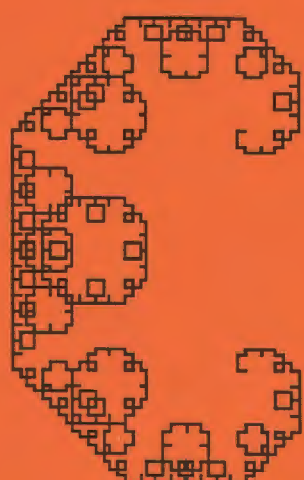
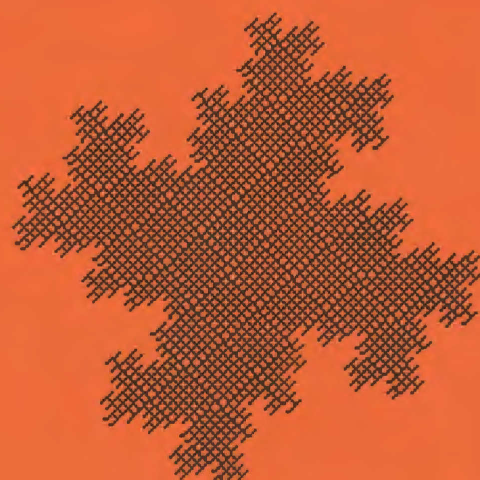
Journal of the ISTE Special Interest Group for Logo-Using Educators

LOGO EXCHANGE

September 1990 Volume 9 Number 1

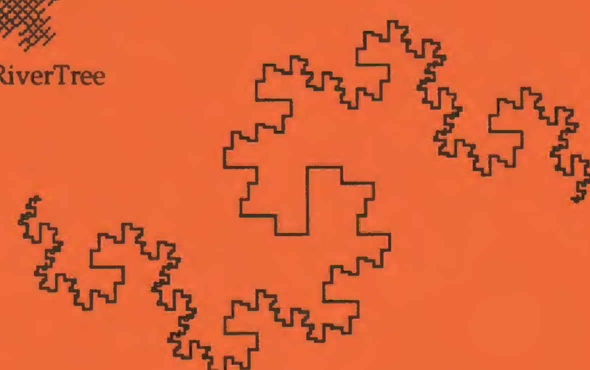


"C"



RiverTree

Spike



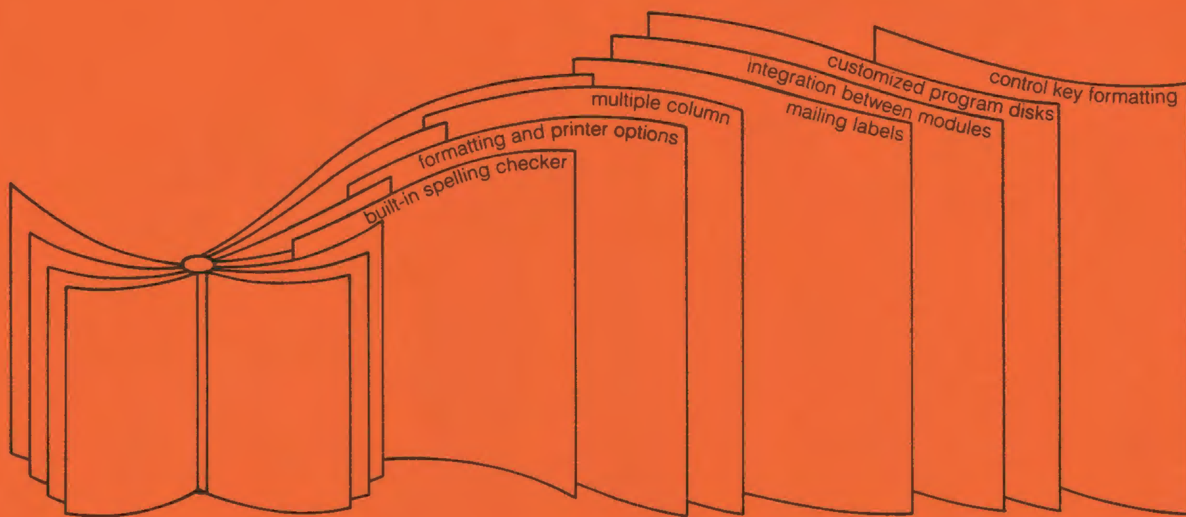
DragonSkin

International Society for Technology in Education

ISTE Publications

AppleWorks for Educators—Updated.

A Beginning and Intermediate Workbook.



AppleWorks® 3.0 continues to set new standards in the Apple II world. Linda Rathje does the same in the education world with AppleWorks for Educators, A Beginning and Intermediate Workbook.

The enhancements of AppleWorks 3.0 are clearly explained in *AppleWorks for Educators.*

The built-in spelling checker, formatting and printer options, multiple column, mailing labels, integration between modules, customized program disks, and control key formatting are just a few of the functions covered in this workbook.

Everything you need to work through AppleWorks (except the AppleWorks program) is included in *AppleWorks for Educators, A Beginning and Intermediate Workbook.* Add to your collection today.

AppleWorks for Educators for version 3.0 sells for $26.95 plus $4.50 shipping. *AppleWorks for Educators* for version 2.0 is available for $22.95 plus $4.50 shipping.

To order, contact: ISTE, University of Oregon, 1787 Agate St., Eugene, OR 97403; ph. 503/346-4414.

LOGO EXCHANGE

Volume 9 Number 1 Journal of the ISTE Special Interest Group for Logo-Using Educators September 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

From the Editor—News and Notes
Sharon Yoder 2

Monthly Musings— Come On In!
Tom Lough 3

Beginner's Corner—How Speedy is your Turtle?
Dorothy Fitch 4

Questions Please!
Frank Corley 7

Logo Ideas—Exploring Language
Eadie Adamson 8

Logo LinX—Taking Stock in Vocabulary
Judi Harris 12

Two Bridges Across the Rocket River
Ralph Olliges and Helen Miller 15

Logo Connection—Six Practical Multimedia Suggestions for Logo
Glen L. Bull and Gina L. Bull 19

Extra for Experts—Fractals I: Making Recursion Visible
Mark Horney 23

MathWorlds—Confluence: Logo, Educational Technology and Mathematics: Further Reflections
Sandy Dawson, editor 30

Logo: Search and Research— Logo or "Logo-like": The Great Debate
Douglas Clements 33

Global Logo Comments—Notes From Around the World
Dennis Harper, editor 35

ISTE Membership

	U.S.	Non-U.S.
	28.50	36.00

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.

Editor's Notes

News and Notes

It's hard to believe that three months have flown past since I completed the May issue of the *Logo Exchange* and now I'm busy with the September issue. Where does the time go? (I'm sure you are asking yourself the same question!)

New columns

Just as you find yourself at the beginning of another academic year, so too I find myself at a beginning. Each new volume of *LX* presents new challenges and new hopes. Just as you hope to do the best you can for your students, I hope to meet the needs of a wider variety of Logo users. To that end, we have added two new columns this year. The first is the "Extra for Experts" column that I promised last spring. Mark Horney begins his tenure as editor of that column with some of his own work on fractals. Even if you have worked with fractals before, I'm sure you will find Mark's ideas fascinating. We are hoping that this column will provide a place for those of you who have been programming in Logo for many years to share new challenges and new ideas.

More recently–in fact, only last week–the decision was made to include yet another new column in *LX*. This column, "Questions Please!" is being edited by Frank Corley. It is a column which can continue only if *you* participate. It will be used as a forum for Logo users to ask questions and for other Logo users to share answers. Frank doesn't intend to be the "Answer Man" for all of your questions. Rather, he will serve as the collector of both questions and responses to questions. Some of the questions will no doubt require only simple answers; others may be best answered by short articles submitted by *you*!

NECC

This latter column grew directly out of the SIGLogo meeting a NECC at the end of June. Approximately 50 people attended the meeting. We discussed many mutual concerns both for the Logo community as a whole and for *LX* in particular (see Tom Lough's "Monthly Musings" as well as Frank's "Questions Please!" column). Out of those discussions came a comittment from Frank, Donna Rosenberg, and a number of other teachers to share their questions and expertise with other readers of *LX*.

In addition to a productive SIGLogo meeting, there were a number of excellent sessions on Logo. Among these sessions was a panel entitled: "What does it mean to be Logo-like?" As each member of the panel presented his or her viewpoint, it became very clear that Logo is many things to many people. At one end of the continuum, some felt that Logo-like should mean only Logo because Logo itself is so special and unique. At the other end were those who felt that Logo-like should include anything that had the same underlying philosophy of education as Logo.

What is Logo-like?

Many of us were particularly struck by the remarks made by Terrapin Logo president David McClees at the opening of the panel. He cited five characteristics that he felt were necessary for something to be Logo-like. He said that Logo was

- flexible
- powerful
- open
- exploratory
- forgiving

The first four of these characteristics are commonly cited when discussions about the value of using Logo occur. However, everyone on the panel was struck by the use of the word "forgiving." While we talk about Logo's being user-friendly and providing easy-to-understand error messages, the word "forgiving" is not usually the word we use. Although David says he drew extensively from Brian Harvey's "Why Logo?" article (*BYTE*, August 1982), we all found David's choice of characteristics and examples particularly striking.

Since David's presentation, I have frequently thought about his description of Logo's being forgiving. How delightful! Isn't that really what we love about the way students relate to Logo? It forgives their mistakes, it forgives their misconceptions, it forgives their fear of technology...and on and on. I must also admit that the idea of a forgiving Logo struck me as particularly important as the Logo community grows more diverse. The divergent points of view presented in last May's *LX* as well as the differing points of view of the panel should be viewed as strengths, not weaknesses, by the Logo community (see Doug Clements' "Search and Research" column and Tom Lough's "Monthly Musings" column). As Logo users, we should be as forgiving, open, exploratory, and flexible as Logo itself is.

Sharon Yoder
ISTE
1787 Agate Street
Eugene, Oregon 97403
Ph: 503-346-4414
CIS: 73007,1645
BITNET: YODER@OREGON

Monthly Musing

Come On In!
by Tom Lough

Did you watch Roy Rogers cowboy movies when you were young? I did. Roy was a handsome singing cowboy who caught the fancy of a whole generation of youngsters back in the fifties. I remember plunking down many a quarter to view his adventures. (Two bits had a very different meaning back then!) When my childhood idol announced plans to open a chain of franchised fast food restaurants a few years ago, I followed developments with interest. One thing that intrigued me was the invitation to "Come on in!" which appeared in the jingle songs accompanying all of the television commercials. It had a nice comfortable ring to it. It was very...well...very Roy. I went on in and ate at Roy's the other day, and got to thinking about some things.

During the recent SIGLogo meeting held at NECC '90, one of the concerns we discussed was how to meet the needs of inexperienced Logo teachers. As a former *LX* editor, I am quite aware of this concern. It was the most challenging issue I faced during my editorial tenure.

Thinking back to my own early excitement, I can remember how inviting it was to plunge forward into the world of Logo enthusiastically, learning left and right. As my experience grew, so did the complexity of my projects, and the complexity of my writing. All too often, I found that, when I wanted to communicate with less experienced Logo teachers, I had to try harder and harder. Even worse, sometimes these teachers turned away in confusion and frustration, gaining nothing they could use in their classrooms.

I can see now that the spirit of Roy's "Come on in!" was often lacking where new Logo teachers were concerned. The world of Logo users is definitely not a closed society, a small family where everybody knows the punch lines and outsiders are barely tolerated. During the formative years of the *LX*, though, I'd guess that many promising teachers might have felt somewhat unwelcome.

It is absolutely essential that this journal always have something for the beginning Logo teacher. I am thrilled to report the spontaneous formation of a group of Logo teachers who are going to make sure that the spirit of "Come on in!" stays between the lines of this publication. These teachers, coordinated by Donna Rosenberg, want to act both as initiators of articles for beginning Logo teachers and as helpers for those who wish to write such articles. I am confident that this group will make a critical contribution as the year goes on.

As we begin another exciting school year, please consider contributing for beginners an idea, a tip, a brief project report, or a full-page article to the *LX*, your Logo magazine. If you are an experienced Logo user, remember back to your own needs when you were just beginning. If you are just beginning, write something about your current activities. Or, if you are having problems or wish to get some answers to your questions, send them to Frank Corley's new "Questions Please!" column. If more of us do this, I am confident that we will not only begin serving the needs of the beginning Logo teachers more effectively, but will also infuse the *LX* with a more welcome spirit, a warm invitation to "come on in!"

And, if Roy had ever learned Logo, I'm sure he would have added his "FD 100!"

Tom Lough
Founding Editor
PO Box 394
Simsbury, CT 06070

P.S. Donna's address is

Donna Rosenberg
3 Selman Street
Marblehead, MA 01945

About the Cover

The fractals on the cover are from the new column, "Extra for Experts." See this column for more details.

Do you have any material that you think should be on the cover of *LX*? Some of your own work? Some of your students' work? We are especially looking for unusual or interesting material for the cover. Send a dark, printed copy to Sharon Yoder at the address given at the end of "From the Editor." Be sure to include a description of how the work was developed, who did it, and a contact address.

Beginner's Corner

How Speedy is your Turtle?
by Dorothy M. Fitch

How many miles per hour does your turtle travel?

This may sound like a strange question, but when you stop and think about it, you and your students can actually figure out an answer! Since you need only know a handful of Logo commands, it's a good problem to solve at the beginning of the year! Or you can wait until the students practice conversion problems in their math class.

The Logo knowledge needed to answer this question is trivial. (If you can move the turtle forward, turn it right or left, and use a REPEAT statement, then you have all the tools you need.) The tricky part is not in using Logo, but in thinking about how to approach the problem (and, for some students, the process of doing the mathematical calculations). Logo can help out in the number-crunching department, but you need to be able to tell it the right things to compute!

Defining the problem

The first step in solving any problem is making sure that you understand it. When you ask your students this question, they will probably give you a blank stare. They may decide that you have finally flipped. So, you first want to get your students to help you rephrase the question so that it begins to make more sense. Thinking about the turtle traveling for miles is rather overwhelming, not to mention tiring for the turtle, who is already beginning to sweat just thinking about it. Try thinking in less strenuous terms—something you can actually measure, like feet, or better yet, inches or centimeters.

