids produce. In the past, I would have spoon fed the whole class all at once. We usually moved along at a rate faster than this. I reminded myself that my objective was to see if I could discern a difference between the learning patterns of this class compared to those of previous classes.

George is a sixth grader who I see during three separate periods daily. He is in my low math class, an art class, and this computer class. Recently he has been showing a change in behavior, doing things in class that he wouldn't have done previously and generally acting out. I found myself being more stern with him than I normally am with students. I know his home situation is unpleasant and I imagine that something has been going on that would cause this change. George's good friend, Sam, is also in this computer class.

George and Sam were having a little trouble working with the assignment, but through most of the class period had been enjoying some splendid random design work. When I asked them about making a triangle using a REPEAT statement, they looked lost. I asked about a square and despite its command sequence already being on the chalkboard, they couldn't come up with an idea of where to start. I offered to help if they wanted to give it a try. Once again, since we were near the end of the class period, I asked if they would be willing to stay after. Their response was an excited "yes." I reminded them that if they wanted my help, I would expect them to try thinking like the turtle. They were adamant—they'd stay.

I sat between them, and we began. I'd give Sam and then George a problem to try. George needed to spend some time working through the turtle square much like Ann and Susan, but Sam quickly had the REPEAT format down. Once he saw that you could use the computer to do the division, he got busy making various polygons. He interrupted me as I helped George, to show me his "forty-one-agon." George took nearly thirty minutes to come to the point where he could produce a polygon with a certain number of sides. He was happy. Given any number he could now make these polygons just as well and as quickly as Sam. I was happy. I realized that his self-confidence was building again, and I hoped it would help with his acting out.

It had already been time well spent when I said, "Let me show you guys how to write a procedure." Once they saw that all they had to do to begin making a procedure was type "TO," they had little trouble figuring out how to make a procedure for building a triangle and a square. I then posed the classic Logo problem, "How could you use these two to make a house?"

George wrote the following procedure:

```
TO HOUSE
BOX
TRIANGLE
END
```

Of course he came up with a box with the triangle inside.

"How can I fix it?" he asked.

"Think like the turtle, George!" Sam said smiling. He proceeded to read George his command sequence for BOX and then for TRIANGLE. As George walked through the triangle, a look of discovery overtook his body.

"Oh, I get it. I need to move it up 50 spaces." He quickly fixed the procedure without being told how. Between BOX and TRIANGLE, he wrote FORWARD 50 and ran it. We all laughed to see his house with the roof on its side sticking straight up.

This time he used the screen to debug the procedure. He wrote:

```
BOX
FORWARD 50
RIGHT 90
FORWARD 50
TRIANGLE
```

He found his triangle on the top right side of his BOX, but the roof was now hanging off to the side, upside-down. This stumped him.

He chose to walk through it again. After crossing the top of his BOX, he could see that he needed to be turned around. It seemed to me, that while he was walking through the BOX, the RIGHT 90, and the FORWARD 50, he would have difficulty remembering the construction and directions in TRIANGLE. Yet, he walked the triangle with just a moment's hesitation. I believe all his practicing while working on the REPEAT statement previously had placed the making of the triangle firmly in his memory.

Neither of the boys could figure how far it was to turn around, but both knew that a full turn-around on a skateboard was called a "360." Sam asked if you could divide the 360 by 2. They both looked at me waiting for an answer. "We can try it," I said, and George wrote:

```
CS
BOX
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 360/2
TRIANGLE
```

and ran it. When he saw that it was going to work, he put it into his HOUSE procedure, then simply wrote one word.

```
HOUSE
```

"DUDE...RAD!" George uttered as he stared, his faced washed in a smile of amazement.

"These computers can do everything," Sam admired.

"Its not the computer that did this! The computer just sits here and does what you tell it. Your excitement should be aimed at each other. You are the ones that told it what to do."

They continued staring blankly at the computer. I reached over and cleared the screen. They looked up in amazement, so I smiled and said, "The nice thing about learning procedures is that all the work, that goes into building something can be forgotten, as you bring back your creation with the title you used to name it. Type 'HOUSE'."

George did and Sam said, "God, George, think about how long it took to make it the first time."

This time I asked, "George, do you mind?" He looked at me and shrugged his shoulders.

I wrote:

```
TO SPIN
HOUSE
PU
HOME
PD
RIGHT 10
SPIN
END
```

"Sam, type 'SPIN'," I smiled.

While the screen produced multiple houses while spinning around the axis, we recalled talking about programming loops. We admired our work. I admired their faces.

Together we shut down the lab and walked to their lockers. I hadn't seen that kind of excitement in George's eyes in a long time. If this project were to have done nothing else for me, it was worth the time reading about Papert and Logo and all this typing just to see George so involved. I wonder if he realizes he was doing math. I know he at least found a place where he could discover, gain confidence, and temporarily forget that part of his world that was troubling him.

Logo is a tool that teachers can use for bonding with students while students learn problem solving and math. Logo is body syntonic, ego syntonic, and maybe "bonding syntonic." That is, it goes together with building personal relationships. This wouldn't be the use Papert intended for "goes together with," but an argument could surely be made here. After the time I spent working with Susan, Ann, Sam, and George, I'd argue that one of the values of using Logo is found in helping students use what they already know to expand their knowledge. While doing this, the student-teacher relationship is growing.

Conclusions

I have used Logo for a number of years. I have never before seen students so quickly begin to "think like a turtle." Certainly my brief experiment this term covers such a short time that the cumulative effects might be debated. But, I am convinced that my students were more receptive. They actually worked to understand Logo, its uses, and its potential.

Here is a list of some of the pros and cons of using Logo as I did in my brief guided discovery unit.

Advantages

1. Students build from a desire, giving their creations *value to themselves.*
2. Students can relate what they are creating to their own body geometry.
3. Students solve through self-exploration, doing learning as opposed to hearing learning.
4. Students discover the importance of sequencing, instead of being told the importance.

Thinking Like the Turtle - continued

5. Students become empowered with their ability to solve problems, create designs, and the chance to share these with others.
6. Teachers will observe two of the things they most enjoy about teaching: watching students taste success and watching students discover through their own abilities.

Disadvantages

1. Students may not initially progress as fast as when more teacher direction is used.
2. Teachers may be frustrated by this lack of progress.
3. It is hard to observe immediately how well a student is internalizing this learning.
4. It may take more one-to-one teaching in order to fully impact a student initially.
5. Some students may not deal well with the concept of "thinking like the turtle."

Afterword

Many of these students did continue to design and create on their own during their computer time. Several became accustomed to walking the problem through, others would draw the problem on paper, and others still felt uncomfortable unless someone would "...just show me how." In the end, the class made at least as much progress as my other classes have. Some students, including George and Ann, excelled. Susan did go on to use circles in her designs. Most important was the fact that many students gained the confidence to be able to work out solutions on their own.

I hope the reader will be able to see why I have come to believe that asking kids to think, talk, draw, or walk their problems through is beneficial. Better yet, I hope readers who have not tried such a guided discovery approach with their Logo classes will do so. The payoff is well worth the the risk!

References

Grammer, V. C., Goldenburg, E. P., and Klotz, L. Jr. (1982). *The Commodore 64 Logo tutorial*, Cambridge, Massachusetts: Terrapin, Inc..

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc..

Taylor, R. P. (ed.) (1980).The *computer in the school: Tutor, tool, tutee*. New York and London:Teachers College Press.

Bert Eliason has been teaching middle school for 16 years and currently teaches math, science, and computers at Cascade Middle School for Bethel Public Schools in Eugene, Oregon.

Logo Connections

Creating a Hypermedia Adventure Story with Logo

by Glen L. Bull, Gina L. Bull, and Paula Cochran

In last month's column, we outlined several suggestions for enhancements that can be used to create a hypermedia environment with Logo. Before you purchase any hardware or peripherals, we would like to discuss the most important enhancement of all: an upgrade to the latest version of Logo. The latest versions of Logo, such as LogoWriter from LCSI and Logo PLUS from Terrapin, have many features that are particularly well suited to a hypermedia environment. For example, Logo PLUS has the capacity to access LEGO-Logo gears and motors through the Apple II game port.

