

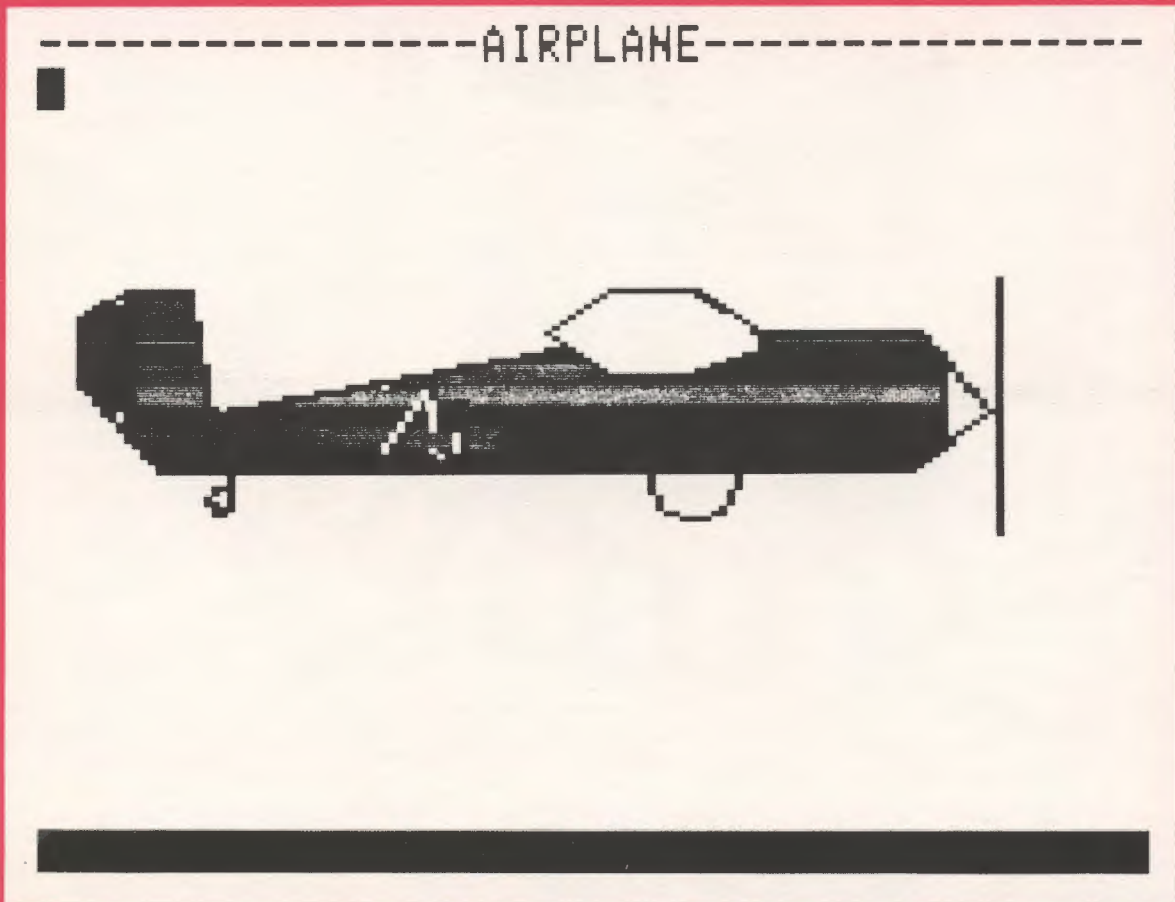
Journal of the ICCE Special Interest Group for Logo-Using Educators