You probably don't have an extra hour in your day to spend timing the turtle, so consider a shorter duration of time, such as minutes, or even seconds.

So now, instead of trying to measure miles per hour, you can think about measuring inches per second, which is much more understandable and manageable!

Analyzing the problem

The next step is to figure out what you already know and then what you need to know in order to answer this question. Ask your students to identify what information you already have and what you need to find out.

You know that there are:

60 seconds in a minute
60 minutes in an hour
12 inches in a foot
5,280 feet in a mile

You also know that the turtle normally travels in dots, not in inches.

What you don't know is:

- exactly how many dots the turtle travels in an inch
- how far the turtle can travel in a second

Note: There are several ways to arrive at these answers. In fact, you can get widely differing results depending on which method you choose. This provides a good opportunity to discuss with your students why these differences exist. This column will suggest one method; at the end of the column, another method will be discussed. If you are comparing speeds of different computers, make sure to use the same method consistently.

How many dots does the turtle travel in an inch?

This is an easy question to answer. Clear your graphics screen and give the turtle a forward command, such as FORWARD 10. Did the turtle move an inch? How about FORWARD 50? Place a ruler against the screen so you can measure the turtle's trail. If you hide the turtle, you'll be able to measure more easily.

When you have figured out how many dots the turtle travels in an inch, turn the turtle RIGHT 90 and move it forward the same distance. Don't be surprised if the distance is not an inch. It doesn't matter for this exercise whether or not the two measurements are the same. In some versions of Logo, you can make them the same, however, by adjusting the aspect ratio, which changes the vertical scale of the screen. For example, in Terrapin Logo or Logo PLUS, you can change the aspect ratio by typing .ASPECT .9. To see the difference, clear the screen and retype the command to move the turtle an inch. You may need to experiment with the input number you use with .ASPECT to make them the same. (When Logo is first loaded, the aspect ratio is set to .8.)

If the numbers that move the turtle forward an inch are not the same on your screen, then decide in which direction you will measure the turtle's distance—either horizontally or vertically. In these examples, the measurement for an inch was taken horizontally using these commands:

```
RIGHT 90
FORWARD 32      (arrived at after some experimentation)
HIDETURTLE
```

On my Apple IIc+ computer, the turtle moves 32 dots in an inch. On my Apple IIGS, the forward distance is 35.

Record the number for your computer for future use.

How far does the turtle travel in a second?

To answer this question, I decided, after a period of experimentation, to repeat the following sequence of commands: hide the turtle, move it some distance and then show it. This causes the turtle to shoot forward several times, making a brief flashing appearance on the screen between moves. The next step is to find the right FORWARD number to make the flashes appear exactly one second apart. Here is what worked on my Apple IIc+:

```
REPEAT 10 [HIDETURTLE FORWARD 15000
   SHOWTURTLE]
```

Converting the units

Now you have all the information you need to compute the turtle's speed in miles per hour. This is merely an exercise in simple arithmetic, once you figure out how to manipulate the units. Don't forget to let Logo help you with the calculations.

If the following is true:
number of dots per inch = 32
number of dots per second = 15000

Then:

inches per second =

```
PRINT 15000 / 32
468.75
```

feet per second =

```
PRINT 468.75 / 12
39.0625
```

feet per minute =

```
PRINT 39.0625 * 60
2343.75
```

feet per hour =

```
PRINT 2343.75 * 60
140625
```

miles per hour =

```
PRINT 140625 / 5280
26.6335
```

To help students figure out how to handle the units, remind them to ask themselves questions like "Do I want the number of inches per minute to be bigger or smaller than inches per second?" If they think about the answers that they get, they can correct themselves.

Have students compare their results with each other. If they are using the same method of figuring out the two "unknowns" and they are using the same type of computer, then their answers should be approximately the same.

Why results may differ

If you find that the results are very different, then you may want to consider the following:

Computers run at different speeds. There are many different speeds at which IBM computers run, and there are three different speeds for Apple II family computers. Apple II+, IIe and IIc computers run the slowest; the Apple IIGS has both a normal speed (like the IIe) and a fast speed; and the Apple IIc+ is faster still (about 4 times the speed of the IIe). My turtle travels at 5.8 mph at on an Apple IIe, 21.8 mph on a IIGS at fast speed, and 26.6335 on a IIc+. (Your mileage may vary.)

If you choose to measure the turtle's speed in a different way, it can greatly affect your results. For example, another way to calculate how far the turtle can go in one second is to create a procedure called INCH that outputs the distance the turtle moves in an inch, like this:

```
TO INCH
OUTPUT 35
END
```

Then you can type:

```
FORWARD INCH
```

to move the turtle one inch. Type

```
REPEAT 10 [FORWARD INCH]
```

to move it ten inches. You can then determine the number you need to use with REPEAT to move the turtle forward for exactly one second. However, your results will be much slower. This is because Logo has to spend a lot of time doing internal "housekeeping" to locate and run the INCH procedure. Since Logo is an interpreted (as opposed to compiled) language, it has to keep looking up this information time after time, thus making for a pokey turtle.

Challenges:

Encourage your students to find the method of measuring that results in the fastest (or slowest) turtle. Make sure that they can explain their approach and their arithmetic!

Consider posing the initial question to small groups of students as an off-computer activity. They don't need a computer to come up with a plan for solving the problem. If the groups come up with different methods, have them predict which one will result in a faster turtle.

If you have a floor turtle or a Lego vehicle that you control with Logo, calculate its speed in miles per hour.

Other questions may arise as you investigate this idea. Follow their lead—you never know where you might end up. Happy turtle timing!

Dorothy Fitch has been the Director of Product Development at Terrapin since 1987. A former music teacher and consultant, she previously provided schools with inservice computer training and curriculum development, taught graduate courses in computer education, and directed a computer classroom for teachers and students. She is the coauthor of Kinderlogo and author of the Logo Data Toolkit. At Terrapin, she oversees the development of Logo languages and curriculum materials and makes presentations at local, regional, and national conferences. She can be reached at:

Terrapin Software, Inc.
400 Riverside Street
Portland, ME 04103

Questions Please!

by Frank Corley

My mother always told me that as a little boy I used to drive her crazy by asking questions about everything under the sun. When I asked her why she didn't tell me to be quiet and go away, she said, "I would never have done that. The only way you can learn is by asking questions." Maybe her attitude is why I grew up to be a Logo teacher. It is with this attitude that we begin this column in the *Logo Exchange*. My vision for this column is that it should contain an ongoing conversation between Logo users of all levels of experience and ability. I think that such a column is absolutely in keeping with the philosophy of education espoused by Logo practitioners. We learn by inquiry. The experienced users have a responsibility to the less experienced users to move them along in the process of learning Logo, to guide their inquiry with questions of their own and with answers to questions. That is why I am so grateful to the editor for accepting my offer to edit this column. This is a very important dialog between Logo users that I hope to mediate.

This column is for YOU!

This is your column. If you have any questions at all, please submit them to us at the address below. Questions may be technical questions about software, hardware, or programming. We welcome philosophical and educational questions about approaches to teaching Logo. Questions originating from your students might be particularly appropriate. We see at least two ways to keep the more advanced users involved in this column. The first is that they should submit answers to the questions, and I certainly hope that they will do so. The second is that we would eventually like to have two sections of questions in each column, one for elementary questions and one for advanced questions. Answers will be provided in separate sections as well. If users want to send in quiz questions for other users, they will also be considered.

How to submit questions

I would like to give some suggestions for submitting questions. Questions that provide a lot of information will be most easily answered. The minimum amount of information should be the version of Logo you are running and the system on which you are running it. Answers should be phrased similarly. The more context you give for a question the greater the likelihood that submitted answers will satisfactorily address your problem. The context could include, but not be limited to, exact syntax of an offending line of code, attempts you have made at answering the problem yourself, and perhaps the form of an expected solution.

We will print as many of the questions that I receive for which we have space, though of course I intend to edit them. I will try to phrase questions in as general a way as possible, so that one question may answer the problems of many individuals. Most all of us are or have been teachers in the Logo tradition, and so the next comment may not need to be made, but I will make it anyway. There are no stupid questions. Please do not be afraid to submit questions because you are afraid that they are too simple. How many times have you told that to a student? How many times have you told a student that for every person brave enough to raise his or her hand and ask a question there are several students in the audience who would have liked to ask the same question? How much larger is that audience here? The point is very important. Nothing is more disappointing to a teacher or speaker than to stand up in front of an audience and have his or her request for questions be met with a thundering silence. Please don't leave us in the lurch here. *Please ask questions. That is how you learn.*

Some questions to get us started

1. What are some ways to get fellow elementary school teachers to see Logo as an exciting tool that they can use in their busy curriculum?
2. What is the appropriate time to teach SETHEADING and SETPOS?
3. Is there a way of making a slide show of student work in Terrapin 3.0 or LogoWriter?
4. I know how to teach bottom-up programming with Logo; I need suggestions on top-down programming.
5. How do I write a command in LogoWriter to check equality of an input word to a certain word, and if equality holds, run a certain procedure?
6. What are some effective ways to introduce students to turns as opposed to movement (i.e., RIGHT vs. FORWARD)?

My immense thanks to the group of teachers and Logo users who met over breakfast early on the morning of the last day of NECC '90 in Nashville to help me formulate my ideas for this column and to submit a first round of questions. I hope we get these answered in a column in the near future.

Until then, are there any questions?

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

Logo Ideas

Exploring Language
by Eadie Adamson

French language teachers make occasional use of commercial software written in French to add an extra dimension to their teaching. Two years ago, instead of using a commercially prepared package, we elected to take advantage of our beginning French students' knowledge of LogoWriter by using the French version of LogoWriter. One class period per week half of the class used the computers. The computer class was team taught by the French teacher, Nicole Baldassini, and the computer specialist (me), while the rest of the class had enrichment work with the other French teacher (there were two sections of the French class, taught by two different teachers).

Exploring French LogoWriter

We began by focusing upon using the word-processing functions of LogoWriter. Right away the students were involved with using the French language, since the French version of LogoWriter uses appropriate French terminology for all its commands. In fact there is only one primitive, **rg**, that produces the same result in French as it does in English. Even the abbreviations differ. Very few of them produce the same results.

We took advantage of every opportunity to familiarize the students with the analogous French terms. We began with the opening LogoWriter screens and asked them to determine the meaning of the French text. Since the French screen looks just like the English screen with which students were familiar, students could recall the words on the English screen and infer the meaning of the text on the French screen. This immediately afforded the opportunity to add meaningful words to a beginner's vocabulary. For example, **ordinateur, aide, formes, contenu, nouvelle page**, can be learned from the first two screens.

After choosing a page, the first command students need to learn for writing is **ct**. Unlike the English version, this command does not clear the text. It stands for **cache tortue**, "hide turtle" in English. The abbreviation for **namepage, np** is the same in English or French, although the full term in French is different. We emphasized use of the full term, **nommepage**, to encourage more vocabulary development.

To begin to write, the up keys are also different. The term for "up" in the French version is **haut**. The keystroke, instead of Apple-U, is Apple-H. Now, the word for "apple" in French is **pomme**, so we began by stressing use of the French term as we helped students learn the commands: **pomme-H** will activate the writing cursor.

Begin with the Words

Since the students knew very little French at first, it made sense to us to begin by creating a French dictionary. We had several reasons for doing this. First, we thought it might be useful for them to keep a record of what they knew. This process was part of the written work that they might have done (creating a vocabulary list) had they not been working with computers. Secondly, we knew the students could then print their words to keep with them.

As soon as we began this process, some added benefits to working in this way became apparent. Nicole, the teacher, would pronounce a word. The students were to listen, then attempt to write it on their screens. Following this, Nicole spelled the word in French while the students checked their work and corrected as they could. (Sometimes, as our students became more adept at this process, we let the first one who appeared to have the word on the screen spell the word in French for the rest of the class, giving added practice in speaking.) They had to ask, in French, for a repetition when necessary. Finally, I would enter the word on my computer, which was connected to a demonstration monitor. The students then made a final check to see if their words matched. We began to realize that as a result of this process, the students were getting much more practice at both hearing and pronouncing the French alphabet than students normally did in a regular French class. Corrections were made quickly and easily. It was not at all difficult to ensure that each student had a correct word list, something not so easily or efficiently accomplished in a pencil and paper exercise. Here we could see the errors, whether the students saw them or not. Corrections were made within the lesson, rather than on papers returned to them later. Since LogoWriter pages can be easily saved, the dictionary exercise could be turned into a large permanent record to which students could add at any time.

Defining the Words

As work progressed, we decided that the dictionary could be more useful if we taught the students how to write a procedure to define the French words. They needed to learn another new term and keystroke for getting to the **flip side**, the **verso** in French. Again, new vocabulary emerged: **tournepage** is the French equivalent of **flip. Pomme-T** is the keystroke.

Introducing procedures allowed us to add still more new words. We discussed the elements of the English procedures and supplied the corresponding French terms: **pour** for **to**,

fin for **end**. Since we wanted the English word corresponding to the French word to appear in the Command Center we needed to learn the French for **type**, **tape**, and the French for **char** (as in **char 13**), **car**, as well. This afforded an opportunity for a digression on derivations and similarities of the French and English words. Then we began with the framework of a procedure to define a word:

```
pour    French word
tape [ here insert the English word ]
tape car 13
fin
```

(**tape**, **type** in English, puts text in the Command Center, leaving the cursor at the end of the line. **Tape car 13** (**type char 13**) moves the cursor down to the next line.)

The students quickly became adept at writing French procedures. Some even went so far as to seek out definitions for **nextscreen** (**ecrans**) and **prescreen** (**ecranp**), **top** (**debutpage**) and **bottom** (**finpage**), and defined these as well.

We began to encounter French accent marks and needed to learn the keystrokes for these. Here we discovered another advantage of working with the computer. Although in previous years, questions on the French accents had not been included on the mid-term exam, this time Nicole added them to the exam. To our amazement, every student was able to answer these questions correctly!