The Metaphor of a Hypermedia Page

LogoWriter is based on a metaphor that is well suited to a hypermedia environment. LogoWriter uses the metaphor of pages in a book; when you first start LogoWriter a table of contents appears that allows the user to go to any page in the book. Several other hypermedia products use the concept of pages as a metaphor. IBM *Linkway* uses this approach, while one of the hottest new hypermedia products, *ToolBook* for Microsoft *Windows*, is based on this metaphor. In a sense, LogoWriter was ahead of its time.

In a conventional book, the reader usually proceeds in a linear fashion. An audio tape is also a linear environment, since it is necessary to fast forward through all the other material on the tape to get to a song at the end. A compact disc (CD) player is a hypermedia device, since it is possible to directly access a song on any track of the CD. (Some CD players can now be controlled by a computer, and hypermedia programs are beginning to appear for musical works such as Mozart's *Magic Flute*.) In a hypermedia environment, it is possible to go directly from any page to any other page without going though intervening pages. That is why the term *hypermedia* is used to describe this capability.

In LogoWriter each page is named (using the **NamePage** command). The **GetPage** command can be used to go directly from any page to any other page. Quite possibly you have used this capability without thinking of it as a hypermedia command, but this is the fundamental construct that underlies other programs such as *HyperCard*. (In *HyperCard* the metaphor used is that of an index card rather than a page in a book, but the underlying concept is the same.)

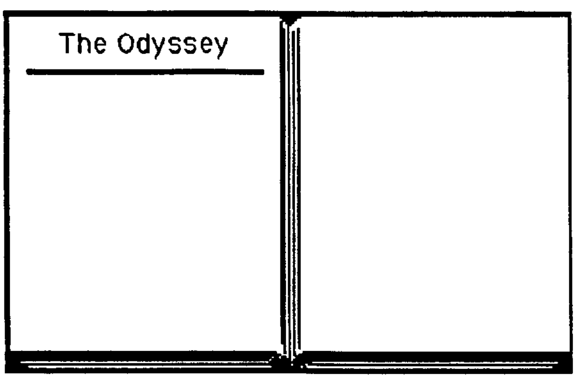
An Epic Adventure

Let's examine the instructional possibilities in this type of hypermedia environment. The following sample excerpt is based on the epic story of Ulysses and the Cyclops, found in Homer's *Odyssey*. We've adopted the model of an adventure game, and the educational purpose—in Logo-like fashion—is motivation rather than direct instruction. Playing the adventure game is intended to introduce students to main characters and events, and encourage them to return to the original *Odyssey* for more detailed information. The gods reward users who make the same choices as Ulysses made! Teachers could use this activity as an introduction to a unit on the *Odyssey*, epic poetry, or heros in mythology.

In the section we have selected, Ulysses and his men have been captured by an evil, one-eyed giant named Polyphemus. Those who have read the book will be familiar with the clues needed to escape this Cyclops. The illustration that we are going to develop is a fragment of what could become a larger LogoWriter program.

Creating the Pages of a LogoWriter Adventure

A drawing of an open book is an effective way of establishing the metaphor of a book. It is not essential to the program, and you may want to proceed to the text elements below, and defer the graphics components until later. We created the drawing with a paint program, and then imported it into LogoWriter using the **Loadpic** command. (In next month's column we will review two of the better paint programs that can be used to do this.) The image could also be created through turtle graphics. The drawing that we imported into LogoWriter looked like this.



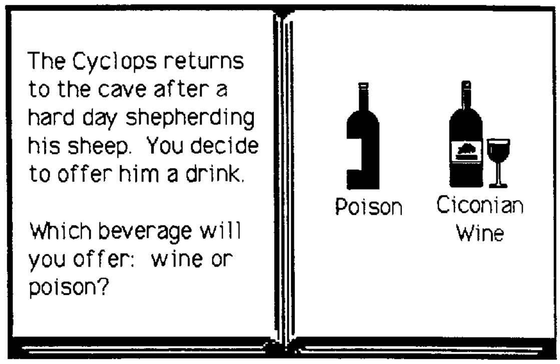
Many paint programs allow different fonts to be created with paint text, and then imported into Logo or LogoWriter, as we have done in this instance.

If you create an illustration of this kind, name the LogoWriter page **Template**, and save the page. While you are on the **Template** page, make a copy to work on by using the **NamePage** command. Name the new page **Dilemma**.

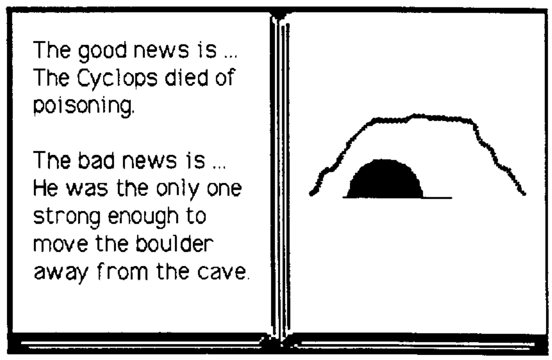
```
NamePage "Dilemma
```

This will create a copy of the **Template** page, leaving the original illustration available for reuse.

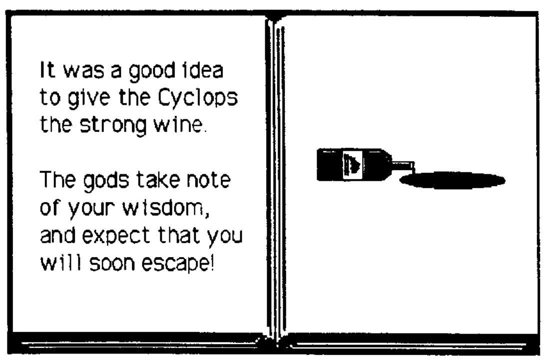
Here is your dilemma, Ulysses:



Two more LogoWriter pages will be needed: one for **Wine** and the other for **Poison**. The LogoWriter page created for the choice of **Poison** looked like this



The LogoWriter page created for the choice of **Wine** looked like this:



Logo Connections - continued

The LogoWriter Procedure

The required LogoWriter procedure is short. This procedure, named **Choice**, asks the user to choose between "Wine" or "Poison." It waits for the user to type a response and goes to the page named "Wine" or "Poison," depending on the answer.

To add this procedure, go to the LogoWriter page **Dilemma**. (You can do this pressing the Escape key to return to the LogoWriter Contents page, or by typing **GetPage "Dilemma** in the LogoWriter Command Center.) Then go to the flip side of the Dilemma page by using the Flip keys. Enter this procedure on the flip side of the page.

```
To Choice
CC
TYPE [Which beverage will you offer?]
MAKE "Input  READLISTCC
IF :Input = [wine] [CC GETPAGE "Wine]
IF :Input = [poison] [CC GETPAGE
   "Poison]
END
```

Most of the commands will be familiar even if you have used a different version of Logo before. The command CC stands for "Clear Command Center." The command **ReadListCC** stands for "Read a List of Words from the Command Center."

After you have entered the procedure, flip back over to the turtle side of the page. Then run the procedure by typing **Choice** in the Command Center, and pressing **Return**.

If you would like to have the question automatically asked when the user enters the **Dilemma** page, add the following procedure on the flip side of the page.

```
To StartUp
Choice
END
```

If you would like to modify the Dilemma page after the automatic startup feature is added, just type "stop" rather than "wine" or "poison" in response to the question.

Contrasting Old and New Versions of Logo

Adventure stories are a time-honored genre in Logo. One of the early issues of the *Logo Exchange* contained a report by Nancy Sopp describing how her learning disabled students used Logo to create adventure stories. Even older versions of Logo can be used to create an adventure story. At the time Nancy Sopp's class created their story, the *Logo Exchange* was printed with a dot matrix printer. However, the features in the newer versions make it much easier to create an adventure, so that more time can be spent on the story and less time on programming details.