LOGO EXCHANGE

OCTOBER 1987

VOLUME 6 NUMBER 2

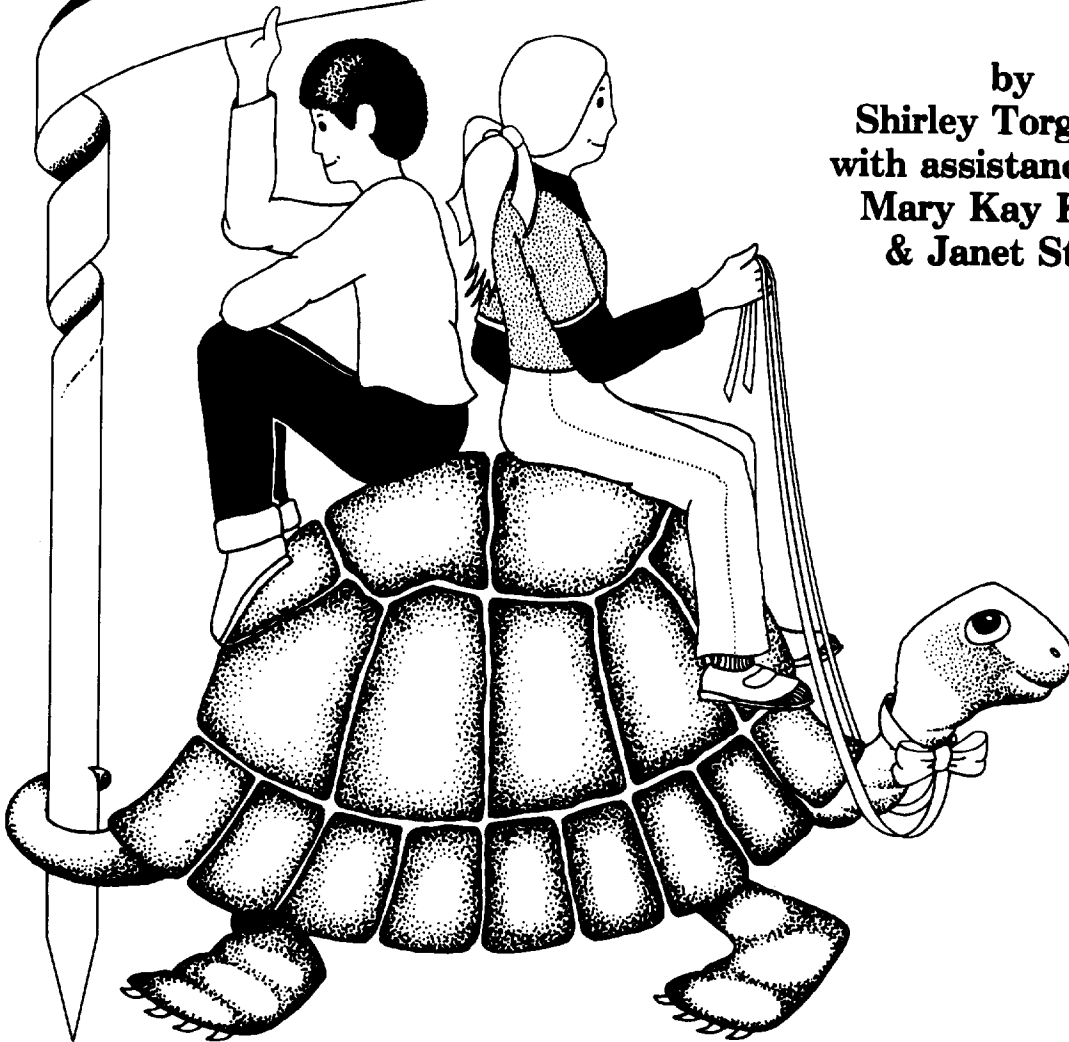


International Council for Computers in Education

Logo

in the Classroom

by
Shirley Torgerson
with assistance from
Mary Kay Kriley
& **Janet Stone**



Logo in the Classroom integrates Logo into your elementary curriculum. Twenty lessons were developed in a classroom setting as a response to "How can Logo work in a classroom where computers are in short supply?"

Detailed teacher information is given for each lesson along with copyable practice sheets, transparency masters and 12 charts. Useful for teacher inservice.

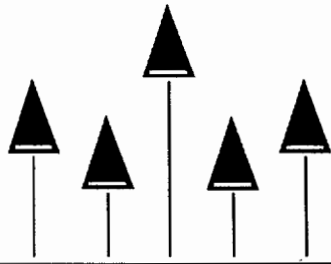
\$13.00 (US)



University of Oregon

1787 Agate St.

Eugene, OR 97403



LOGO EXCHANGE

VOLUME 6 NUMBER 2

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

OCTOBER 1987

Founding Editor

Tom Lough

Editor-In-Chief

Sharon Burrowes

International Editor

Dennis Harper

Field Editors

Eduardo Cavallo
Patricia Dowling
Anne McDougall
Richard Noss
Fatimata Seye Sylla
Hillel Weintraub