French Dialogues

As the students' vocabulary increased, we tried using the French computer time for a *dialogue*, in which the teacher either dictated sentences for the students to hear and type or asked questions to which the students would respond on the computer. More advantages to this approach emerged. Because the computers were ranged about the edges of the room, we could stand in the middle and observe. Previously the teacher would need to wait until papers were turned in and she had time to correct them in order to discover who did not know how to respond. Now we could see immediately if a single student was in trouble. Positive reinforcement and assistance could be given immediately, rather than after a considerable delay. When a dictation was complete, each student could turn in a printed copy. Sometimes we took time to make corrections in class, especially if we perceived immediately that the entire class was having trouble. We could then go over each question, explain the answers, and allow them time to make the corrections on their pages.

Last year, when we had a group of students who had serious behavior problems, we also found that working this way vastly increased their focus and attention. We began to conclude that perhaps computer time was the best time to plan for dictations or dialogues. No more doodled upon or empty papers turned in. Less frustration for the students. Instead of feeling helpless and hopeless, they could seek assistance immediately.

A Curious Response to Writing in French

We also found a curious response that is somehow related to the difference in the way students respond to writing on paper or writing with a computer. When Nicole asked questions, the students would no longer simply respond with what they had memorized from the text. "How do you say 'I am now in the computer room'?" "How do you say 'computer teacher'?" (In a response to a question asking who their teacher was.) There was the beginning of a real effort to use the language. Were they responding to the computer as an active being? We were never quite sure, but reading Sherry Turkle again, I begin to think they were "talking back" in some way vastly different from pencil and paper work.

A Hyper Dictionary

The dictionaries grew larger and larger. I began to realize that we were going to exhaust the memory of a page far too quickly if we continued to write procedures. Drawing on my own experience programming with hypertext (see *LX* May, 1989), I created a "hyper dictionary" in French. Pressing **Control-R** on a word would select the word and return its English equivalent in the Command Center. To save memory, the students needed to add a startup procedure (**depart** is **startup** in French) to get the tools and run a procedure we called **mots**. The procedure created names for the words. Expressed in French, it looked like this:

```
pour mots
donne "appartement "apartment
donne "voiture "car
    ... and so on
fin
```

The students entered their new words as pairs rather than writing procedures. For multiple words, we linked with an underline character, which I left out of the list of delimiters in French. On the front of the page **au revoir** was written **au_revoir**, which allowed it to be taken as a whole when selected. In the **mots** procedure, it was written:

```
donne "au_revoir "goodbye
```

This strategy allowed us to incorporate common French phrases in the dictionary as well as single words.

Mother's Day Fun

As we headed towards spring, we decided to have a little fun. Right after spring vacation, we began working on French Mother's Day cards. This was fun for the students and also allowed us to let the parents know in a rather nice way that their children were working with French on the computer as well as with English. We made color prints of each card for the students to take home, and then made extra copies of the most interesting to use in a display in the hallway, eliciting much comment from faculty and students alike. The cards below were done by Wesley Stanton and Freddy Victoria.

Exploring Programming in French

This past spring I also used the last three or four weeks of computer classes to explore programming in French LogoWriter. We began by talking about exploring new ideas and being adventurous. I asked them how they handled new situations and quickly made the point that we were now going to explore something about which they already knew a bit, that what would be left to find out would be what was new and different. I emphasized that LogoWriter worked in exactly the same way, except that the language for the commands themselves would be different. I gave each student some representative samples of the French activity cards as well as copies of some of the pages from the first project book. Their challenge was to produce a graphic and then, if they wished, some animation and soundin just a few short classes.

We began collecting new terms. Every time someone called out "How do you.... in French?" we would find the answer on the handouts or I would supply it, but only if it wasn't there. I kept a list on my computer in both English and French. By the end of the first class we had quite a long list! This turned out to be a very interesting exercise. Some students in one class felt enormously threatened by the very idea of working in French and took refuge in endlessly creating shapes, the only place in which *no* language needed to be used. My other class, however, made enormous strides in their first class, and finished by creating some simple animations and some nice scenes that we printed in color for them to keep.

Logo Express, Too!

At the end of the year I received via Logo Express some French letters from Meadowbrook School in Quebec. We sent them a few answers, but the school year ended too swiftly for us to develop this new aspect. It does however add the potential for active exchanges in foreign language, both for correspondence and for exchanging LogoWriter pages.

I think we have begun to explore only the tip of the iceberg in this exciting new application. The added dimensions to using LogoWriter in the context of the study of French are nonetheless exciting and without a doubt worthwhile. Now, what about Spanish... or Dutch...or... (LogoWriter versions exist in a number of foreign languages, although the LogoWriter materials have so far been produced only in English, French and Spanish.)

If you are interested in the French dictionary idea, please write me (include a stamped, self-addressed envelope, please!) for a copy of the handout I created when I presented these ideas at NECC '90 last June in Nashville. Meanwhile, why not try exploring Logo in a foreign language? It's really fun!

About the software: LogoWriter in three languages and Logo Express, a telecommunications package written in Logo, are products of Logo Computer Systems, Inc.

Eadie Adamson is now teaching at The Dalton School in New York City, where she now has the opportunity to work with boys *and* girls! Eadie's students at Dalton will be working this year with LogoWriter, LEGO TC logo, and with Logo Express.

Eadie Adamson,
The Dalton School
108 East 89th Street,
N.Y., N.Y. 10128
(212) 722-5160

1199 Park Avenue
Apartmentt. 3A
New York, N.Y. 10128

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 have been combined to celebrate ECCO's tenth anniversary.

There will be two strands of Logo sessions during the two days of the conference.

Thursday
If There Had Been Logo I Might Have Learned Math (Gary Stager)
Entry Level Logo (Albie Weiss)
Art and Technology with LEGO TC logo (Cathy Helgoe and Eadie Adamson)
Cooperative Learning Activities with Logo (Greg Brownell)
The Writing Side of LogoWriter (Eadie Adamson)
The Expressive Arts Meet Logo (Leslie Thyberg)
Logo Language Experience (Ron Moxley)

Friday
Terrapin Robotics (Dorothy Fitch and David McClees)
What's New With Logowriter 2.0 (Carl Fechko)
Logo Based Geometry (Douglas Clements and Michael Battista)
Logo Script for a Dynamic Classroom (Ihor Charischak)
Getting Acquainted with Logo Plus (Dorothy Fitch)
Developing Scientific Thought With Logo (Gary Stager)
LogoWriter Lessons for Cooperative Creations (Jandy Bird)

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

For further information, or to request the registration form, contact

ECCO
1123 S.O.M. Center Road, Cleveland, Ohio 44124
(216) 461-0800.

Logo LinX

Taking Stock in Vocabulary
by Judi Harris

Robin Williams once quipped that if you look in the dictionary under *redundant*, it says "see redundant." How redundant do your students' vocabularies seem now that school has begun again? Are you looking for a way to motivate them to use new words in their writing? If so, perhaps you might encourage them to take stock (quite literally) in certain of their favorite words.

If we drastically over-simplify our understanding of the workings of the stock market, we might state that newer stocks are often less expensive than well-established ones, and that share prices increase as stocks become more popular. If we encouraged our students to "invest" parenthetically in words that might appear in class members' writing, perhaps they would "buy low," taking shares in new vocabulary words that are introduced, then use the new words so that the prices of their word stocks would rise. Next week's spelling words might become quite popular this week; this month's science vocabulary might be more eagerly studied; words from the current history lesson might be used in language arts work without specific requirements to do so.

A vocabulary stock market simulation might take its fluctuating prices from the frequencies of different words used by a group of students in their expository, communicative, drill-and-practice, and creative writing completed over a certain period of time. Students could invest in words using much the same prediction and risk-taking strategies as are employed by successful stock market investors, managing their "word portfolios" according to their language explorations. A simple set of LogoWriter tools would take the tedium out of the share maintenance process, allowing students to concentrate on using new words and phrases and explore probability and mathematical problem-solving strategies.

Tabulating Word Frequencies

One of the most useful attributes of computer tools is their seeming patience with tiresome calculation. Perhaps the most tedious and error-prone component of the activity described above is summing the number of times that words appear in a class' written products over, say, a week's time. Since the educational objectives of a simulation such as this would probably involve higher-level mathematical problem-solving processes rather than simple tabulations, it is appropriate and time-efficient for the computer to compute word frequencies. The following four LogoWriter procedures will perform this function.

```
TO COUNT.WORDS.IN :LIST
IF EMPTY? :LIST [STOP]
IFELSE MEMBER? FIRST :LIST
   :WORDS.USED [TALLY.WORD.USED
   :LIST] [RECORD.NEW.WORD :LIST
   TALLY.NEW.WORD :LIST]
COUNT.WORDS.IN BUTFIRST :LIST
END

TO TALLY.WORD.USED :LIST
MAKE WORD " (FIRST :LIST) (THING
    (WORD " FIRST :LIST)) + 1
END

TO RECORD.NEW.WORD :LIST
MAKE "WORDS.USED LPUT FIRST :LIST
   :WORDS.USED
END

TO TALLY.NEW.WORD :LIST
MAKE WORD " (FIRST :LIST) 1
END
```

Processing Manuscripts for the Stock Market

The :LIST that the superprocedure COUNT.WORDS.IN uses is a piece of word processed writing that a "stockholder" composes with a ProDOS word processor such as AppleWorks or FrEdWriter. The process with which you can transfer word processed files saved to disk into LogoWriter code is surprisingly simple. Let's say that one of your students has written the following piece and saved it to disk with FrEdWriter; in actuality, it is one of my favorite Shel Silverstein poems.

Hug of War

I will not play at tug of war.
I'd rather play at hug of war.
Where everybody hugs
Instead of tugs,
Where everyone giggles
And rolls on the rug,
Where everyone kisses,
And everyone grins,
And everyone cuddles,
And everyone wins.

(from *Where the Sidewalk Ends*, page 19)

To read this ProDOS file into LogoWriter,

1. Flip to the procedures side of your VOCAB.STOCKS page.
2. Hold down the Open-Apple key and press the letter D to make the cursor jump *down to the Command* Center.
3. Type LOADTEXT "filename, substituting the poem's file name after the quotation mark.
4. Once you see the text of the poem above the command center, hold down the open-apple and press the letter U to move the cursor *up above the Command* Center.
5. Change the title of the poem to: TO HUG.OF.WAR
6. Insert an OUTPUT command and opening bracket just before the first line of the poem: OUTPUT [I will not play...
7. Insert a closing bracket after the last word of the poem:
8. Insert the END command on a separate line after the closing bracket.

Now that the piece is in the form of a procedure that outputs a single list containing the writing itself, COUNT. WORDS.IN can be invoked by typing COUNT.WORDS.IN HUG.OF.WAR. The computer will then tally the frequency of each word's appearance in the poem, storing the totals for each word in global variables named with the words themselves. For example, the word EVERYONE in the example above would become the variable :EVERYONE with a corresponding value of 5.

Each time a new piece of writing is processed with the COUNT.WORDS.IN procedure, the values for word variables are updated with TALLY.WORD.USED, and new word variables are created as necessary with RECORD. NEW.WORD. As long as the VOCAB.STOCKS page is re-saved each time that it is used, these global variable values (along with :WORDS.USED, the variable which stores the master list of words in which shares may be purchased) are recorded to disk in updated form.

Checking the Market

Once students purchase shares in specific words (the prices to be determined by the tabulated frequencies of word occurrence in writing already processed with COUNT.WORDS.IN), they can use another LogoWriter tool to check on current "prices" of their word investments.

```
TO STARTING :IGNORE :LETTER :IGNORE
   :LIST
IF EMPTY? :LIST [PRINT CHAR 13 STOP]
IF :LETTER = FIRST FIRST :LIST [PRINT
   (SENTENCE FIRST :LIST "- (THING
   FIRST :LIST)]
STARTING :IGNORE :LETTER :IGNORE
   BUTFIRST :LIST
END

TO IN
OUTPUT "
END

TO WITH
OUTPUT "
END
```

For example, typing

```
STARTING WITH "W IN :WORDS.USED
```

would yield:

```
will - 1 war. - 2 Where - 3 wins. - 1
```

Word Portfolios

Additional LogoWriter tools can also help you and your students to keep track of which words are in each stockholder's "word portfolio." The following procedures, invoked with superprocedures UPDATE.PORTFOLIO.FOR and LIST.HOLDINGS.FOR, allow new word share acquisitions to be added to the computer's stock records, and print out current "market prices" for an individual stockholder's portfolio. Both are invoked in similar format:

```
UPDATE.PORTFOLIO.FOR "DONALD.TRUMP
LIST.HOLDINGS.FOR "AMELDA.MARCOS

TO UPDATE.PORTFOLIO.FOR :STOCKHOLDER
IF NOT NAME? :STOCKHOLDER [MAKE
   :STOCKHOLDER [ ]]
CC
TYPE SENTENCE [In what words do you
   own shares,] WORD :STOCKHOLDER "?
TYPE CHAR 13
UPDATE READLISTCC :STOCKHOLDER
END

TO UPDATE :WORD.LIST :STOCKHOLDER
IF EMPTY? :WORD.LIST [STOP]
```

```
IFELSE MEMBER? FIRST :WORD.LIST
    :WORDS.USED
        [HOLDINGS.FOR :STOCKHOLDER
          :WORD.LIST]
        [CC TYPE SENTENCE [Sorry,
          there are no stocks for the
          word] WORD FIRST :WORD.LIST
          ". TYPE CHAR 13]
UPDATE BUTFIRST :WORD.LIST
    :STOCKHOLDER
END

TO HOLDINGS.FOR :STOCKHOLDER
    :WORD.LIST
IF NOT MEMBER? (FIRST:WORD.LIST)
    THING :STOCKHOLDER
    [MAKE :STOCKHOLDER SENTENCE (FIRST
       :WORD.LIST) (THING :STOCK-
       HOLDER)]
END
```



```
TO LIST.HOLDING.FOR :STOCKHOLDER
IFELSE NAME? :STOCKHOLDER
    [LIST.WORDS.FOR :STOCKHOLDER]
    [CC TYPE (SENTENCE [Sorry,]
    :STOCKHOLDER [is not in my records
    as owning any word stocks.]) TYPE
    CHAR 13]
END

TO LIST.WORDS.FOR :STOCKHOLDER
CT
PRINT (SENTENCE WORD :STOCKHOLDER "'s
    [Word Stocks:])
INSPECT THING :STOCKHOLDER
END

TO INSPECT :LIST
IF EMPTY? :LIST [STOP]
PRINT (SENTENCE WORD FIRST :LIST ",
    [which costs] WORD "$ THING FIRST
    :LIST [today.])
INSPECT BUTFIRST :LIST
END
```

The classroom organization, regulation, word trading frequency, investment tip information distribution, and initial investment funds allocations are, of course, up to you and your student stockholders. You may even want to challenge some of your more adept LogoWriter programmers to create procedures that will graph individual word stock activity over time, or keep track of shares bought and sold for individual word share holders. Perhaps, though, it would be wise to keep Thomas Hobbes' words in mind as your students watch their word investments fluctuate:

> Words are wise men's counters, they do but reckon by them; but they are the money of fools. (*Leviathan*, 1651, 1.4)

Reference

Siverstein, S. (1974). *Where the sidewalk ends,*. New York: Harper & Row.