Our Logo adventure, called *Ulysses and the Cyclops*, was developed in Apple Logo II several years ago, and later adapted for IBM Logo. We developed a LogoWriter version to provide a contrast between the steps required to create the older version, and the ease of development possible today. In the older version, it was necessary to use a **PRINT** statement to create every line of text.

```
TO DILEMMA
PRINT [The Cyclops returns]
PRINT [to the cave after a]
PRINT [hard day shepherding]
PRINT [his sheep.  You decide]
PRINT [to offer him a drink.]
END
```

Aside from the added work of typing PRINT in front of every line, this approach makes it much more difficult to edit the story. Any revision may require all of the **PRINT** statements to be modified. Every change becomes a major effort which may impede effective modeling and instruction on the writing process. A simple change in metaphor transforms a programming task into a word processing task. A student or teacher can enter the text directly on a LogoWriter page rather than typing **PRINT** statements in front of each line.

Changing the metaphor simplifies the programming task in other ways. In the older versions of Logo used to create the first edition of *Ulysses and the Cyclops*, only one "page" was available, and it was not possible to have more than a few lines of text at the bottom of the graphics screen. Therefore, much of the program consisted of control structures that juggled the transitions from graphics to text, and that spent much of the time rewriting or repainting the screen.

Rationale for Upgrading

Transferring the story to a series of LogoWriter pages dramatically reduces the programming required. A dozen different print, graphics, and control procedures are reduced to a single five-line procedure in this updated revision. Alan Kay, noted for multimedia concepts such as the *dynabook*, has said,

"A point of view is worth 50 IQ points."

The thought and care put into current versions of Logo attest to the truth of this statement.

Surprisingly, many school districts that have little hesitation about upgrading hardware are sometimes reluctant to upgrade software. Recently we were visiting a school system that was using Version 0.9 of an early release of one of the first Logo's from the early 1980's. This particular version could only print in upper case, and was restricted to 40 letters per line. It was not possible to combine graphics and text on a single screen with this version, and bugs in the first release (that had been corrected in the second revision) sometimes caused the program to crash. Since the oldest computer in the lab was an Apple IIe, we asked, "Why are you using this version of Logo?" We were told, "Because that's what you told us to buy!" After recovering from our surprise, we noted that even a program that was the best in its class a decade ago should now be upgraded without any qualms of conscience.

A decade ago the IBM PC did not exist, the Macintosh did not exist, and the goal of many teachers was to upgrade their Apple II+ from 48 kilobytes to 64 kilobytes of memory. Since then, many advances in hardware have taken place. However, in order to take advantage of these advances in hardware, it is necessary to upgrade software as well.

When Logo was first introduced, BASIC programmers often said, "It is possible to do that in BASIC as well." They were correct; it was possible to create many of the effects of Logo in BASIC—but with greater difficulty. Similarly, it is often possible to develop programs in either early or current versions of Logo, but in many cases the development is simpler with advances that take advantage of a decade of experience of teaching with Logo.

Classroom Use

With the metaphor of a book page in LogoWriter, the class can work on much of the design away from the computer. The design can be created on notebook pages, and charted on a bulletin board. (In the film industry, this technique is referred to as a *storyboard*.) As much or more is learned in the creation of a Logo program as in the use of it. Perhaps it will be necessary for members of the class to go to the library to consult references and resource documents. Are there other references to the Cyclops from ancient times? Who was Homer? What is the oldest copy of the *Odyssey*? Who translated it, and where is it found? The thesaurus, the dictionary, the encyclopedia, and reference works will be needed to answer these questions. Collaboration and cooperation among members of the class will be required, since a project of this kind is too large for any one person.

Pages pinned to the bulletin board can be translated directly into LogoWriter pages. Because of this one-to-one correspondence, the design process becomes much more easily comprehended, even by younger children. There are many other events from the Odyssey that could be incorporated into a Logo adventure story. What other decisions did Ulysses make? Here are some choices a class might want to add to the adventure:

- Which type of wood should be used to create a spear to blind the Cyclops? (Those concerned about violence in the cinema might be startled at the levels of violence in the classics.)

- What will you say when the Cyclops asks your name? (Ulysses said his name was "No Man", a ploy that figured prominently in his later escape.)

- How will you disguise yourself to get past the blinded and enraged Cyclops at the mouth of the cave?

With minimal changes, the adventure story model could be adapted for other stories. The underlying concept would work as well for *Huckleberry Finn* as for the *Odyssey*. (What should Tom Sawyer tell Aunt Polly? How can Huck and Jim escape the Duke?) The adventure can be designed to closely parallel literature the class is studying, or an entirely original adventure can be created.

Summary

The structure of a computing environment has a great effect on what may be reasonably attempted in a classroom setting. The first versions of Logo developed for early microcomputers had to be designed to fit into comparatively small amounts of memory. Current versions of Logo take advantage of expanded memory and faster processor speeds. Equally important, they reflect the experience of a decade of classroom use.

The latest versions of Logo are particularly well suited to development of hypermedia applications. In future columns we are going to discuss peripherals and hardware that lend themselves to use with Logo. Before considering these options, however, the most important step that you can take is to upgrade the version of Logo you are using to its most current version. The cost of upgrading is relatively small compared with other enhancements you might acquire, and the potential benefits are great.

Glen and Gina Bull, Curry School of Education, Ruffner Hall, University of Virginia, Charlottesville, VA 22903
BITNET addresses: Glen: GBULL@ VIRGINIA.
Gina: GINA@VIRGINIA.

Extra for Experts

Fractals II: Representations

by Mark Horney

Fractals are a hot topic today. They form a basis for the new branch of mathematics called Chaos Theory. Yet, some of the basic concepts of fractals go back to the turn of the century when mathematicians such as Peano, Cantor, and von Koch created curves that they described as monstrous and pathological. Today, using Logo, these shapes can be easily drawn and provide an excellent introduction to recursion and other topics in programming and mathematics.

A fractal can be drawn using a simple procedure such as the one shown below for the "C" curve defined by Donald Knuth (see the level 10 C curve in figure 1). The C curve is based on repeated uses of the "Hat Function," which is a 90-degree angle drawn to the left of the turtle's path. (A level 1 C curve will show this shape, which is called the fractal's generator. See last month's column for more details about how to draw fractals. This procedure is written in Macintosh Logo 1.0):

```
TO  C  :SIDE  :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
LEFT 45
C  :SIDE / SQRT 2  :LEVEL - 1
RIGHT 90
C  :SIDE / SQRT 2  :LEVEL - 1
LEFT 45
END
```

This procedure is for a left C curve. A right C curve, with the hat function drawn to the right of the turtle's path, can be drawn by switching the lefts and rights in the procedure. This can be done with any fractal procedure and soon leads to the idea of mixing left and right fractals together. This uncovers a critical insight: *There is no reason why recursive fractal procedures must always call themselves. Different fractals can call one another.* For example, the left and right C curve procedures shown below have been modified so that each calls the other. Together, these two procedures draw the fractal known as the Dragon:

```
TO  DL  :SIDE  :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
LEFT 45
DL  :SIDE / SQRT 2  :LEVEL - 1
RIGHT 90
DR  :SIDE / SQRT 2  :LEVEL - 1
LEFT 45
END
```