Contributing Editors

Gina Bull
Glen Bull
Doug Clements
Bill Craig
Sandy Dawson
Judi Harris
Robs Muir
Rebecca Poplin
Barbara Randolph
Linda Sherman
Gary Stager

Managing Editor

Anita Best

ICCE Special Interest Group Coordinator

Keith Wetzel

Advertising Director

Kathleen Geygan

Publisher

International Council for
Computers in Education

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

CONTENTS

From the Editor		
	Sharon Burrowes	2
Monthly Musings		
	Tom Lough	2
Letters to the Editor		3
Stager's Stuff		
	Gary Stager	5
Logo LinX		
	Judi Harris	9
Math Worlds		
	A.J.(Sandy) Dawson	11
Customizing Your Logo Environment		
	Glen L. Bull and Gina L. Bull	13
Logo Picture Tools		
	Judi Harris	15
Pen Points		
	Ginny Nyahay	18
Using Apple Logo II and the Gappler		
	Tom Lough	19
Testudinal Testimony		
	Douglas H. Clements	20
LXionary		
	Bill Craig	24
The Adventures of Jacques and Elsie		
	Linda Sherman	25
IntLXual Challenges		
	Robs Muir	25
Logo Pals		
	Barbara Randolph	27
LX International Logo News		
	Dennis O. Harper	28

SIGLogo Membership (includes the Logo Exchange)

	U.S.	NON-U.S.
ICCE Member	24.95	29.95
Non-ICCE Member	29.95	34.95

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

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

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

From the Editor

Whether you are an old friend or a new reader, I want to welcome you to this second issue of the Logo Exchange in its new home with ICCE (International Council for Computers in Education). The Logo Exchange is now the journal of the newly formed SIGLogo (Logo Special Interest Group), and as such will play a broader role in the Logo community than it has in the past. Because the SIGLogo is a part of ICCE, an organization that has been in existence for some years, its readers have more of a vehicle for expressing their views through action, such as SIGLogo meetings at local and regional conferences. See Tom Lough's "Monthly Musings" for some ideas on how you can become a more active participant.

As I write this editorial, movement is afoot to establish officers and by-laws for the SIGLogo. This effort is being coordinated by Peter Rawitsch who teaches in New York state. In a future issue we will let you know the results of his work.

This month you will again find a number of familiar "faces." Take special note of Judi Harris' Logo LinX column with its delightful ties to fall colors! As usual, Robs Muir offers you a Logo challenge and Glen and Gina Bull give you some new insights. And we welcome back Sandy Dawson's MathWorlds and Doug Clements' Testudinal Testimony. In addition, two letters to the editor provide very different viewpoints on research on Logo.

This month we welcome a new columnist, Gary Stager. Some of you may know Gary from his frequent conference presentations. Gary's column "Stager's Stuff," will address project ideas in a variety of curriculum areas. Of the title of his column, Gary writes "The word 'stuff' has a number of positive images for me. It symbolizes the collection of stuff every teacher has in her/his bag of tricks as well as the collection of ideas, some gems and some junk, I have floating around in the attic of my mind. The word 'stuff' itself has a connection to many facets of Logo's philosophy. 'Messing about' and 'playful exploration' with Logo requires that there be a bunch of stuff — syntax and powerful ideas — for the learner to manipulate. Stuff is also a word which I was never allowed to use while doing 'serious' writing in school. This is an example of imposition and restriction of learning styles which Logo opposes. This monthly column will provide teachers with Logo tools and ideas for projects which have direct connection with a variety of curriculum areas. My goal is to share ideas for relevant Logo projects which can be instantly implemented by teachers of varying Logo expertise. I hope that this column will inspire educators to think of Logo's potential as an educational environment in ways they never perceived."

And finally, we need to hear from you. What do you want to see in the Logo Exchange? How can the SIGLogo be of assistance to you in your work? Better yet, submit an article for publication. It's a relatively painless process that allows others to share your ideas.

Sharon Burrowes, Editor
ICCE-SIGLogo
University of Oregon
1787 Agate Street
Eugene, Oregon 97403-9905



Monthly Musings

by Tom Lough

October is a special month, particularly this year. In addition to being the host of Halloween (in the USA), this October has been designated as "Computer Learning Month."

Computer use has exploded during the past five years to the point that, in several countries, it is unusual now to find a school without at least one computer. And, although the statistics show that only a small percentage of homes have personal computers, a closer look would reveal many computers embedded in the various appliances and toys. The computer has permeated our lives to a surprising extent. The special activities scheduled during the month of October will heighten this awareness.