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.

Two Bridges Across the Rocket River

by Ralph Olliges and Helen Miller

Introduction

George Polya (1973) stated that to solve any problem requires a four-step process. First, one must understand the problem. Second, one must devise a plan to attack the problem. Third, one must carry out the plan. Fourth, one must look back and check the result. Either one has solved the problem or else one must devise a new plan of attack.

Jean Piaget (1976) discovered that students learn best when they begin by having concrete experiences and when they are interested in learning. If the student can physically manipulate items, they will be able to mentally construct relationships. This provides a natural progression from the concrete to the abstract.

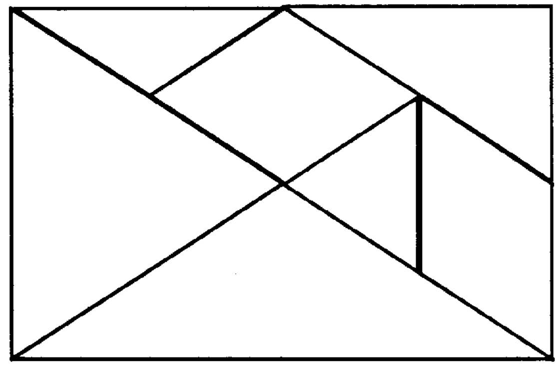
Seymour Papert (1980) designed Logo to allow for this progression from the concrete use of the Turtle on the floor to a more abstract use of the Turtle on the computer screen. Papert's purpose was to provide children with the means by which they could be actively involved in the learning process and be self-directed in extending their own learning.

Tangram puzzles will enable teachers to implement Polya's four-step process while engaging the interest of the students. In addition, these ancient Chinese puzzles provide students with concrete experiences. The concrete experiences can progress to an abstract level of thinking using the computer. Basic left/right orientation can be taught at an abstract level. Further, the student must decipher both direction and position for the turtle prior to using our Logo procedures.

Understanding The Problem

A tangram is an ancient seven-piece Chinese puzzle. All seven pieces can be cut from a single square. The cuts consist of five right triangles, one square, and one parallelogram. These seven polygons can be arranged in many different compositions to provide extremely provocative pictures. The trick is to use all seven pieces exactly once to construct some design – birds, animals, buildings, boats, bridges, and so forth. (See Read (1965) for some interesting designs.) Bridges and sailing ships were the earliest types of puzzles depicted. The first book of tangrams was published in the Treaty Ports of South China in 1813.

The students need to make their own puzzle pieces.



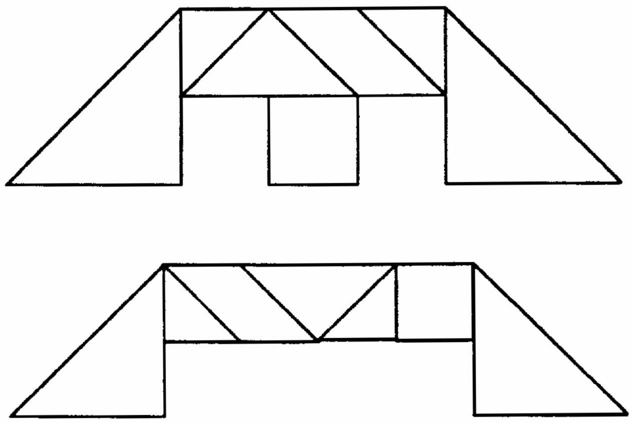
Allow the students to create their own shapes. Once they have arranged the seven pieces on a white sheet of paper, instruct them to draw around the entire shape before attempting another one. When everyone has had the opportunity to experiment making shapes, the students should exchange papers. The objective is to see if the shapes made by one person can be duplicated by another. The final step is to transfer the seven-piece puzzle to the computer and work in the abstract. To make the transition easier procedures LTPARA, RTPARA, LTTRI, RTTRI, and SQUARE were written.

A Sample Story

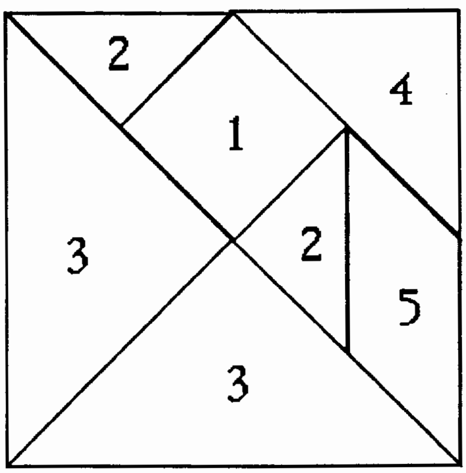
The raging Rocket River raced between the two cities of East Rocket and West Rocket. Year after year the citizens have struggled to cross the Rocket River by paddling their small boats. The boats are attached to a long rope stretched across the river, but it is a long, dangerous trip by boat. Now the people in East Rocket and West Rocket have decided that each town would construct a bridge to walk across the deep, rough river. One of the bridges will take the people across and the other will bring them home. Everyone agreed that the bridges will be a wonderful solution to their problem. However, there exists a slight difficulty. The citizens of East Rocket have long, narrow boats—boats that they want to continue using on the river. The citizens of West Rocket, also, have boats. Their boats are wide, much wider than the boats that people in East Rocket use to traverse the river. Each town has the same materials and construction blocks that are the same size and shape—five right triangles, one square, and one parallelogram. One of the bridges must allow very wide boats to float safely underneath. The second bridge must allow narrow, long boats to float safely underneath. The bridge for the wide boats will be located at a place spanning the river that will allow the wide boats to leave their harbor and sail toward the ocean. The bridge for the narrow boats will be located upstream. The narrow boats will have to sail under two bridges in order to reach the ocean.

Devising A Plan

The two different bridges that need to be designed are pictured below.



Recall that the only building blocks that can be used are seven specific polygons: five right triangles, one square, and one parallelogram. In fact, upon closer examination one can see that instead of seven unique polygons, there exist only five polygons of different sizes. Let L represent a length of L units. Then, the five polygons consist of (1) one square with side = L, (2) two right triangles with legs = L, (3) two right triangles with legs = 2L, (4) one right triangle with leg = R(2) L, and (5) one parallelogram with adjacent sides = L, R(2) L.

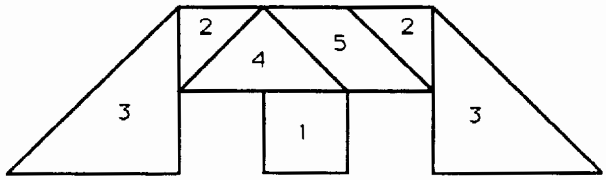


Logo procedures were written for a square, a parallelogram, and a right triangle. To make it easier, the triangle and parallelogram procedures were written twice, depending whether they pointed left or right.

Carrying Out The Plan

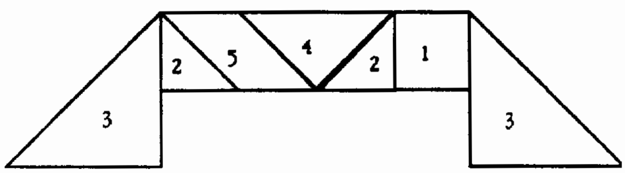
The bridges were from Read's [4] book on tangrams. The Logo procedures were created by the authors. First, one must examine the bridges and divide them into component parts that use our building blocks.

In BRIDGE1, two right triangles will be needed to form the entrance and exits. The square will be needed for the middle support base. Thus, the three remaining right triangles and the parallelogram will be used for the bridge span.



Provide the students with the procedures LTPARA, RTPARA, LTTRI, RTTRI, and SQUARE. Do not give the students the procedures BRIDGE1 and BRIDGE2. Instead guide the students through the procedure BRIDGE1 as follows: Have the students type PENUP, LEFT 90, FORWARD 70, RIGHT 90, BACK 50, PENDOWN, LTTRI 70. One bridge support should be drawn. Now ask the students where they must be located at to draw the next shape (LTTRI 35)? How do we get there? Following the discussion, the students should type in FORWARD 70, RIGHT 180, LTTRI 35. Continue in this manner until the entire BRIDGE1 procedure has been constructed. This is an excellent exercise in direction and orientation. Once BRIDGE1 has been constructed, let the students construct BRIDGE2 on their own.

In BRIDGE2, once again, two right triangles will be needed to form the entrance and exit of the bridge. However, this time no middle support base is present. Hence, three right triangles, a square, and a parallelogram will compose the bridge span.



Looking Back And Checking The Result

We have accomplished what we have set out to build. Perhaps there is another way to build a bridge. In figure 6, Turtle constructed a bridge that both long, narrow and very wide boats can use. Read (1965) suggests many ways to build bridges, birds, boats, and so forth. Using Logo, one can implement many of these designs very easily. More importantly, by using this activity, students will build some important problem solving skills as well as gain needed practice in both direction and orientation.

References

Papert, S. (1980). Mindstorms. New York: Basic Books, Inc.

Polya, G. (1973). How To Solve It. (2nd ed.). Princeton, N.J.: Princeton University Press.

Piaget, J. (1976). To Understand Is To Invent. New York: Penguin Books.

Read, R. C. (1965). Tangrams: 330 Puzzles. New York: Dover Publications, Inc.

Logo Procedures for Drawing the Two Bridges.

```
TO BRIDGE1
PENUP
LEFT 90
FORWARD 70
RIGHT 90
BACK 50
PENDOWN
LTTRI 70
FORWARD 70
RIGHT 180
LTTRI 35
LEFT 90
FORWARD 35
RIGHT 45
RTTRI 49
LEFT 45
FORWARD 35
RIGHT 90
LTPARA 35
LEFT 90
FORWARD 35
RIGHT 90
RTTRI 35
FORWARD 70
LEFT 180
RTTRI 70
FORWARD 35
LEFT 90
FORWARD 35
LEFT 90
SQUARE 35
HIDETURTLE
END
```

```
TO BRIDGE2
PENUP
LEFT 90
FORWARD 70
RIGHT 90
BACK 50
PENDOWN
LTTRI 70
FORWARD 35
RTTRI 35
RIGHT 90
FORWARD 35
LEFT 90
LTPARA 35
RIGHT 90
FORWARD 35
LEFT 135
RTTRI 49
RIGHT 135
FORWARD 35
LEFT 90
LTTRI 35
SQUARE 35
RIGHT 90
FORWARD 35
LEFT 90
BACK 35
RTTRI 70
HIDETURTLE
END
```

```
TO LTPARA :L
LEFT 45
FORWARD (:L*SQRT 2)
RIGHT 135
FORWARD :L
RIGHT 45
FORWARD (:L*SQRT 2)
RIGHT 135
FORWARD :L
RIGHT 90
END
```

```
TO RTPARA :L
RIGHT 45
FORWARD (:L*SQRT 2)
RIGHT 45
FORWARD :L
RIGHT 135
FORWARD (:L*SQRT 2)
RIGHT 45
FORWARD :L
RIGHT 90
END
```

```
TO LTTRI :L
FORWARD :L
LEFT 135
FORWARD (:L*SQRT 2)
LEFT 135
FORWARD :L
LEFT 90
END
```

```
TO RTTRI :L
FORWARD :L
RIGHT 135
FD (:L*SQRT 2)
RIGHT 135
FORWARD :L
RIGHT 90
END
```

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

Helen Miller is an elementary teacher with 17 years experience. Presently she is a doctoral candidate in Education at Saint Louis University in the area of Curriculum and Instruction.

Ralph Olliges has a doctoral degree in Education with a Computer Emphasis from Saint Louis University. He has taught Computer Science courses for the Mathematics Department as well as computer-related courses in the teacher preparation program for the Education Department at Saint Louis University.

Dr. Ralph Olliges
Department of Mathematics and Computer Science
Saint Louis University
221 N. Grand Blvd.
St. Louis, MO 63103

Look at our Logo list!

Introduction to Programming in Logo Using LogoWriter $18.95

Introduction to Programming in Logo Using Logo PLUS........$18.95

LogoWriter for Educators: A Problem Solving Approach.......$10.95

Logo PLUS for Educators: A Problem Solving Approach.......$10.95

Logo users at all levels benefit from these ISTE selections.

The *Introduction to Programming* books, written by Sharon Yoder, provide beginners with a Logo base to build on and experienced users with a reference to rely on. Both are excellent resources for teacher training or introductory computer science classes.

New from ISTE, *LogoWriter (Logo PLUS) for Educators: A Problem Solving Approach* takes Logo learning to new depths. The focus is entirely on learning and practicing general problem solving skills while using Logo. Great for beginning programming experience. Appendices include keystroke summaries, turtle shape pictures, and a quick reference card. Written by Dave Moursund and Sharon Yoder.