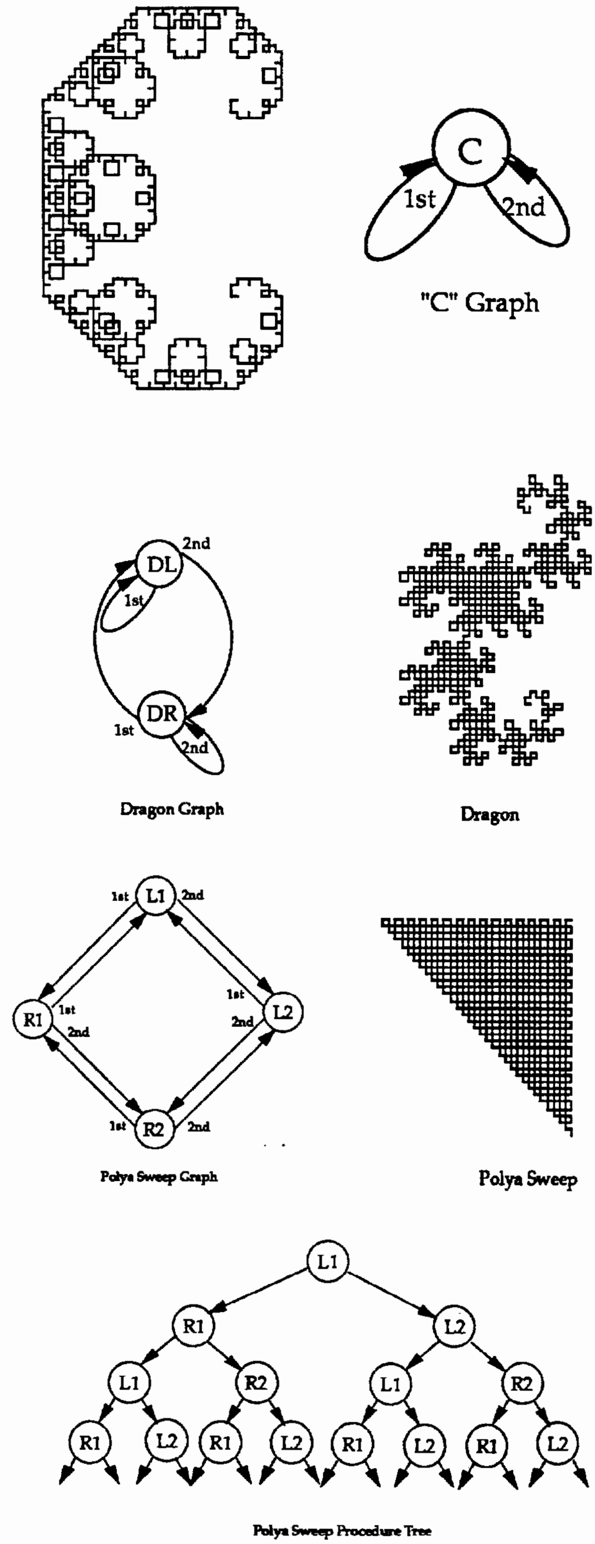
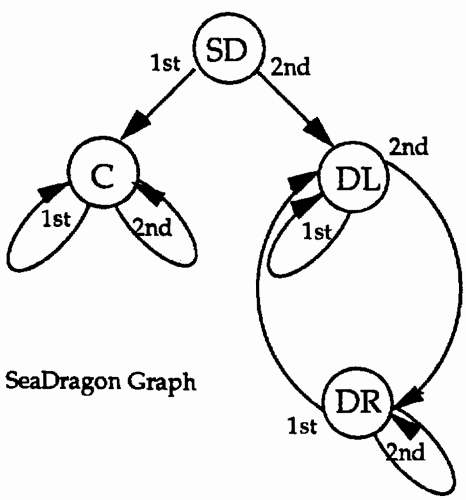


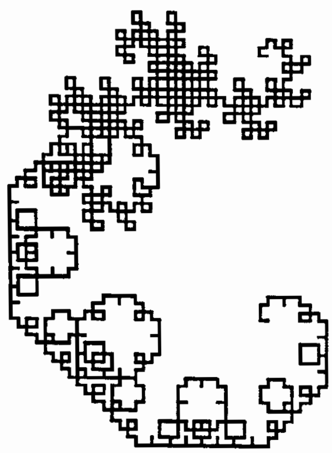
Figure 1

```
TO  DR  :SIDE  :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
RIGHT 45
DL  :SIDE / SQRT 2  :LEVEL - 1
LEFT 90
DR  :SIDE / SQRT 2  :LEVEL - 1
RIGHT 45
END
```

Another multi-generator fractal is the Polya Triangle Sweep shown last month. Benoit Mandelbrot, in his book, *The Fractal Geometry of Nature*, defines describes this curve as: "The initiator is [0,1], the generator is [the hat function], and



SeaDragon Graph



Sea Dragon

Figure 2

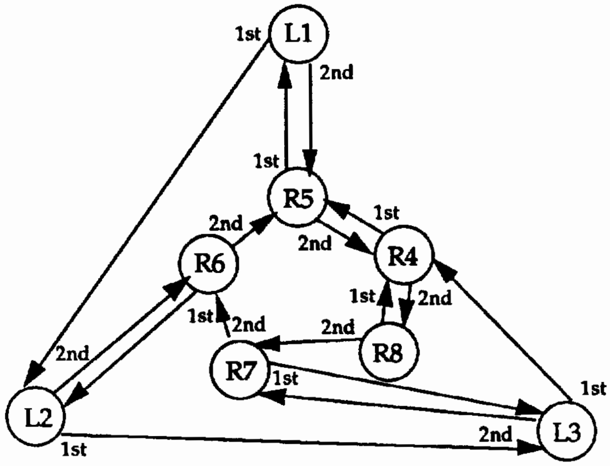
it alternates between the right and the left of the teragon [Mandelbrot's word for these fractals]. The first position also alternates." (p. 64, See figure 1). Thus, a level 1 Polya Sweep is the left hat. A level 2 sweep is a right hat, followed by a left (each reduced in size of course). Level 3 is a left, a right, a left and a right. Higher levels continue these alternating patterns. It's easy to see that at least two procedures are needed for the sweep, something like a left C curve and a right C curve. It turns out that four procedures are needed, two lefts and two rights. Discovering this, and how the four procedures interrelate is a natural application of the procedure trees all Logo students are taught to make. The first few levels of the Polya Sweep procedure tree is shown in figure 1. Note how **Left1** calls a right and a left, in that order, but **Left2** calls a left and a right, and similarly with **Right1** and **Right2.**

Once students understand the use of procedure trees to unravel fractal interconnections, they can create a host of ever more complicated curves. Soon, they will find that although drawing procedure trees is necessary to decode fractals drawn by someone else, it's a cumbersome method of creating new fractals. A more compact fractal representation is needed. One way is to change the infinite procedure trees into finite directed graphs.

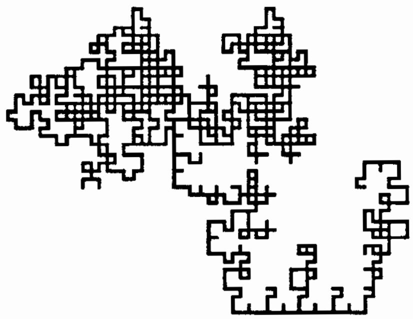
Graphs are a concept from the branch of mathematics called Combinatorics. These graphs are not line nor bar graphs. Rather, they are ball and stick diagrams where "vertices" are used to stand for objects that are "linked" together. Here, vertices represent procedures and links show how procedures call one another (this is just as in procedure trees but with each procedure only represented once). These are *directed* graphs because the links are one way connections (i.e., this procedure calls that one, but not vise versa.) Figure 1 shows a directed graph for the Polya Sweep.

Once students have these two key ideas—that fractals can be drawn with several generators, and that fractals can be represented by graphs—hours, days and weeks can be spent creating wonderfully complex shapes. This enthusiasm can be harnessed by teachers to guide students into explorations of topics in programming, graph theory, geometry, and combinatorics. Here are a few ideas for exploration:

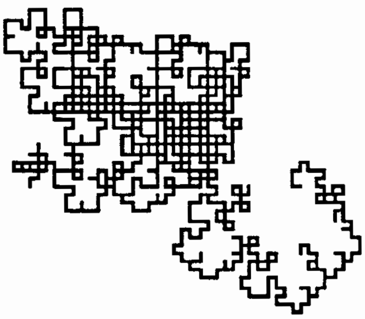
1. The Polya Sweep program is run by calling the **Left1** procedure. This is natural because the curve is defined as starting to the left, but there really is nothing special about **Left1.** What happens if you start with **Left2,** or **Right2**? The graph of a fractal actually represents a whole family of curves, each obtained by starting the turtle from a different vertex.