For a number of students and teachers, Logo has become an important part of their computer related learning. But this opportunity is not yet available to all students everywhere. There are several factors which hinder the spread of Logo. Once such factor was brought forcefully to my attention this past summer.

I was selected as a specialist in residence for the "Information USA" exhibition touring the USSR as part of the current cultural exchange program. My job was to help the Soviet citizens who visited the exhibit understand better the role of the computer in US education. While talking with the Soviets, I introduced Logo into the discussion whenever it was appropriate. Without exception, the Soviets I talked to shrugged the language off as something "just for young children." I took great delight in showing them some of the things I had been doing with my college physics classes, and succeeded in convincing them to take a second look at this powerful learning tool.

Cover: George Beikler was a fifth grade student last spring when he and his adult tutor from 3M Company came up with the idea of making an airplane. George is now a sixth grader at Galtier Magnet Elementary School in St. Paul, Minn.

The point, of course, is that Logo is *not* just for young children. Educators, administrators, parents need to be made aware of this. As long as this misconception persists, the continued growth of Logo everywhere is in doubt.

How can we focus attention on Logo, and help more people understand what this marvelous language is really about? Should we follow the example of October and designate a future month of the school year as "International Logo Month" and distribute information packets and other materials? Perhaps this is an idea worth exploring and acting upon. I'd be interested in your thoughts and suggestions on this topic.

FD 100!

Tom Lough
PO Box 5341
Charlottesville, VA 22905

Note: Action on issues such as this can happen more easily now that the Logo Exchange is associated with an organization with members such as you. If you want to do something about an issue instead of just talking about it, then the SIGLogo provides your opportunity to get involved. For more information write to SIGLogo, ICCE, University of Oregon, 1787 Agate Street, Eugene, Oregon 97403-9905. ▲

Letters to the Editor

Dear Mr. Lough,

Thank you for your editorial in the May issue. At the time you pointed to the benefits of your exposure to Logo, you also pointed to some of the barriers to its wider acceptance. You might want to explore them in a subsequent editorial.

We face two significant problems in trying to measure the influence of Logo or any other exposure to educational computing. The first is that the Logo material is almost never presented as anything but an end in itself. It is not made available to the students as a tool they can use in a variety of situations. An idea cannot become a "powerful idea" unless the user can grasp it and apply it freely. In this sense, Logo is in the same position as mathematics. Mathematics is taught within a "set piece" environment — one need only look at the content of elementary and middle school texts to see page after page of drill and practice material. Does the student relate what he has learned of mathematics to problems in the world he faces? Absolutely not! One need only talk to young students about their interest in mathematics to see that the "powerful ideas" it embodies have been dismissed as unimportant.

The second problem is that we seek to measure things that are woefully difficult to measure. Month after month, "Logo Exchange" carries reports of dissertations and theses which seek correlation between teaching and testing at some level of statistical significance. Think about the things you mentioned in your article. How would you measure your increased pleasure in

finding and solving problems using Logo? Has writing in Logo changed the way you write in English? Has it changed the way you think about writing in English? By insisting on statistical correlations to "prove" the efficacy of Logo in an educational environment, we sink to treating education as a mechanical process. Papert's book all but screams out that new ways of looking at learning and teaching are required. A lot of good that did. Most schools and school systems have adopted the computer and have forced it into the classic subject pattern. Logo simply hasn't made a difference in these schools and it is unlikely to do so. Experiments which try to measure differences in learning based on exposure to Logo in a normal school setting are all but doomed to failure before they start. More than just Logo is needed.

I set up a Logo curriculum for a private school a few years ago. It was a little unusual in that it used such things as Raymond Smullyan's "The Lady or the Tiger" problems to stimulate procedural thinking and mental agility. The course was not to be graded and we would judge its success by seeing how many students came to the computer lab during the after-school "open computer hour" and by seeing the level of complexity of the things they were trying to do. I thought that, since we were just starting out, we probably need to be graded more than the students did. There has been considerable progress since our first stumbling beginnings. The curriculum now uses BASIC and has taken its place as part of the normal curriculum with grades, tests and standard textbooks. "Computing" is now a "subject."

I sometimes think that we would be better off if we came right out and stated that it was our goal to subvert the current educational process and replace it with something different. The gains we look for in our students are not the same as those looked for in a typical geography class. We want to open up the student's ability to think, even to the extent of thinking about thinking. In a typical geography class, remembering is enough.