To order, contact: ISTE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905; ph. 503/346-4414. (Please add $3.25 shipping for single copy orders, $4.50 for up to 4 copies, and $6.00 for 5 copies)

Logo Connections

Six Practical Multimedia Suggestions for Logo

by Glen L. Bull and Gina L. Bull

This column is about Logo connections:

- connections between the Logo programming language and other kinds of software and hardware, and,
- connections between the teaching philosophy represented by Logo and related instructional philosophies.

Human languages fall into two categories: dead languages like Latin which are rarely spoken, and living languages. All living human languages change and evolve in response to changing conditions, in contrast to the dead languages which remain forever the same. Computer languages are much the same as human languages. They evolve in response to changing technologic conditions, or gradually become unused. Logo has evolved considerably during its lifetime. The first versions of Logo did not have turtle graphics, because the computer terminals used for Logo in the early 1970's did not support any kind of graphics. Logo output was printed on Teletype-terminals. When inexpensive graphics terminals became available, the now-familiar graphics features of the language were added.

Hardware has a surprisingly strong influence on the potential shape of a computer language. The first version of Logo produced for a commercially-available microcomputer was developed for the Texas Instruments TI 99/4. This early microcomputer had a hardware graphics chip that made it easy to support moving objects. Not surprisingly, the TI version of Logo supported sprites, and Logo programs on this machine were apt to be filled with planes, rockets, birds, planets, and other moving objects.

The Apple II and IBM microcomputers that followed did not have a graphics chip of this kind, and no widely-used Logo available today has sprites as fully-featured as those in the original TI Logo. (A "Sprite Logo" developed for the Apple required a special hardware board, and was never widely used or long supported.)

Hardware features that are taken for granted today were often considered specialized or exotic at first. The original Apple II computer only had upper-case text, and required a special hardware adaptation to display lower-case letters. Disk drives and printers were far from standard equipment. One large school district in Virginia ordered Atari computers with cassette tape drives, because they could purchase "three Atari's for every Apple equipped with a disk drive." In many cases printers were considered too expensive for ordinary classroom use. At the time, we found it somewhat bemusing to see teachers with high-technology computers forced to copy text from screen much like monks in medieval times, due to lack of printers. We argued then, and still feel, that it does not make sense to provide computers without disk drives and printers. Granted, at that time a floppy disk drive cost $600 and a printer cost more than $800. But, we felt it made more sense to provide greater functionality, even if it meant that it was necessary to extend the schedule required to equip all classrooms with computers.

Logo was another feature considered "too expensive and too exotic" by many. The standard Apple of that time only had 48 kilobytes of memory. Logo required 64 kilobytes of memory. Therefore it was necessary to purchase an additional "language card" (which essentially consisted of 16 kilobytes of memory) before it was possible to run Logo. Further, some argued that the cost of Logo itself was an unnecessary expense, since the Apple (and later the IBM) came equipped with the built-in BASIC computing language.

At that time we told school administrators that buying a computer and failing to buy software for it was akin to buying a Mercedes and failing to buy gasoline to run it. If it was also necessary to purchase an additional 16 kilobytes of memory to run Logo, we felt this was a worthwhile investment.

Today, when most computers come routinely equipped with a half megabyte to a megabyte of memory, it might be thought that these issues would have vanished. Most computers, after all, were eventually equipped with sufficient memory to run Logo. In fact, as memory expanded beyond 64 kilobytes it became possible to develop some very good versions of Logo indeed.

However, in today's world and its renewed concern for the ecology, these issues have not vanished–they've just been recycled and reissued in a different form. With today's technologies it is possible to do marvelous things with Logo that might have been considered exotic a decade ago. In some cases it is necessary to acquire software or an inexpensive hardware enhancement to take advantage of all the possibilities.

In this column we are going to suggest a half-dozen enhancements for your computer that will allow you to do multimedia with Logo. We have titled the column "Six Practical Multimedia Suggestions for Logo," because in most cases the cost is relatively modest. We thought the beginning of the school year would be a good time to make some suggestions, so you can place a request with the PTO or superintendent's office. Fred D'Ignazio has coined the phrase "scavenged multimedia" to describe multimedia activities that can be developed with hardware and software that is already available in schools. We can't promise that you'll find you already have everything we mention, but these items are the next best thing.

The term "multimedia" literally means "working with more than one medium." These suggested enhancements will permit you to extend the uses of Logo, and combine it with other media. We have listed an approximate cost for each of these items, so that you will have a basis for budgeting. By way of comparison, it is interesting to note that at the time Logo was first introduced, a memory upgrade was $300, a disk drive was $500 to $600, and a printer was $500 to $1000.

1. Upgrade Logo (cost: varies)

As we mentioned, there have been many enhancements to computer hardware over the last decade. New and better versions of Logo have been developed to take advantage of these hardware enhancements. Yet a surprising number of school systems are still requiring their teachers to use Version 1.0 of Terrapin or LCSI Logo. (Some of the older versions of Logo do not even allow use of lower-case letters.) New and better versions of Logo such as Logo Plus (Terrapin) or LogoWriter (LCSI) take advantage of the greater power of today's computers.

School systems sometimes say that it is too expensive to pay to upgrade their Logo licenses. We haven't yet heard it, but any day we expect someone to say, "If the original version of Logo was good enough for me when I was growing up, it should be good enough for today's students." (And besides, they have lost the license and can not find the serial number needed to obtain the upgraded version.) Yet, this is one of the best investments that it is possible to make. No matter how much power the computer has, if ancient versions of software are used, teachers and students are using the technology of 1980 even though it is now 1990. It doesn't matter if the computer can display lower case letters or 256 different colors if the software does not take advantage of those features.

Site licenses and 10-packs make it very reasonable to acquire Logo for a school system today, and the cost of upgrading a license is even less. This might be a good year to consider purchasing one or two fewer computers for the school system, and upgrading to a more modern version of Logo.

2. Acquire a Paint Program (cost: about $100)

It is probable that future versions of Logo may incorporate some types of paint and drawing tools, similar to those found in many other programs. But there's no need to wait. The newer versions of Logo have an "Import Picture" command that allows you to create an image with a paint program, and then import it into Logo. (It is possible to import pictures into many of the older versions of Logo, but using a built-in command makes the process quick and easy.)

If you have a mix of Apple IIe and Apple IIgs computers in your school, you may want to consider a program called "8/16" which can be used to create 8-bit images for the Apple IIe computer and 16-bit images for the Apple IIgs computer. (There are also many good paint programs for IBM and other brands of computers as well.) The images that can be created with today's paint programs are simply stunning–there's no other way to describe them. In a future column we will review several of these paint programs, but almost all of the newer ones have excellent capabilities. If you have an emerging artist in one of your classes, you may want let them experiment with this new medium. Then, images created with this tool can be imported into Logo. Careful planning will result in the best of both mediums–Logo and paint.

We should also mention, in this context, that many versions of Logo also have an "Import Text" feature as well. Therefore if a document has already been created in a word processor such as *AppleWorks*, it is possible to import it into Logo without retyping it. Just save the file in a text format (sometimes called an "ASCII" file), and then import it into Logo. This would allow members of an English class to create a story using a word processor, for example, and then turn it into a text adventure using Logo commands.

In order to use a paint program, you will need one peripheral–a mouse. Almost all Apple and IBM computers now being sold are equipped with mice. A mouse port is a built-in feature on the Apple IIgs and all of the IBM PS/2 computers. However, if none of the computers in your school have mice, you can acquire this peripheral for less than $100. At first you will not need one for every computer –just one or two to set up "graphics stations" in some area of the school.

3. Acquire a Video Digitizer (cost: about $300)

A video digitizer is one of the most powerful instructional tools available. This tool makes it possible to create a computerized image of anything that can be captured with a video camera or recorded with a videocassette recorder. For example, an earth science teacher can point a video camera at different samples of minerals, and digitize the various samples. In life sciences, any type of flower, tree, shrub, or plant can be captured as a computer image. Members of the class can digitize one another for a "living yearbook."

Once the image has been captured, it can be edited with a paint program, or imported directly into Logo. Inexpensive video digitizers are now available for most computers. Digital Vision makes models of their "Computer Eyes" digitizer for the Apple IIe, Apple IIgs, IBM, and Macintosh computers. This tool will allow you to capture dozens of images from real-life models.

A digitized image can be edited with a paint program, or imported directly into Logo. Now we have three tools in our multimedia arsenal–Logo, a paint program, and a digitizer. All three tools can work together to enhance each other's capabilities.

In order to make good use of the digitizer, you will need either a videocassette recorder (VCR) or a video camera to acquire images for the digitizer. However, most school systems already have one or both of these items, so the only new acquisition required is the digitizer itself. (This is a good illustration of the concept of scavenged multimedia.)

4. Begin Working with Robotics (cost: about $500)

For anyone who has been marooned on a desert island (without your copies of *LX*) for the last 10 years, LEGO TC logo is a refinement of Logo that allows Logo programs to control LEGO gears, motors, lights, and sensors. This allows classes to beginning experimenting with robotics for far less than would have been dreamed possible a few years ago.

It is also the application that students in a Logo class often find most exciting. It is true that it is difficult to keep track of all of the LEGO pieces in a classroom, and that some classroom management skills are required to keep them from falling into the radiator or being eaten by the vacuum cleaner. However, we have never met a teacher (or class) who regretted getting a LEGO TC logo kit.

5. Purchase a Videodisc Player (cost: about $600)

A videodisc is much like a compact music disc, except that it is a little larger and contains images rather than sound. A single videodisc has the capacity to hold more than 50,000 high resolution pictures. In one sense, a videodisc might be thought of as a slide projector that can hold 50,000 slides. A videodisc can also display moving pictures as well, so it might be considered a combination slide and film projector all rolled into one. Many of today's videodisc players, such as the Pioneer 2200, can be controlled by a computer as well. Therefore, if you could control your film projector with a computer, and if the tray of your slide projector could hold 50,000 slides, then you would have the equivalent of a videodisc player.

A wide range of videodiscs are becoming available that contain slides and film clips in life science, physical science, art (imagine, all of the paintings from the Louvre art gallery on a single disc), social studies (videodiscs such as *Martin Luther King* and the *In the Holy Land* offer new instructional possibilities), and many other areas.

The local educational price of the Pioneer 2200 videodisc player is $620 (at the time this is written). Other models such as the Pioneer 4200 and the Sony 1200 are similar but more expensive. The Pioneer 2200 is especially attractive because of its low price, and because it includes both a remote control and a bar code reader.

Some consider this type of peripheral too expensive for a school–but it is about the same price as a disk drive used to be only a few years ago, and it has enormous instructional potential. Virginia has just adopted a technology plan for its schools that calls for a minimum foundation level of one videodisc player in every elementary and secondary library by 1994, as well as a minimum of one player for every eight classrooms.

It also might be thought that use of this type of peripheral with Logo is exotic or "futuristic," but that's not the case at all. Optical Data has even developed a program that allows videodiscs to be controlled by *AppleWorks*, and in that context it is difficult to argue that use with Logo is impractical or visionary. The videodisc player is connected to the serial port, and then simple commands control the videodisc player. For example, sending the two letter command "PL" to the serial port causes the Pioneer 2200 to begin playing.

6. Buy a Modem (cost: about $100)

We have listed these suggestions in order of increasing cost and practicality. Although a modem (about $100) is less than a videodisc player (about $600), we have put it last, because it is the most "experimental" of these suggestions. The computer is connected to the modem, and the modem, in turn, is connected to the phone line. A telecommunications program is then used to access a commercial service such as CompuServe, or an academic network such as the Internet or FrEdMail, or a local computer bulletin board. This is not a difficult task, and in a moment this *Logo Exchange* column will be sent to the editor, Sharon Yoder, through such a network. Although the process is not difficult, it is usually best to have a friend to lend a helping hand the first time you try this.

Since this is so, we have placed this suggestion at the end of our list. However, the combination of Logo and telecommunications is potentially very powerful. It can permit classes to do "networked science," in which several classes from across the nation or around the world participate in a common experiment. This experiment might consist of mapping patterns of acid rain on the National Geographic Kid's Network, or charting radon via the TERC Star Schools network.

In one exciting activity, students using LEGO TC logo at Murray High School in Virginia used telecommunications to interact with a parallel class at School 57 in Moscow last year. Logo programs can be exchanged via telecommunications. There are also discussions of Logo on the CompuServe Logo forum. At present, telecommunications is a bit like ham radio; the rewards are great, opening connections to a larger world, but a minimal level of technical knowledge is required at first. Then again, Logo itself was like that in its early years.

Summary

We have mentioned a half dozen possibilities to get you started for the new year. Some of them, like a paint program, are simple to begin using. Other suggestions, such as combining Logo and telecommunications, require a more extensive support structure for first time users. We could have mentioned other possibilities, such as the Apple Overlay Card which makes it possible to display Logo graphics and video output on a single screen.

All of these possibilities extend our vision of an evolving Logo. Over the next year we hope to discuss these Logo extensions and a few others at greater length. We hope you're looking forward to the new year as much as we are. We'll stop now so you'll have time to write that request for a new piece of software or hardware to enhance Logo in your classroom. Feel free to use us as a reference.

Glen Bull is a faculty member in the Instructional Technology program of the Curry School of Education at the University of Virginia. He is currently serving as president of the Virginia Educational Computing Association, and is a member of the Virginia Department of Education planning commission for educational technology. Gina Bull is a system administrator for the University of Virginia Department of Computer Science. By day she works with UNIX; by night, with Logo.

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 I: Making Recursion Visible

by Mark Horney

Introduction

This new column will feature the work of advanced Logo programmers of all ages. It will provide an opportunity to show what happens when Logo is pushed beyond the ordinary and beyond limits imposed by those who think Logo is just a toy for kids with little importance in the real world. The work described here will demonstrate the important role Logo has to play in programming, in problem solving, and in all parts of the learning environment.

If you are beginner, you may not understand everything that you see in this column, but it will provide an opportunity to expand your horizons. You can be the "student" when possible classroom lessons are discussed in this column. If you are an intermediate level Logo programmer, then this column should give you opportunities to explore some new ideas. If you are an advanced programmer, we hope you will share your work with us. Send your contributions to me at the address given at the end of this article. If you are a teacher of more advanced Logo students, then this column may provide you with ideas to use with your classes.