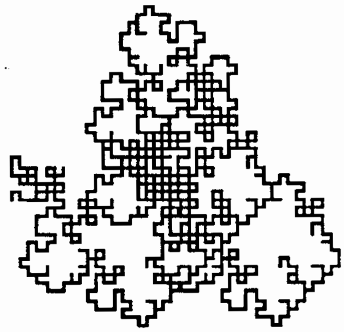
Massachusetts Graph



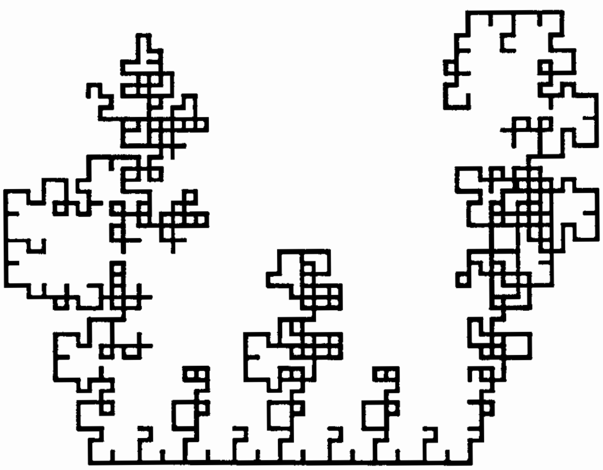
Mass L1



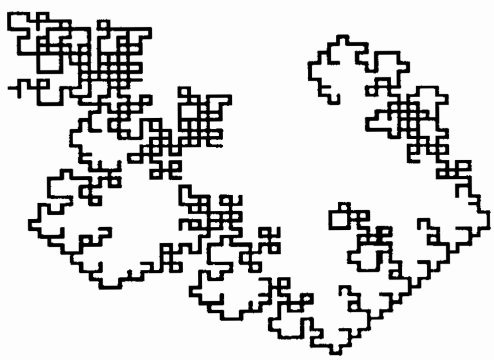
Mass L2



Mass L3

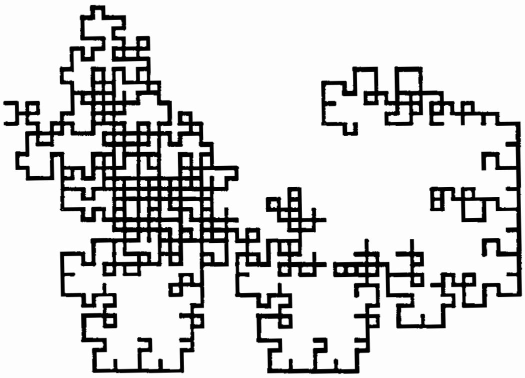


Mass R4

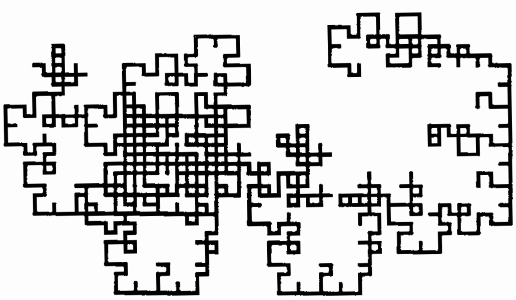


Mass R5

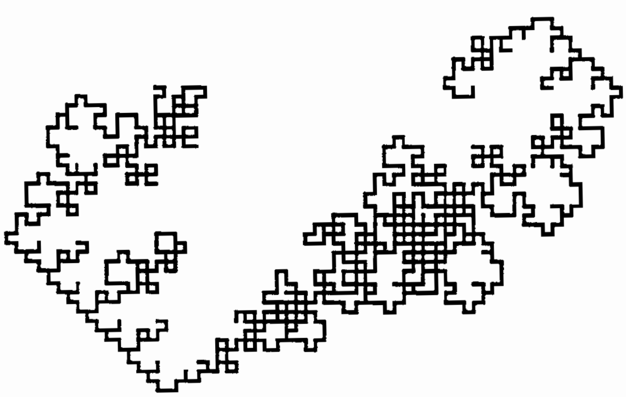
Figure 2 (continued)



Mass R6



Mass R7



Mass R8

Figure 2 (continued)

Figure 2 shows a family of curves called Massachusetts. Notice how the overall shape of family members are repeated as sub-shapes in others.

2. The fractals shown here were drawn with just two basic generators, the left and right hat functions. How many different fractals can be drawn with just these two shapes? Obviously an infinite number can be drawn because an infinite number of procedures can be used. This question is better posed by asking how many fractals can be drawn with just 1 procedure (two: the left and right C curves), or just 2 procedures (32) or 3 procedures? How do the group of fractals drawn with a given number of procedures relate to one another? Are there different graphs that represent the same fractal? Are all the fractals unique?

3. How are properties of the graph representing a fractal visible in the shape of the fractal itself? For example, the graph for the Polya Sweep is "strongly connected." That means it is possible to travel from any vertex to any other. Not all directed graphs have this property, for instance the Sea Dragon shown in figure 2. This fractal is a combination of the C curve and a Dragon fractal. The Sea Dragon graph shows the two vertices needed to drawn the dragon curve (on the right) and the one vertex for the C curve (to the left). Between them is a "Bridge" procedure connecting the two. Note how the C curves only calls itself, as does the Dragon curve. What happens when the turtle starts at one of the Dragon vertices?

This article and the one from last month on Visible Recursion have shown how fractals can be used to learn about concepts in programming and mathematics, but very little has been said about fractals themselves. Next month "Extra For Experts" will look at just what makes a fractal a fractal.

Mark Horney was a middle school math, science and reading teacher for 10 years and later spent 3 years as a high school computer science instructor. He is currently a doctoral student at the University of Oregon.

Mark Horney
1787 Agate Street
Eugene, Oregon 97403
BITNET: MHORNEY@OREGON

MathWorlds

edited by
A. J. (Sandy) Dawson

The new found mathematical world of fractals is attracting untold attention by mathematicians, mathematics teachers, and mathematics students alike. Some months back, Harold Broachman, a teacher in North Vancouver, Canada, wrote here about his work on fractals. I know that Harold expanded his interest and knowledge of fractals this past May when he attended a special, three-day study session devoted to fractals sponsored by the Canadian Mathematics Education Study Group (CMESG) and held at Simon Fraser University in Vancouver.

And now from the other side of the world we find Hannu Korhonen, a mathematics researcher and educator equally taken by the mysteries and magic of fractals. As with so much of what is delightful in mathematics, the chaotic attractors that Hannu presents can be described simply, yet the patterns created are rich and complex. I am delighted to be able to devote a second column to the wonderfully fascinating world of fractals.

Chaotic Attractors in Logo by Hannu Korhonen

In school mathematics we are used to dealing with uncomplicated functions. However, using just elementary functions it is also possible to get quite intricate graphs. One characteristic of some such functions is that it is not possible to directly get a single point for such a graph, because all previous points must be calculated one by one in sequence. This kind of a definition is called recursive. The next point on the graph depends on the coordinates of the previous point. The graph is not necessarily continuous and the points may seem to appear in an arbitrary way. Nevertheless, the points can constitute a regular pattern. When this happens, we say that the graph is an attractor. There are different kinds of such graphs: connected plane figures, fractals, and sets of isolated points. This article is about attractors, and in particular, chaotic attractors.

Gingerbreadman

The rule for a Gingerbreadman in words is as follows:

1. The next value for x can be obtained by subtracting the previous y-value from the number 1 and then adding the absolute value of the previous x-value,

2. The next y-value is equal to the previous x-value (e.g., the image of the point (2, 5) is (-2, 2), because

$x = 1 - 5 + |2| = -2$
$y = 2$).

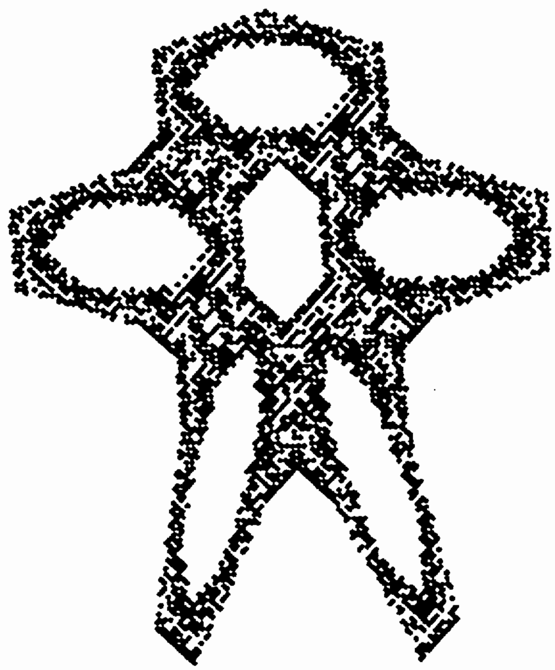
The following points are (1, -2) and (4, 1) when calculated in the same way. The rules are applied to the last pair of numbers every time, and the computing is continued without limit.