FD 50?

John Boddie
Unusual Software
Box 7898
Newark, DE 19714

Dear Tom,

We read with interest your editorial column in May 1987 issue of Logo Exchange and wanted to share this past year's experience with you as well as some observations about research on Logo.

One year ago we decided to integrate Logo into the first grade curriculum of one first grade class. (From this point on this letter will make more sense if we write in "third person personal." Mary Ann was the first grade teacher carrying out a thesis research project. George was the thesis advisor and colleague for the research effort.) We had five first grade classes in a local public elementary school. Mary Ann's was the experimental group; the other four were pooled to provide a control or comparison group. The experimental group received one hour per day of pre-Logo, pre-keyboarding and Guided Discovery Logo experience. The other four classes received no Logo instruction and limited keyboarding experience of any type. We used two standardized measures (Peabody Mathematical Readiness Test and the Berry Visual Motor Integration Test) and one unobtrusive measure (attendance during the experimental period). At the end of the experimental period, the experimental group has significantly higher scores on the PMRT and VMI and significantly better attendance than the pooled comparison group. (We just received the Stanford Achievement Test results and while we have not completed the statistical analysis, the experimental group was higher on every subtest but spelling.) A legitimate question is this: Why did Logo have an impact in this setting when so many times there are "no significant difference?" We think there are several reasons.

First, much of the research literature — particularly the dissertation research literature — can be discounted as a true assessment of the effects of Logo because of either limited experimental treatment time and/or limited time on task, either because of high machine-student ratios or restricted access to laboratories.

Second, (at least it seemed to us) some of the dependent variables used in many of the studies are insensitive to the kinds of changes

that Logo may cause. A case in point is the whole genre of Locus of Control. We really doubt that learning to control the turtle is going to change measurable Locus of Control variables any more than learning to control the television or learning to drive a car. It is a nice idea; the internal-external measures we have seen are not sensitive enough to pick up any changes that might occur. Third, particularly with first graders, pre-Logo activities such as learning the placement of the keys on a simulated keyboard, knowing the difference between left and right, understanding the basic Logo commands, etc., should be taught before going to the laboratory. Time is wasted, we think and frustration enhanced, if the students are placed before the keyboard without any preparation. (In this regard, we chose a Guided Discovery Logo approach.)

Fourth, it may be that our first graders were different than the control group, although we could find no statistical or biographical data to suggest this. Students are placed in the first grade in a generally non-systematic way, with the one goal of being certain that black/white, boy/girl, and repeater/nonrepeater ratios are similar. (Incidentally, this was a public school in a district with heavy busing activity to achieve racial and socioeconomic balance.) We chose not to contaminate the study by heavily pre-testing all first graders, and our own analysis of available data showed no significant differences between the two groups.

Fifth, it may be teacher effects. Mary Ann's enthusiasm and creativity would probably have guaranteed success on the part of the students in the absence of Logo. On the other hand, by the end of the school year, these first graders were word processing with IBM Writing Assistant, writing letters, paragraphs, and short stories. (Incidentally, suggesting to us that the so-called "low end" word processors such as Bank Street Writer and LogoWriter are unnecessary.) They have acquired skills and values which should serve them well in future years.

Sixth, there may be unusual motivational currents affecting this first grade class. The class demonstrated their Logo skills at the Alabama Council for Computer Education annual meeting in January. This session was attended by other teachers, computer users and parents. The class was videotaped at least five times: twice by local TV stations for the evening news, once by the Chamber of Commerce for a public relations film, and twice by groups developing an in-service videotape on Logo instruction. Consequently, these students received a lot of recognition and external support. Moreover, the student's enthusiasm carried over to the home. About eight of the 27 or so students received computers for Christmas.

Our plans for next year are fluid. If we can do it, we would like to keep this class intact and find a second grade teacher interested in continuing the Logo and word processing effort, and possibly adding IBM Filing Assistant. We would like to find a way to get all of the teachers at one grade level, such as first grade, using Logo as part of their daily program. We have one teacher of 3rd and 4th grade "learning disabled" youngsters interested in seeing what effects Logo might have on them.