Some examples

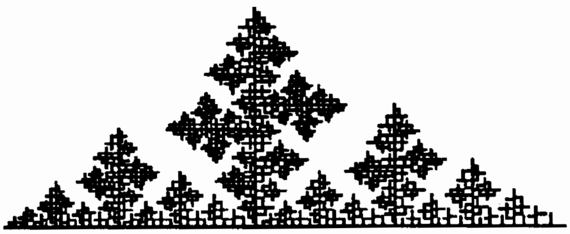
"Fractals I: Making Recursion Visible" is the first of a three part series about fractals, their usefulness in teaching recursion and their application, through Logo, into other parts of the curriculum.

Recursion is the closest thing to magic I know. A few words, a wave of the cursor, and *poof*, a world of chimerical creations bursts onto the screen. I did not know of recursion when I first came to Logo, I was just an old BASIC hacker. But, once I saw the magic, I became enchanted and never returned to plain vanilla programming.

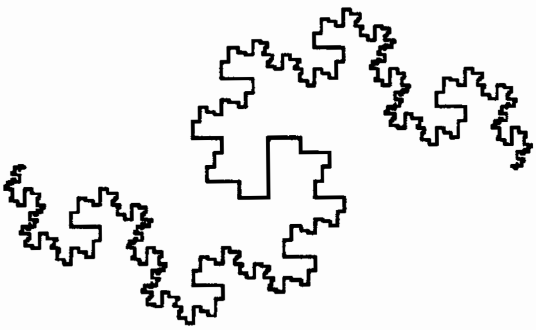
Recursion extracts a tariff from us however. Recursive programs are easy to write, but often monsters to debug. So, when teaching recursion it is worthwhile to seek tasks for our students which make the flow of procedures as visible and graphic as possible. An area affording this is the creation of fractals. Fractals are odd mathematical creatures that were first described around the turn of the century and have become increasingly notorious in the past ten years. The figure on the next page shows several examples.

Here is the code used to generate these figures.

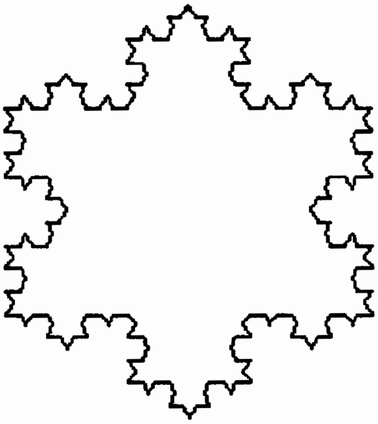
```
TO SPIKE :SIDE :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
SPIKE :SIDE * 0.45 :LEVEL - 1
LEFT 90
SPIKE :SIDE * 0.4 :LEVEL - 1
RIGHT 180
SPIKE :SIDE * 0.4 :LEVEL - 1
LEFT 90
SPIKE :SIDE * 0.55 :LEVEL - 1
END

TO DRAGONSKIN :SIDE :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
RIGHT 45
DRAGONSKIN :SIDE / ( 2 * SQRT 2 ) :LEVEL
   - 1
LEFT 90
DRAGONSKIN :SIDE / SQRT 2 :LEVEL - 1
RIGHT 90
DRAGONSKIN :SIDE / ( 2 * SQRT 2 ) :LEVEL
   - 1
LEFT 45
END

TO SNOWFLAKE :SIDE :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
SNOWFLAKE :SIDE / 3 :LEVEL - 1
LEFT 60
SNOWFLAKE :SIDE / 3 :LEVEL - 1
RIGHT 120
SNOWFLAKE :SIDE / 3 :LEVEL - 1
LEFT 60
SNOWFLAKE :SIDE / 3 :LEVEL - 1
END

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

TO RIVERTREE:SIDE :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
LEFT ARCTAN (1/2)
RIVERTREE:SIDE / SQRT 5 :LEVEL - 1
```



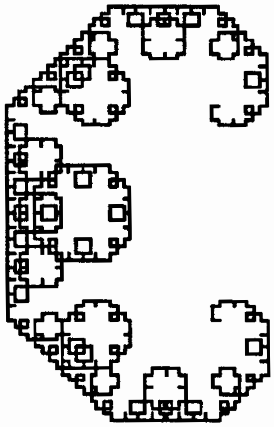
Spike



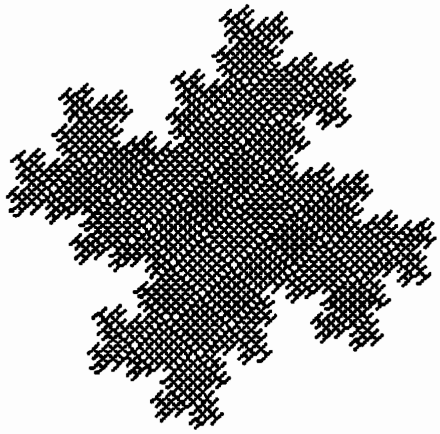
DragonSkin



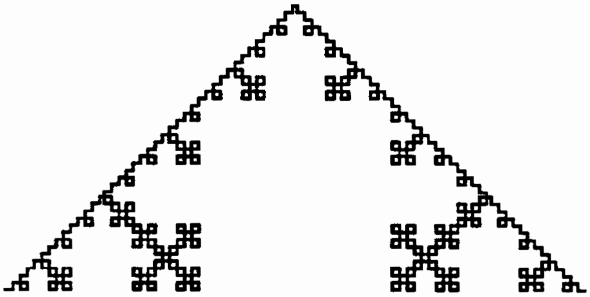
Snowflake



"C"



RiverTree



SquareFlake

```
RIVERTREE :SIDE / SQRT 5 :LEVEL - 1
PU
BACK :SIDE / SQRT 5 PD
PD
RIGHT 90
RIVERTREE :SIDE / SQRT 5 :LEVEL - 1
LEFT 90
PU
BACK :SIDE / SQRT 5
PD
RIVERTREE :SIDE / SQRT 5 :LEVEL - 1
RIVERTREE :SIDE / SQRT 5 :LEVEL - 1
RIGHT ARCTAN (1/2)
END
```

```
TO SQUAREFLAKE :SIDE :LEVEL
IF :LEVEL = 0 [FORWARD :SIDE STOP]
SQUAREFLAKE :SIDE / 3 :LEVEL - 1
LEFT 90
SQUAREFLAKE :SIDE / 3 :LEVEL - 1
RIGHT 90
SQUAREFLAKE :SIDE / 3 :LEVEL - 1
RIGHT 90
SQUAREFLAKE :SIDE / 3 :LEVEL - 1
LEFT 90
SQUAREFLAKE :SIDE / 3 :LEVEL - 1
END
```

Because of their naturally recursive structures, and their intricate, quirky appearance, fractals make a good environment for exploring recursive procedures. Fractals also provide a rich environment for mathematical problem solving extending over a wide range of student programming and mathematical ability. This is of critical importance. It is not enough for students to simply draw a bunch of fractals. We teach fractals so students can learn about recursion; we teach recursion so they can learn other things. For fractal lessons to be complete, students must not only learn about Logo, they must learn with Logo.

Introducing Fractals

The easiest way to introduce students to fractals is to give them several fractal procedures and have them experiment with various inputs. This introduction should probably not involve any discussion of recursion, or of fractal characteristics; just have them muck about for a bit. After a suitable interval, perhaps a period or two, give the students a set of questions about what they see happening. Here are some examples of what might be asked:

1. Estimate the length of high level fractals (those that take too long to actually draw) and present an argument to support the estimate. In the same way, estimate the time it takes to draw fractals. From these estimates, calculate the velocity of the turtle.

2. Estimate the level needed to draw a fractal that would stretch around the world if straightened out. To the Moon? To Mars? Just how far is it to Mars?

3. Measure the length and width of a complete fractal. Are "higher" level fractals bigger than "lower" level fractals? Just how big can fractals get?

4. How do fractals with many sides differ from those with just a few sides?

5. As the turtle draws a fractal it often stops and turns repeatedly from side to side. Other times it will sit and spin around for several seconds. It looks as though the turtle can't make up its mind where to go next. What is causing the turtle to behave in this fashion? Is it right to talk about turtle "behavior?" After all, computers are machines, not people.

6. When Logo draws "straight" lines, they are often composed of short, jagged pieces. This happens because computer screens do not have enough pixels to position straight lines everywhere on the screen. This effect can be lessoned by drawing lines "evenly," that is, drawing where pixels naturally occur. This can be accomplished by adjusting the :SIDE variable. For a given fractal, what are better values for :SIDE?

7. Draw the procedure tree for a fractal procedure. How many nodes does it have?

Obviously there are few "right" answers to questions such as these. Their purpose is to initiate discussions and to lead students into active observation fractal procedures at work. The next step is to have the students modify the procedures. This will lead them to a deeper understanding of fractals and recursion. Here are some ideas for procedure modifications:

1. Insert code into procedures to sum side lengths and total turning.

2. Modify a procedure to count the number of procedure calls.

3. Note how some fractal procedures use division to reduce the size of the generator and others use multiplication.

How can this be? Rewrite the division procedures to use multiplication and vice versa.

4. Change a fractal procedure so that instead of drawing the fractal, it prints out a list of the commands given to the turtle. How to the lists for high level fractals differ from those for low level fractals?

This introduction to fractal procedures will take 3 to 5 periods. This time can be increased or diminished by adjusting the number and difficulty of the activities students are given.

Constructing New Fractals

No doubt students will want to make fractals of their own. Bringing them to this desire is the purpose of the introductory activities. In making their own fractals students receive an intense experience with recursion.

There are many different types of fractals and a legion of different recipes for creating them. The examples shown here trace their origin to Helge von Koch, who in 1906 first described what is now generally known as the Snowflake.

The technique for drawing von Koch fractals is simple:

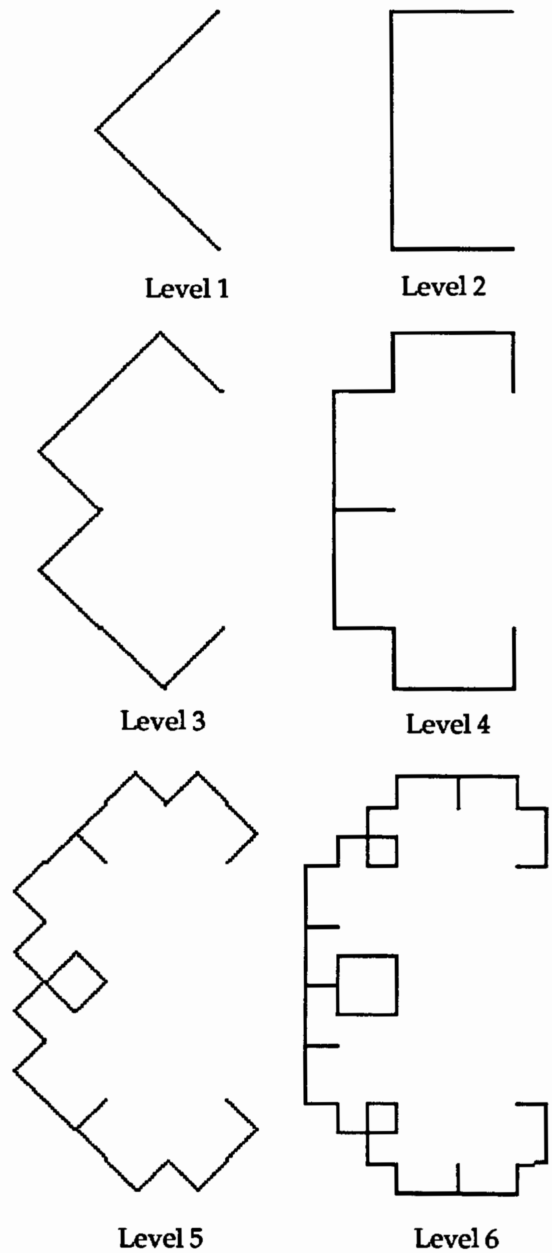
Step 1: Start with a shape, usually a line segment, called the "initiator." (In Figure 1, the Snowflake uses a triangular initiator, the others use the unit line segment)

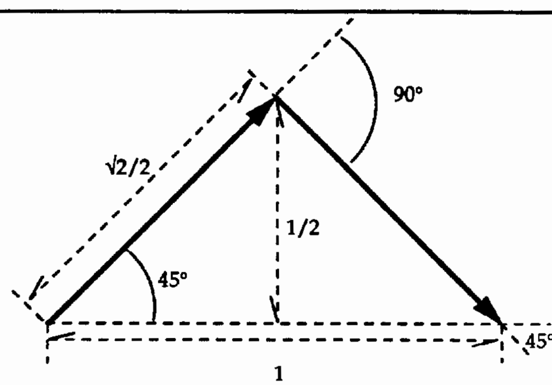
Step 2: Replace each piece of the initiator with another series of shorter line segments, called the "generator."

Step 3: Treat each segment of the generator as if it were an initiator and replace them with reduced copies of the generator. Repeat this process for as many "Levels" as you have patience, and computing power.

One of the simplest fractals is the "C" curve invented by Donald Knuth. The figure at the top of the next column shows the first six "Levels" of the C curve. The generator for the C curve is at the bottom of the column. Mathematicians call this a "hat" function.

The following code shows how the procedure for drawing a hat function is converted into the recursive procedure that draws the C fractal:





```
TO GENERATE  :SIDE
LEFT 45
FORWARD  :SIDE
RIGHT 90
FORWARD :SIDE
LEFT 45
END
```

```
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
```

In general, such conversions are done in two steps. The first is to insert a stopping condition. This command (always the first line of these fractal procedures) halts the recursion when :LEVEL reaches 0. Note that level 0 of a fractal is the initiator. Thus, the stopping condition also includes the FORWARD statement that draws the fractal.

The second step in the conversion process is to replace each FORWARD command of the generator procedure with a recursive call to the same procedure. The two parameters, :SIDE, controlling the size of the fractal, and :LEVEL controlling the depth of the recursion are both reduced in value. The reduction of :SIDE is particularly important. The generator must be diminished by the exact amount so it fits perfectly into just one segment.