The function for the Gingerbreadman is defined by means of the following equations:

$$xn_{+1} = 1 - y_n + |x_n|$$
$$yn_{+1} = x_n.$$

Here (x_n, y_n) signifies one point and (x_{n+1}, y_{n+1}) the next point.

There is nothing strange in these expressions. One might not expect any interesting results. But the truth is really amazing. When you start with the point (-0.09, -0.6) you will get the following regular, easily recognizable shape.



The Logo program is as follows:

```
To Gingerbreadman :x :y
dot list 14 * :x 14 * :y - 40
Gingerbreadman 1 - :y + abs:x :x
End
```

```
To Dot :List
seth 0
pu
setx first :list
sety :last :list
pd
forward .5
pu
back .5
End

To Abs :Num
output ifelse :num < 0 [minus :num ]
   [ :num]
End
```

The dot line is really useful because with its help you can move, enlarge, shrink or rotate the figure without altering the computation of the x- and y-values. The code is in LogoWriter, where dot and abs are not primitives.

Five thousand points of the Gingerbreadman.

The body of the Gingerbreadman depends critically on the starting values. When the first value of y is decreased, the Gingerbreadman gets smoother corners at first and then fragments into separate hexagons and finally into isolated points. Because the points of the graph are hopping here and there randomly, we say that the process is chaotic. Because the points appear in the definite area, the process has an attractor—a chaotic attractor.

The Gingerbreadman is only a mathematical oddity, but its message is profound. The underlying rules of a seemingly complex phenomenon may be quite simple. The simplicity does not mean that the phenomenon would be completely regulated. When looking at the expressions of the mathematical definition of the Gingerbreadman, it is not easy to foresee the result, although all details in the expressions are known to and can be understood by every school child.

Fern

The next body—a Fern—is a well-known symbol for fractal designs, but you don't see the algorithm behind it very often in Logo. There is a general algorithm for generating fractals called the Random Iteration Algorithm. It uses six times four parameters as starting values. The result critically depends on these initial values.

The above fern has over one million points. The values used are given below the picture.

```
a.        [0   0.85   0.2   -0.15]
b.        [0   0.04   -0.26   0.28]
c.        [0   -0.04   0.23   0.26]
d.        [0.16   0.85   0.22   0.24]
e.        [0   0   0   0]
f.        [0   0.2   0.2   0.2].
```

If you change the parameters to

```
a.        [0.5   0.5   0.5]
b.        [0   0   0]
c.        [0   0   0]
d.        [0.5   0.5   0.5]
e.        [1   1   50]
f.        [1   50   50].
```

you'll get a sophisticated triangle. Using other values you can get squares, trees, etc. It is absolutely marvellous that the same algorithm gives so many completely different forms.

The Logo program for the random iteration algorithm is as follows:

```
To Ria :x :x :k
dot list 240 * :x 160 * :y - 90
Ria :x * (item :k :a) + :y * (item :k
   :b) + item :k :e :x * (item :k :c)
   + :y * (item :k :d) + item :k :f
   1 + random 4
End
```

and the program call (e.g., Ria 0 0 1). The Fern is clearly a fractal, an extremely broken pattern that consists of details similar to the whole body.

Cockatoo

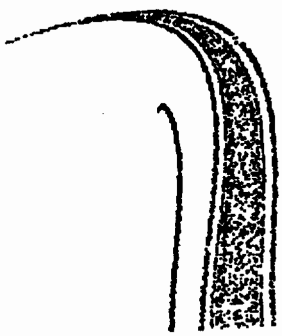
The Cockatoo function is another peculiarity. There is no advanced mathematics either in this case: only the basic operations of absolute value and square root. The *LogoWriter* program is as follows:

```
To Cockatoo :x :y
dot list 180 * :x - 60 120 * :y -30
Cockatoo 0.21 - :y :x * (1 + sin 180/
   pi * 0.7 * :y) - 1.2 * sqrt abs :y
End
```

MathWorlds - continued

Here is the crest of the Cockatoo containing 10,000 points.

Now you can choose any starting point near the origin. It is not easy to see that the cockatoo is fractalized, too. But if you enlarge a detail of the sketch 10 times you can see that the feathers have a fine intrinsic structure.



In the case of enlarging a detail of the Cockatoo, all the previous points must be calculated, although only a restricted part of them is shown on the screen.

References:

Barnsley, M. (1988). *Fractals everywhere*, 87,91-92. New York: Academic Press.

Martin, B. (1987). The cockatoo. *The mathematical intelligencer*, 9 (1), 72.

Peitgen, H. O. & Saupe, D. (eds.)(1988). *The science of fractals images*,149-150. New York: Springer Verlag.

Hannu Korhonen is a mathematics education researcher and teacher at the Institute for Educational Research, University of Jyväskylä, Jyväskylä, Finland. He can be reached through Bitnet as KTLTAO@FINJYU.BITNET.

A. J. (Sandy) Dawson is a member of the Faculty of Education at Simon Fraser University in Vancouver, Canada. He can be reached through Bitnet as USERDAWS@SFU.BITNET

Logo: Search and Research

Turtle Soup: A Beginning Look at Logo Research

by Douglas H. Clements

What does Logo research say to the beginner? What effects can be expected? How can benefits best be garnered? This month I provide an overview of hundreds of studies reviewed previously in this column. This summary reveals a surprising source of educational nutrition in the many ingredients of "turtle soup."

Mathematics

Logo was first developed as a conceptual framework for learning mathematics (Feurzeig & Lukas, 1971; Papert, 1980). Many researchers took this to mean that merely programming in Logo would increase students' mathematics achievement. They were half right.

Students do use mathematical concepts in Logo. Even first graders apply ideas of number, arithmetic, estimation, measure, patterning, proportion, and symmetry (Kull, 1986). Similar observations show that Logo allows intermediate grade students to explore certain mathematical concepts earlier than is currently believed (Carmichael, Burnett, Higginson, Moore, & Pollard, 1985; Papert, Watt, diSessa, & Weir, 1979). Such observations lend credence to this "exposure approach" to teaching with Logo. On the other hand, it is uncertain whether just any exposure leads to increased test scores. Some studies say "yes," others "no" (Clements, in press).

However, using Logo as a conceptual framework is *not* the same as "providing practice with," or directly teaching, mathematical ideas. Rather, Logo is a tool for building new mental structures—frameworks for all future learning and problem solving. As such, Logo allows students to manipulate representations of mathematical ideas. It serves as a transitional device between concrete experiences and abstract mathematics.

Geometry provides a good example. Feeling the side of a table or walking a straight path provides concrete experiences with the concept of straightness. Logo helps by bringing the concept to a more explicit level of awareness. It is easy to have students use the turtle to discover that a straight path is one that has no turning. As another example, students often think of a shape such as a parallelogram as a visual "whole." Drawing a parallelogram with Logo "...allows, or obliges, [children] to externalize intuitive expectations. When the intuition is translated into a program it becomes more obtrusive and more accessible to reflection" (Papert, 1980, p. 145). Students must analyze the visual aspects of the parallelogram to construct a PARALLELOGRAM procedure. What are its parts? How are they put together? In other words, what are the *properties* of a parallelogram? Writing a PARALLELOGRAM :SIDE.LENGTH procedure is tantamount to writing one type of formal definition. In doing so, students build intuitive knowledge about the concept of defining geometric figures. Finally, asked whether they can use that procedure to draw a rectangle, rhombus, or square, students have to confront the *relationships* among quadrilaterals.