In this way, a procedure to draw any generator can be converted into a recursive procedure for drawing a fractal. Generators can be composed out of any number of line segments so long as the length of each segment is shorter than the distance from the generator's beginning and ending points. The generators for the fractals shown at the beginning of the column can be drawn by using level 1 in the procedures shown below the figures.

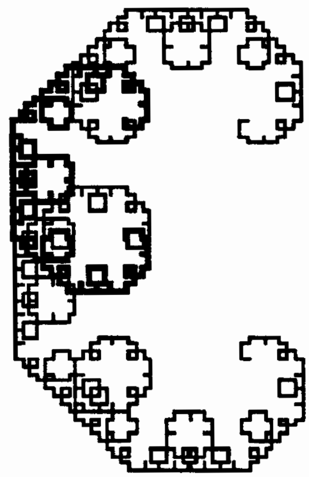
There are different ways for students to create new generators: using Geo Boards, rulers and protractors, Cartesian and polar coordinates, and with vectors. Students should be taught a technique appropriate to their mathematical sophistication and their need for practice with the various tools and concepts required. This mathematical work can be extended by asking students to "prove" that generators meet the criterion listed above. These proofs can involve plane geometry (particularly the Pythagorean theorem), vector addition, and trigonometry.

Once they've made their own fractals, students can repeat the introductory activities list above. They can also be ask to do "reverse engineering." That is, start with a fractal and find its generator. These activities can continue for five to ten periods initially and can be repeated as students advance in mathematics.

More Advanced Topics

While the recursive nature of fractals is very visible, some students will need extra help to "see" what's going on. One way to accomplish this is to have students place "markers" in their procedures that are activated at particular times and places. A simple example of this is to insert a command at the beginning of procedures to play musical notes of a pitch determined by :LEVEL. With this, students can hear a program moving up, down, and across the procedure tree.

Marking procedures can be achieved in a variety of fashions.



Marked " C "

This C curve has been marked so that certain low level procedures draw with wider lines. This demonstrates that "high" level fractals are built up out of "low" level fractals.

The procedure for drawing this "Marked C" is shown below. The procedure works via the addition of three new variables: the global variable :COUNT, which labels each instantiation of the C procedure with a unique number, :MARKER, which designates whether or not a particular procedure is marked or not, and :TRIGGER, which determines which procedures are to be marked. In the figure above, the 513th call of C is marked with wider lines.

```
TO MARKED_C :SIDE :LEVEL :MARKER
      :TRIGGER
IF :LEVEL = 0 [IFELSE :MARKER = "ON
   [SETPWIDTH 2 FORWARD :SIDE STOP]
   [SETPWIDTH 1 FORWARD :SIDE STOP]]
MAKE "COUNT :COUNT + 1
LEFT 45
IFELSE OR ( :MARKER = "ON ) ( :COUNT =
   :TRIGGER )
        [MARKED_C  :SIDE  / SQRT 2
         :LEVEL - 1 "ON :TRIGGER]
         [MARKED_C  :SIDE / SQRT 2
         :LEVEL - 1 "OFF :TRIGGER]
RIGHT 90
IFELSE OR ( :MARKER = "ON ) ( :COUNT =
   :TRIGGER )
        [MARKED_C  :SIDE  / SQRT 2
         :LEVEL - 1 "ON :TRIGGER]
         [MARKED_C  :SIDE  / SQRT 2
         :LEVEL - 1 "OFF :TRIGGER]
LEFT 45
END
```

Fractals can be marked in many ways: changing line width; or with sound (as above); or by changing line colors, line patterns,or turtle shapes, or by printing messages. Different techniques will be appropriate to different students and versions of Logo. These markers can be used to show fractal parts of a particular kind or of a particular level. They can be used in combinations. Students should be prompted to invent their own techniques and to employ them for a variety of purposes.

Here are some other ideas for advanced projects:

1. Write procedures that make it appear as though several turtles are drawing the fractal in parallel.

2. Derive a formula to calculate the length of a fractal. Derive a formula to calculate the number of procedure calls a fractal makes.

3. Estimate the length and width of a box that a fractal would fit entirely within. Present an argument for the correctness of this estimate.

4. Use an infinite series to prove that an infinite level fractal is infinitely long.

5. Examine the similarities between marking fractals and the chemical and radioactive markers used in Biology and Medicine.

6. How could fractals be used to simulate Brownian Motion?

7. Draw fractals in other programming languages. What are the advantages and disadvantages of each? How are recursive procedures written in non-recursive languages such as BASIC?

8. Draw 3-dimensional fractals.

9. Noneof the diagrams shown here are of actual fractals. Real fractals are "limit functions." That is, they are the limiting case of an infinite series of "levels." Any picture of a fractal is only an estimate of its true appearance. How could an algebraic definition of the form "f(x) = ..." be written for these functions?

Challenge

The modern day interpreter of fractals is Benoit Mandelbrot, who's book *The Fractal Geometry of Nature*, displays many wonderful inhabitants of the fractal bestiary. The figure on the next page shows one of them, the Polya Triangular Sweep. Mandelbrot's directions for constructing this fractal are: "The initiator is [0,1], the generator is [the hat function], and it alternates between the right and the left of the teragon [Mandelbrot's word for these fractals]. The first position also alternates." (p. 64). From these clues can you draw this fractal? The answer, along with more information about constructing fractals will be included in a later article.

There is a wide variety of books and articles about fractals. The list below gives details of the items mentioned above and others as well:

Abelson, H. & diSessa A. (1980). *Turtle geometry*. Cambridge, Massachusetts: MIT Press.

This book was the source of information about the C fractal (p. 93) and is an excellent introduction into the mathematics of the Logo. Other topics include vector methods, turtle topology, 3-dimensional turtle geometry, and relativity.

Mandelbrot, B.B. (1983). *The fractal geometry of nature*. New York: W. H. Freeman and Company.

Mandelbrot's book contains a myriad of pictures and diagrams showing a huge collection of fractals. For many he gives directions (albeit somewhat terse) for their construction. The SquareFlake (p. 139), the DragonSkin (p. 67), the RiverTree (p. 73), and the Polya Sweep (p. 64) fractals where drawn from his patterns. Mandelbrot also discusses the mathematics of fractals, although at a very high level.

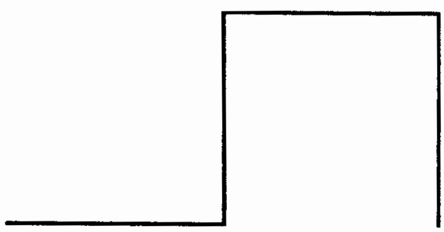
Peitgen H. O. & Richter P. H. (1986). *The beauty of fractals*. Berlin: Springer-Verlag.

Peitgen and Richter have done the most to display fractals in amazing color and detail. This book contains few references to the von Koch fractals discussed in this paper, but it does give a good introduction to other types of fractals and is much more approachable than Mandelbrot.

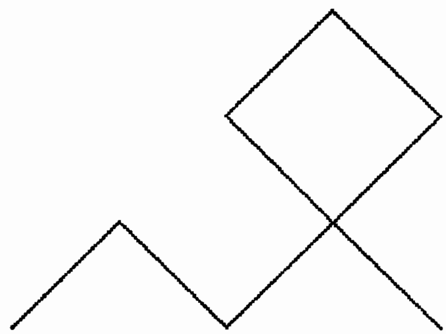
von Koch, H. (1906). Une methode geometrique elementaire pour l'etude de certaines questions de la theorie des courbes planes. *Acta Mathematica*, 30 145-174.
I include this reference to von Koch simply to give him credit for his early work with fractals although you might want to take a look at this paper if you read French and want to see a real mathematician at work.

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

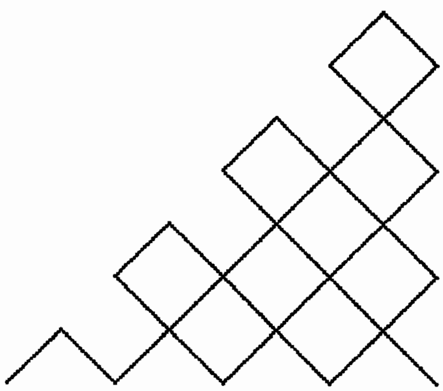
Mark Horney
1787 Agate Street
Eugene, Oregon 97403
BITNET: MHORNEY@OREGON



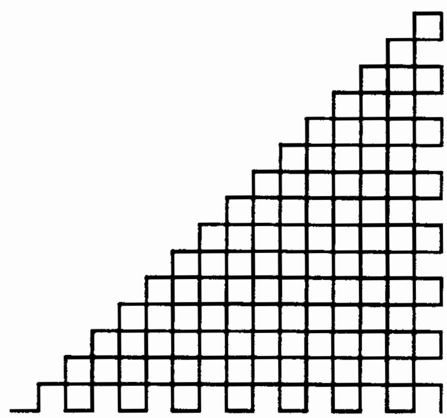
Polya Sweep Level 2



Polya Sweep Level 3



Polya Sweep Level 5



Polya Sweep Level 8

MathWorlds

Confluence: Logo, Educational Technology, and Mathematics–Further Reflections

edited by A. J. (Sandy) Dawson

Now where was I before I was interrupted by the summer? Oh yes, I was talking about the philosophy one holds regarding the nature of mathematics and how this might or might not be compatible with Logo pedagogy. I was saying that in my view Logo is constructivist in orientation, which means that

1. even if there is a reality external to the human mind, the meaning of that reality still must be created by each individual (this is the weak constructivist position), and
2. mathematics is created, not discovered, and
3. computer technology is not neutral and value-free, and
4. language is culturally biased (the only meaning which words have, as Alice said, is that which we create for them!)

The confluence of these four factors are the basis of the Logo philosophy, in my view, and only when these streams merge will we achieve a river of thought and action which has breath, depth, and power in an educational sense.

To adopt this stance necessitates looking at the world and our knowledge of it in different ways. But if Logo and its many cousins (see Sharon Yoder's editorial in the March 1990 issue of *Logo Exchange*) are to be embraced by the educational community, then those of us who are already of the constructivist perspective will have to assist others in coming to see the world in different ways. To that end let me expand on each of the four items.

1. Reality

Personally, I find it difficult to deny there is an external reality. The computer desk at which I set and the chair in which I recline as I reflect on this column have an existence beyond my mind conceiving of them. The Mac portable I am using right now to prepare this column will, I believe, exist even if I expire during the night. There is in my view an external physical reality.

However, the meaning I attach to that reality, the manner in which I use (or abuse) that reality, is a matter of my own creation, a matter of my mind teaching my brain about the world my mind perceives (Gattegno). My mind is my meaning maker, creating sense of the external world as it impinges on me through my senses. It is truly a miracle that communication takes place between any two people as they create their own meaning about that external reality.

2. Math is created

Pirie and Kieren have recently enunciated a theory of mathematical understanding that I would argue paints the broad landscape of the manner in which learners create their own meaning about mathematics. They call their theory transcendent recursion, by which they mean that ...

> new constructs transcend but are compatible with old ones (they are not simple extensions). What is in fact more critical is that the 'new' transcendent knowing frees one from the actions of the prior knowing. (Pirie and Kieren, p. 8)

Pirie and Kieren's first level is "doing" from which flows "image making." As described, this seems very close the exploratory activity learners experience when first introduced to Logo. It is a necessary but not necessarily a sufficient condition for later deeper learnings to manifest themselves. They go on to say that

> at the next level these action-tied images are replaced by a form for the images. From the mathematical point of view it is this "image having" which frees a person's mathematics from the need to take particular actions as examples. This is a first level of abstraction; but it is critical to note that it is the learner who makes this abstraction by recursively building on images based in action. For understanding to grow, these images cannot be imposed from outside. (Pirie and Kieren, p. 8)

Learners examine these images, noticing similarities and differences, comparing and contrasting them one with the other, and seeing (or not) connections among them. This "property noticing" level is at a level of unselfconscious knowing according to Pirie and Kieren, and is a precursor to formal and conscious consideration when mathematical abstractions are more fully developed.

> One is now in a position to observe one's own thought structures and organize them consistently. One is aware of being aware, and can see the consequences of one's thoughts. (Pirie and Kieren, p. 9)

Clearly, Pirie and Kieren are of the view, which I share, that learners must construct their own mathematical realities. This doesn't imply that there isn't a world "out there" which may provide sensory experiences as input to the process of meaning making, but it does imply very strongly that only the learner can make that meaning.

3. Computers are non-neutral

Recall that one of Bowers' central points is that computer technology ..

> ...cuts out of the communication process ... tacit heuristic forms of knowledge that underlie common-sense experience. While the technology amplifies the sense of objectivity, it reduces the awareness that the data represent an interpretation influenced by the conceptual categories and perspective of the person who "collected" the data or information.

What needs to be noted here is that all forms of instructional technology, including the printed page, amplify certain things and reduce the importance of others. When seen in conjunction with point (2) above, it is clear that what is needed is the awareness and recognition that all learners and all assists to learning—such as computers—carry with them built-in biases. Of course, saying that this is so is not enough. One needs to then act on this knowledge so that these biases can be examined, critically scrutinized, and modified where that seems appropriate. We all have a biased view of that external reality. It cannot be otherwise. What we do not want to have is a prejudice about that reality, because prejudices are pre-judgements which are not then subjected to careful analysis.

So, yes, computer technology is not neutral and value-free, and we must guard against claims that it is so, just as we have to be wary of all claims of value-free neutrality. In particular, the statements that computers give completely objective analyses because the machines only deal with 'hard' data must be challenged at every opportunity.

The Logo programs which students create, the designs they draw on the computer screen, are "their" designs, and are subject to not only the restrictions of the programming language being used–in this case Logo–and the processing capabilities of the machine, but also to the flexibility, inventiveness, and degree of awareness of the mind of the learners who are doing the creating.

I think it was Bertrand Russell who said that mathematics is the subject in which we know not what we were talking about, nor if what we say is true. I personally find it difficult to see mathematics as part of some reality external to the human mind. Though mathematical ideas and concepts might be derived from interactions with physical reality, I cannot envisage how those ideas and concepts could exist in the absence of any human mind.