Research on this "conceptual framework approach" is much more positive. Appropriate use of Logo helps students analyze and understand the properties of geometric figures (Clements & Battista, 1989; Clements & Battista, in press-a). Still, misconceptions persist for many students, and they do not always transfer what they learn in Logo to other situations (Clements, in press). The problem is that students do not always think mathematically, even if use of Logo encourages such thinking. They often rely too heavily on visual clues ("It looks like about 75"). Although important in beginning phases of learning, overuse of such nonanalytic strategies prevents students from making mathematical generalizations related to their Logo activity.

What is special about the research programs in which students consistently *do* analyze and *do* learn? The teachers employ carefully planned sequences of Logo problems. They help students abandon visual strategies by presenting problems that require analytic solutions. They talk to students about their work and encourage them to talk to each other. In short, these teachers help *shape* their students' learning—by mediating the students' experiences with Logo. As a result, the students reflect on their work and forge links between what they know in and out of the Logo context (Clements & Battista, 1989).

So, there are two main recommendations for instruction. First, recognize that neither the "exposure approach" nor the "conceptual framework approach" is satisfactory. Instead, use the "mediated conceptual framework approach" (Clements, in press). What does such mediation involve?

- Building awareness of the mathematics in Logo work.
- Elaborating the ideas encountered in Logo.
- Providing some structure for Logo problems and explorations.
- Constructing links between Logo and non-Logo work.

These links can be made in two different ways. One uses Logo to deliver the traditional curriculum. The other expands traditional activities so that students use higher-level thinking processes. The latter is more consistent with the National Council of Teachers of Mathematics' recommendations (1988).

The second recommendation for instruction is to ensure that students work at the right level of representation. Programming problems should not overwhelm mathematical problems. In geometry, for example, instead of having elementary students write Logo procedures to perform geometric motions, have them solve significant motions problems using Logo tools (Clements & Battista, in press-b; du Boulay, 1986).

Problem Solving

Logo was also developed to be a tool for thinking. Again, different approaches to developing problem-solving abilities yield different results.

Researchers taking the "exposure approach" claim that programming and problem solving use equivalent mental processes, so exposure to one should develop the other. Research results are mixed. After Logo programming, students improve on some tasks such as permutation and classification problems, but not on others, such as the Tower of Hanoi problem (Clements, in press; Statz, 1974).

Other researchers use variations of the "conceptual framework approach." For example, some suggested that programming involves extensive planning. In one study, however, middle and high school students exposed to Logo did not display greater planning skills than students in a matched group (Kurland, Pea, Clement, & Mawby, 1986).

With proper mediation, however, Logo can increase problem-solving ability. Why is mediation necessary? Consider planning again. Observations of students working on Logo tasks show considerable growth in planning (Kull, 1986; Nastasi, Clements, & Battista, 1990; Noss, 1984). This growth is slow, however, and without teacher mediation to highlight planning processes, use in noncomputer situations is unlikely. Students must become aware of their planning and lift it from the Logo context. For example, Bamberger (1985) stressed planning procedures before putting them in computer code and using such strategies as breaking a large idea into parts. Her students planned more on later noncomputer problems.

Search and Research - continued

Effects on processes other than planning may be more profound. One of the strongest benefits of Logo is increasing students' tendency and ability to "debug"—find their mistakes in—all types of problem situations (Clements, 1986; Clements, 1990). These and other studies with positive results employed much teacher mediation (Lehrer, Harckham, Archer, & Pruzek, 1986; Littlefield et al., 1988). Further, this mediation had a conceptual framework—at its foundation was a theory of human problem solving. What might the teacher do to mediate students' learning of problem solving in Logo?

- Be aware of students' problem-solving strategies and processes, including current theories of problem solving.
- Ask high-level questions so that students, too, become aware of the problem-solving strategies and processes they are using.
- Discuss how the strategies used in Logo might be applied in other situations.
- Discuss different ways of thinking about problems, including errors people often make.
- Have students work in small groups.

Social and Emotional Development

A surprisingly significant benefit of the use of Logo involves students working together. Research shows that Logo facilitates social interaction and focuses that interaction on learning. Logo students work cooperatively more than students working on Computer Assisted Instruction (CAI) or noncomputer tasks. They also disagree more—but they disagree about *ideas*, and they are more likely to resolve these disagreements successfully (Clements & Nastasi, in press; Nastasi et al., 1990).

Students working with Logo learn to listen, criticize in a helpful fashion, and appreciate the work of others (Carmichael et al., 1985). They help and teach each other (Clements & Nastasi, 1985; Fire Dog, 1985).

What of students' affective or emotional side? Students working with Logo experience an increase in self-esteem and confidence if their teacher gives them greater autonomy over their learning and fosters social interaction (Carmichael et al., 1985; Fire Dog, 1985; Kull, 1986). Logo especially provides special needs students with prestige and respect from their peers, enhancing their self concepts (Michayluk, Saklofske, & Yackulic, 1984).

All types of students learn to judge situations for themselves and accept responsibility for their actions (Blumenthal, 1986; Horner & Maddux, 1985). They improve their attitudes toward school. In sum, student-student and student-teacher interactions may be as important to social, emotional, and cognitive development as are the student-Logo interactions.

Conclusions: The Nutrients of Turtle Soup

Several ingredients are critical to any good recipe. Logo is but one ingredient. Full realization of Logo's potential requires a mediated conceptual framework approach—a qualitatively different way of teaching and learning. (In other words, Logo programming must be "souped up" through mediation by the teacher.)

Finally, mediated Logo environments provide subsistence for many different aspects of students' lives. It is likely that using CAI or noncomputer materials might lead to similar gains on a single test, but they probably would *not* show benefits of such a wide scope. From the mathematical and cognitive to the social and emotional, Logo encourages the synthesis of the abstract and logical with the concrete and aesthetic, the analytic with the intuitive, and social knowledge with personal discovery. Recall the story "Stone Soup," in which a nutritious concoction resulted from the combination of many small offerings. Similarly, the main nourishment of turtle soup may emerge from many moderate but interrelated contributions. But, as in the story, the local culture has to change—become more willing to experiment and share—for such contributions to emerge.

References

Bamberger, H. J. (1985). The effect of Logo (turtle graphics) on the problem solving strategies used by fourth grade children. *Dissertation Abstracts International, 46*, 918A. (University Microfilms No. DA8512171)

Blumenthal, W. (1986). *The effects of computer instruction on low achieving children's academic self-beliefs and performance.* Unpublished doctoral dissertation, Nova University.

Carmichael, H. W., Burnett, J. D., Higginson, W. C., Moore, B. G., & Pollard, P. J. (1985). *Computers, children and classrooms: A multisite evaluation of the creative use of microcomputers by elementary school children.* Toronto, Ontario, Canada: Ministry of Education.

Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology, 78*, 309-318.

Clements, D. H. (1990). Metacomponential development in a Logo programming environment. *Journal of Educational Psychology, 82*, 141-149.

Clements, D. H. (in press). Research on Logo: Effects and efficacy. *Education.*

Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education, 20*, 450-467.

Clements, D. H., & Battista, M. T. (in press-a). Geometry and spatial reasoning. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching*. Reston, VA: National Council of Teachers of Mathematics/Macmillan.

Clements, D. H., & Battista, M. T. (in press-b). *Logo-based geometry curriculum: K-6 [Computer software]*. Morristown, NJ: Silver Burdett & Ginn.

Clements, D. H., & Nastasi, B. K. (1985). Effects of computer environments on social-emotional development: Logo and computer-assisted instruction. *Computers in the Schools, 2*(2-3), 11-31.

Clements, D. H., & Nastasi, B. K. (in press). Computers and early childhood education. In T. Kratochwill, S. Elliott, & M. Gettinger (Ed.), *Advances in school psychology: Preschool and early childhood treatment directions.* Hillsdale, NJ: Lawrence Erlbaum.

du Boulay, B. (1986). Part II: Logo confessions. In R. Lawler, B. d. Boulay, M. Hughes, & H. Macleod (Ed.), *Cognition and computers: Studies in learning* (pp. 81-178). Chichester, England: Ellis Horwood Limited.