4. Language is culturally biased

Mathematics is also the subject in which the same name, the same word, is used to stand for many different things, and the same thing can have many different names. In the former case, for example, the word "ten" in base seven is not the same thing as "ten" in base five. In the latter instance, one half can also have the name two-fourths, three-sixths, four-eighths, and so on. Further, the meaning that different individuals have for the same word, the images that a single word can evoke for different individuals, is not unidimensional. A brief discussion among 12 mathematicians and mathematics educators of which I was a part a number of years ago, generated no less than 26 different images and meanings for the term "limit."

Language cannot be a simple conduit to carry without contamination the meaning attached by the sender of the message. The message sent is not necessarily the message received, the reason being that the sender and the receiver create their own meanings based on their own life experiences. As mentioned earlier, it is truly a miracle that communication does occur, what with everyone creating and sending their own meanings, which are then received and interpreted by others who have their own belief systems through which to sift the incoming data. That data is not pure. It is not free from bias. This bias is created by the sender and the receiver.

What does this all imply about the teaching and learning of mathematics and the use of Logo, both as an object of study, and as a tool for learning. At the very least, I would argue that we must as teachers deal with the meanings and images that the learners can elucidate, describe, and draw. It is their meanings we must work with, even if our goal is to try to have them share some of the collective meanings our society has about mathematics. If Pirie and Kieren are on a productive track, we must also recognize how learners recursively develop their images and meanings, and that this may mean that learners return to earlier levels but with new understanding. It means that all involved in the educational environment must work towards the identification of their own biases, not because biases in and of themselves are bad, but because they represent just one way of viewing the world, and other ways might be more useful, more powerful, more predictive, and more comprehensive than other ways.

However, it must also be noted that these other more useful, powerful, predictive, and comprehensive ways cannot be "put into" the learner. They must grow from the inside of the learner. That is the educator's challenge. And the educator who succeeds in this task most certainly has a wonderful gift.

NECC '91

Call for Participation

National Educational Computing Conference
June 17-20, 1991
Phoenix, Arizona

Those who would like to participate in NECC '91 are invited to submit original papers, and/or proposals for project reports or preconference workshops. Individuals representing all academic disciplines and all facets of education are welcome to submit.

To obtain a complete *Call for Participation* brochure (which describes the format in which proposals must be submitted), contact:

NECC '91, ISTE
University of Oregon
1787 Agate St.
Eugene, OR 97403-9905

Ph. 503/346-4414
FAX 503/346-5890

Submission deadline for all Papers, Project Proposals, and Preconference Workshop Proposals is October 15, 1990.

References

Bowers, C. A. (1988) *The cultural dimensions of educational computing*. New York: Teachers College Press.

Gattegno, C. (1974) *The mind teaches the brain*. New York: Educational Solutions Inc.

Pirie, S., and Kieren, T. (November, 1989). A recursive theory of mathematical understanding. *For The Learning of Mathematics*, *9* (3), 7-11.

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

Logo or "Logo-like": The Great Debate
by Douglas H. Clements

Do research results have ramifications for the recent debate about Logo vs. "Logo-like" computer environments? The richness of the educational issues virtually guarantees that research will not help answer every question. There are, however, a small group of relevant findings.

Computer Languages: Superiority or Equivalence?

Several researchers have compared programming in Logo to programming in other computer languages. One clear finding is that well-designed languages are easier to learn. Most comparisons of the effects of such programming, however, show few differences. For example, several studies have found that Logo and BASIC equally increased students problem-solving ability, self-esteem, and internal locus of control (Brown & Rood, 1984; Dvarskas, 1984; Shaw, 1985). There may nevertheless be some differences. In one comparison, students programming in BASIC performed better on a pre-algebra mathematics concepts test, whereas those programming in Logo initiated more program expansion activities and interacted more with the teacher (MacGregor, 1988).

Research has also addressed whether initial learning of Logo supports later learning of other languages. Sixth-grade students were able to transfer their experiences from the Logo environment to the Pascal environment (Sweetland, 1986). They continued to use top–down and procedural thinking. Interestingly, they frequently chose to extend programs they had written in Logo within the Pascal environment. Similar effects have not been found when Logo is followed by BASIC (Calamari, 1984).

Despite minor differences, then, the language chosen for these types of educational interventions does not appear to make a substantive effect.

Chauvinism or Coherence?

One difficulty with such studies is that they tend to compare the effects of programming in different languages over short periods of time (e.g., two weeks). What about the long term? Doesn't it make sense to provide students with a variety of languages so they can compare and contrast for themselves, and a variety of programs for performing different tasks? It depends on your purpose. To teach students programming per se, such an approach may be helpful (although pre-college research is mixed on this issue). Developing meaningful "computer literacy" via familiarity with a number of computer utilities might also be a reasonable goal. However, the reality of time constraints—especially in the elementary and middle school—may militate against such an approach. Teachers report having insufficient time to teach even one or two powerful tools. And what students need is a *thinking* tool, not just a computer tool.

We know that after students really learn to use such a tool, they can perform impressive feats of creativity and problem solving. The computer program has become a tool for thinking. This implies that to engage students in substantive intellectual enterprises, we need to develop their mastery of a flexible, extensible tool. Extensions to this tool are welcome. But the tool must be meaningfully extensible within the realities of classroom life. Can different programs in the same genre, such as different computer languages, serve as extensions? Ideally, perhaps, but a concern is that this may lead to a potpourri of applications with no internal coherence.

To document the validity of this concern, consider a research study my colleagues and I conducted. We compared two computer environments: Logo on one hand, and a set of utility and problem-solving programs on the other. Logo students developed significantly more in the area of self-esteem and intrinsic motivation. They showed more pleasure at the discovery of new information (Nastasi, Clements, & Battista, 1990). We postulated that only Logo provided a consistent medium for the expression of ideas. Repeated and self-directed experience led to feelings of control and mastery of this medium–that is, of a single, coherent intellectual tool. The other environment provided diversity in theme, but also a less consistent and thus more restrictive medium of expression.*

Fanaticism or Enthusiasm?

An unfortunate side effect of the language chauvinism debate is the rebirth of the "Logo religion" criticism. Such admonishments are old and were just as counterproductive and misleading then as now. What is enthusiasm for me is termed fanaticism by others (especially if they are advocating something different).

---

* Coherence should also be considered with regard to an educational community. I am reminded of an influential Logo organization that "broadened" their scope to include virtually everything interesting…and disbanded. If we attempt to be everything for everybody, are we doomed to be nothing for nobody?

Fanaticism is uncritical devotion. Enthusiasm is ardent engagement. We need students who are mathematics (or reading or computing) enthusiasts. We are in dire need of more adults who are education enthusiasts. Both groups should be appropriately critical.

This brings us to a final point related to research. Criticism must be informed criticism. Reflection is essential. There is a real danger that we will leave each tool behind long before we know enough about how to use it. Certainly, this works against building a research base adequate to inform educational practice. The human tendency to "play with new toys" often prevents us from finding out—with both classroom-based and more formal types of research—how to use any computer tool well. It relegates each to the role of fad.

We know quite a bit about using Logo. We should seriously consider committing ourselves to the difficult but rewarding work of building on this knowledge.

Finally, more important than any computer tool is a community of people who care about students' self-esteem and learning. I've always believed this to typify the Logo community. Debate is necessary—one sign of a vital field. But "us vs. them" thinking is divisive and destructive. The challenge is to balance enthusiasm and open-mindedness, coherence and innovation.

References

Brown, S. W., & Rood, M. K. (1984, April). *Training gifted students in Logo and BASIC: What is the difference?* Paper presented at the meeting of the American Educational Research Association, New Orleans, LA.

Calamari, M. D. B. (1984). A comparison of two methods of teaching computer programming to secondary mathematics students. *Dissertation Abstracts International, 45*, 67A. (University Microfilms No. DA8409574)

Dvarskas, D. P. (1984). The effects of introductory computer programming lessons on the learners' ability to analyze mathematical word problems. *Dissertation Abstracts International, 44*, 2665A. (University Microfilms No. DA8400949)

MacGregor, S. K. (1988, April). *A study of the relationship between Logo and BASIC programming, cognitive style, and mathematics achievement.* Paper presented at the meeting of the American Educational Research Association, New Orleans.

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.

Shaw, D. G. (1985). The effects of learning to program a computer in BASIC or Logo on the problem solving abilities of fifth grade students. *Dissertation Abstracts International, 45*, 1985A. (University Microfilms No. DA8423897)

Sweetland, R. D. (1986). Learning Pascal in a sixth grade classroom: After Logo experiences. *Dissertation Abstracts International, 47*, 2442A.

Douglas H. Clements is associate professor at State University of New York at Buffalo, Department of Learning and Instruction, 593 Baldy Hall, Buffalo, NY 14260. He has conducted numerous research studies on the benefits of Logo environments for children's development of creative, metacognitive, problem-solving, and social abilities. Through a National Science Foundation grant, has co-developed a K-6 elementary geometry curriculum, *Logo Geometry* (published by Silver, Burdett, & Ginn). His most recent book, *Computers in Elementary Mathematics Education*, was published by Prentice-Hall in September, 1988.

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 Exchange Continental Editors

Africa	Asia	Australia	Europe	Latin America
Fatimata Seye Sylla	Marie Tada	Jeff Richardson	Harry Pinxteren	Jose Valente
UNESCO/BREDA	St. Mary's Int. School	School of Education	Logo Centrum Nederland	NIED
BP 3311, Dakar	6-19, Seta 1-chome	GIAE	P.O. Box 1408	UNICAMP
Senegal, West Africa	Setagaya-ku	Switchback Road	BK Nijmegen 6501	13082 Campinas
	Tokyo 158, Japan	Churchill 3842	Netherlands	Sao Paulo, Brazil
		Australia		

This issue marks the beginning of the ninth year for the *Logo Exchange* and the seventh for this column. Having attended the NECC in Nashville and the WCCE in New Zealand and Australia this past summer, I am convinced that the use of Logo in the world's classrooms is on the rise. Numerous articles were solicited for this column, so *LX* readers can look forward to reading about how Logo is being used all over the world.

During September of the 1989/90 school year, Hurricane Hugo hit the Virgin Islands knocking out electricity, phone, and cable TV lines for months. After spending some time cleaning up the mess, we went to California went to California for a few weeks' break; arriving just in time for a few weeks' break, arriving just in time for the earthquake. After that we started getting calls from relatives begging us not to come near their areas. At any rate, I hope that this school year will see all the moving and shaking occurring in Logo and not in Mother Nature.

From Brazil

Our first report of the new school year is from our Latin American correspondent, José Valente of Brazil. He reports on five Logo projects taking place in Rio Grande do Sul.

As was mentioned in the September 1988 issue of the *Logo Exchange*, one of the centers of the EDUCOM project in Brazil is located in the southern city of Porto Alegre in the state of Rio Grande do Sul. One of the branches of this center is attached to the School of Education of the Federal University of Rio Grande do Sul, and one of its objectives is to study cognitive processes and affective aspects of the use of Logo.

The research projects developed or in development are:

- The effects of using Logo on the intellectual behavior of first grade students who have been retained in first grade. This study was initiated in 1986/87 in two schools, involving two classes of 18 students who had not reached a minimum level of literacy. Their cognitive performance was also evaluated with respect to conservation, logical thinking, and spatial concepts. After weekly interactions with Logo, these students showed a remarkable improvement in their self esteem and school performance, especially in the areas of reading and writing. Their cognitive performance also improved with respect to conservation of length and classification.

- A study using Logo with 12 educable mentally deficient children aged 8 to 14 was conducted during the 1986/87 school year. They participated in weekly Logo sessions and regular classroom activities. According to biomedical, psycho-pedagogical evaluations, and their given grade level in the special classrooms, it was possible to confirm that Logo is a viable tool to work with these populations. During 1986, 50% of the group went to a regular second grade classroom, and in 1987, only one of the children stayed in the special classroom.

- Construction of mathematical concepts through the use of Logo was the thrust of a study consisting of observing 13 students from a first grade public school classroom. Comprehension of numbers, and partial and temporal concepts were investigated. The students developed spontaneous Logo activities and played with certain games specially developed to study and overcome conceptual difficulties they presented. The study showed that the Logo activities associated with open-ended activities improved students' conceptual notion of laterality, representation of numbers, and logical aspects of order. It was possible to observe that some children did not comprehend addition and subtraction and reversibility. The study suggested pedagogical changes in the current process of teaching mathematics.

- Development of a methodology for talented children was the focus of a study involving a group of eight 9 to 13 yearolds from different public schools. All the students were identified as exceptionally talented. All the students worked with Logo, and some worked with other educational software as well. In addition, group sessions were conducted to investigate emotional and affective uses. It was possible to observe positive changes in interpersonal relationships, convivial relationships in competitive situations, leadership in school activities, school performance, interest and motivation for the activities, proposition of projects of greater exploratory complexity, and mastery of educational computer packages and their application in school activities. The activities also favored decision making, self-direction, and independence.

- Creation of stories by first grade students using Logo was the focal point of a study involving five 9 to 11 year-old students. The objective was to make it possible for these children to use literature to recreate their reality through graphic and textual expressions. They used the text editor to produce text and Logo to produce graphics. There were three different types of play situations. First, students produced stories through the use of semi-structured sentences. Second, the idea of personage was introduced, which motivated the process of story writing. Third, they constructed the personage and used Logo animation capabilities. This type of activity led to the creation of a story book. The analysis of the results is in progress, but it indicates a great deal of spontaneity and creativity in the writing and graphic expressions, as well as a better quality in extension and coherence of text production.

There are several other studies in progress: the use of Logo with trainable mentally deficient children; the study of the meaning of the Logo commands by deaf children, with the objective of interface development; the intellectual development of children with perceptual difficulties; and metacognitive experiences in the reading and writing processes in early first grade students.

For more information please contact:

Lucila Santarosa
Project EDUCOM
Faculdade de Educação - UFRS
Rua Paulo Gama information please contact:
90.049 - Porto Alegre - RS - Brazil

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

Non-Profit Organization
US Postage
PAID
Eugene, OR
Permit No. 63