Feurzeig, W., & Lukas, G. (1971). LOGO—A programming language for teaching mathematics. *Educational Technology, 12*, 39-46.

Fire Dog, P. (1985). Exciting effects of Logo in an urban public school system. *Educational Leadership, 43*, 45-47.

Horner, C. M., & Maddux, C. D. (1985). The effect of Logo on attributions toward success. *Computers in the Schools, 2*(2-3), 45-54.

Kull, J. A. (1986). Learning and Logo. In P. F. Campbell & G. G. Fein (Ed.), *Young children and microcomputers* (pp. 103-130). Englewood Cliffs, NJ: Prentice-Hall.

Kurland, D. M., Pea, R. D., Clement, C., & Mawby, R. (1986). A study of the development of programming ability and thinking skills in high school students. *Journal of Educational Computing Research, 2*, 429-458.

Lehrer, R., Harckham, L. D., Archer, P., & Pruzek, R. M. (1986). Microcomputer–based instruction in special education. *Journal of Educational Computing Research, 2*, 337–355.

Littlefield, J., Delclos, V. R., Lever, S., Clayton, K. N., Bradsford, J. D., & Franks, J. J. (1988). Learning Logo: Method of teaching, transfer of general skills, and attitudes toward school and computers. In R. Mayer (Ed.), *Teaching and learning computer programming: Multiple research perspectives* (pp. 111-135). Hillsdale, NJ: Erlbaum.

Michayluk, J. O., Saklofske, D. H., & Yackulic, R. A. (1984, June). *Logo*. Paper presented at the meeting of the CAP Convention, Ottawa, Ontario.

Nastasi, B. K., Clements, D. H., & Battista, M. T. (1990). Social-cognitive interactions, motivation, and cognitive growth in Logo programming and CAI problem-solving environments. *Journal of Educational Psychology, 82*, 150-158.

National Council of Teachers of Mathematics. (1988). *Curriculum and evaluation standards for school mathematics*. Reston, VA: Author.

Noss, R. (1984). *Children learning Logo programming. Interim report No. 2 of the Chiltern Logo Project*. Hatfield, England: Advisory Unit for Computer Based Education.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Papert, S., Watt, D., diSessa, A., & Weir, S. (1979). *Final report of the Brookline Logo Project. Part II: Project summary and data analysis (Logo Memo No. 53).* Cambridge, MA: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Statz, J. (1974). The development of computer programming concepts and problem solving abilities among ten-year-olds learning Logo. *Dissertation Abstracts International, 34*, 5418B-5419B. (University Microfilms No. 74-10,180)

Douglas H. Clements
State University of New York at Buffalo
Department of Learning and Instruction
593 Baldy Hall
Buffalo, NY 14260
CIS: 76136,2027 BITNET: INSDHC@UBVMS

Global Logo Comments

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

Logo at the WCCE

Every five years the International Federation of Information Processing (IFIP) hosts the World Conference on Computers in Education (WCCE). The Fifth WCCE was held this past July in Sydney, Australia, while the previous conferences were held in Amsterdam, Paris, Lausanne, and Norfolk. Having arranged the Logo activities for the 1985 WCCE in Norfolk, I was anxious to see how many Logo papers and meetings would be given in Sydney. To my surprise, Logo was very well represented in the paper, seminar, and panel sessions. In fact, the Logo sessions were consistently the most well attended in the conference. One had to arrive 10 minutes prior to the session to get a seat.

During one crowded session, Seymour Papert made some statements, filling in for one of the panelists. An Australian teacher asked SIGLogo president Gary Stager, "Who was that guy?" When Gary told her it was Seymour Papert she replied, "How do you spell Papert?" Professor Papert gave the final keynote address entitled, "The Perestroika of Epistemological Politics." As usual, he had many thought-provoking things to say to the audience of about 2,500 computer educators.

A packed house of nearly 100 people attended the ISTE SIGLogo meeting. This was considerably more than attended the SIGLogo meeting a couple weeks prior at the NECC in Nashville. As both conferences had similar attendance, the SIGLogo meetings seem to substantiate that Logo is more healthy in Australia than in the United States. In fact, there were many first-rate papers on Logo given at the conference. Of course, I solicited many Logo users throughout the world to submit results of their work to our continental editors for future inclusion in this column.

A sampling of papers appearing in the WCCE proceedings follow:

Problem Solving in Lego/Logo Environment: Cognitive and Metacognitive Outcomes by K.W. Lai, New Zealand

Children, Recursion, and Logo Programming: An Investigation of Papert's Conjecture about the Variability of Piagetian States in Computer-Rich Cultures by A. McDougall, Australia

Step by Step: Lego Legged Locomotion by P. J. Carter, Australia

Logo Microworlds by Bruce W. McMillan, New Zealand

The Warp and the Weave: Logo and Textiles—or is it Cross Curriculum? by Leonie Fraser, Australia

Logo: Using Thousands of Turtles to Explore Self-Organizing Behavior, by Mitchel Resnick, USA

The Treasure Game: A Logo-Based Environment for the Study and Exploration of Feedback by Uri Wilensky and Aaron Brandes, USA

Logomate: A facilitator of Logo Learning for Preliminary and Early Secondary Students, by Robert Pitts, Australia

Pre-Logo Learning Environments for Turkish Children by Bahar Alakent and Emra Alfran, Turkey

Lego Logo for Big Kids: Control Technology and Senior Secondary Curricular by Arthur Tatnall, Australia

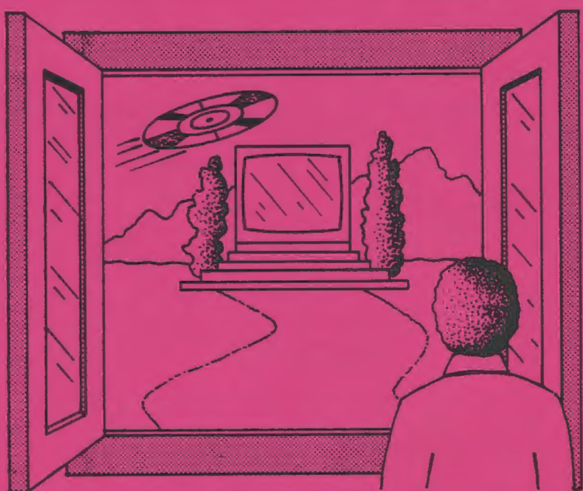
Logo in Secondary Education by Tom Lough, USA

Hardware and Software Interfaces to Lego for LogoWriter by Kin P. Leung, Australia

Preservice Teacher Education: Logo as a Potentially Rich Computing Situation by Anthony Jones, Australia

So, Logo is alive and well in many countries of the world. The Netherlands, Iceland, Australia, and Bulgaria continue to be hotbeds of Logo activity. While others seem to be catching up, the United States and United Kingdom speakers seemed to be emphasizing teleteaching over Logo. The conference was certainly a huge success. Many of the participants are already looking forward to the next WCCE in Birmingham, England in 1995.

A Window on the Past.



A Vision of the Future.

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

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

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

ISTE/University of Oregon
1787 Agate Street, Eugene OR 97403-9905
(503) 346-4414

Telecommunications: Make the connection.

Whether you want to hook up with a teacher in Kenya, or a teacher across town, ISTE's *Telecommunications in the Classroom* will help you make the connection.

Authors Chris Clark, Barbara Kurshan, Sharon Yoder, and teachers around the world have done your homework in *Telecommunications in the Classroom.* The book details what telecommunications is, how to apply it in your classroom, what hardware and software you'll need, and what services are available. *Telecommunications in the Classroom* also includes a glossary of telecommunications terms and exemplary lesson plans from K-12 teachers.

Telecommunications in the Classroom is an affordable, informative resource for workshops, classes, and personal use. **$10** +$2.85 shipping

Make your connection today with ISTE's *Telecommunications in the Classroom*

ISTE, University of Oregon, 1787 Agate St.,
Eugene, OR 97403-9905; ph. 503/346-4414

Logo Exchange
ISTE/University of Oregon
1787 Agate Street
Eugene, OR 97403-9905