It would be inappropriate not to mention the help we received from individuals whose names frequently appear on your pages. Sharon Burrowes provided us with some early leads, as did Dennis Harper who George met in Singapore last fall. Jim Muller and the Logo Forum on CompuServe provided a lot of leads to resources. Tim Riordon initially got George interested in Logo at the University of California-Berkeley NCTI training program three years ago. Barbara Elias' column referred us to several dissertations we had missed in our original search. Please let this letter serve also as a note of appreciation to them.

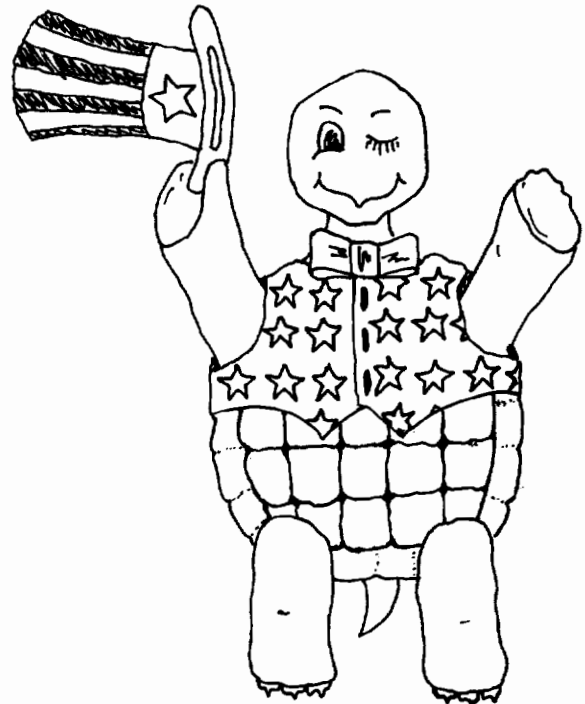
FD 100!

Mary Ann Robinson
George E. Uhlig
The College of Education
University of South Alabama
Mobile, Alabama 36688

STAGER'S STUFF

by Gary Stager

LogoWriter Adventure Stories and a Database of the U.S. Constitution



Adventure stories, stories which allow the "reader" to select the path through the story, are a popular programming project among students. Teachers encourage their students to create such branching stories because of their obvious connections to creative writing and the language arts curriculum.

In LogoWriter, adventure stories are simple to create. Even third graders can easily create a branching story with very little programming experience. This article will illustrate how the simple programming techniques used in creating an adventure story can be used to develop a social studies database.

The Page Environment

The ease of creating an adventure story in LogoWriter is due to the LogoWriter page environment. Like a "Choose Your Own Adventure" book, plot twists in a LogoWriter adventure send the reader to a different page. The reader is presented with a page (or more) of text and is then asked to make a decision. For example:

While walking home from school, you meet a large dog. Do you want to pet the dog or run away because the dog

might be mean? Type PET if you want to pet the dog or RUN if you want to run away.

The reader types one of the choices in the command center, the immediate mode area at the bottom of the screen. A rule of Logo grammar is that: when a word not preceded by a colon or quotes is typed, Logo thinks it is a procedure to be run and attempts to execute that procedure. Because of this rule, we know that there must be a Logo procedure on the flip-side of our page for each choice offered to us.

```
TO PET          TO RUN
GETPAGE "PET   GETPAGE "RUN
END            END
```

The LogoWriter primitive, GETPAGE (GP is the abbreviation), saves the current page and gets a specific page from your disk. GETPAGE requires a one word input, the name of a page on the disk.

The new page that is loaded contains the consequence of your decision, in other words, a new plot twist. This process of reading a page, making a decision, and having the story branch can go on and on. The adventure story is limited only by the student author's imagination and the most scarce resource, disk space.

(For more information on creating LogoWriter adventure stories, see pages 13-16 in the LogoWriter Word Adventures project booklet. The following light green activity cards introduce skills that may be of use in creating an adventure story or database: "Shape Your Stories," "Melody," "Score Sheet," and "Give Yourself Credit.")

What is a Database?

To oversimplify, a database is essentially a notecard file which you can quickly search for a particular piece of information. Of course databases can do many other things like sort large collections of data and make associations between pieces of information.

The potential of databases as an educational tool is enormous. Databases enhance research, critical thinking, inference, organizational, and writing skills. It is even thought that electronic databases in this information age will lead to a democratization of knowledge.

There are many commercial databases available, yet there appears to be a disproportionate emphasis placed on learning about databases rather than using a database as a personal tool. I believe that the educational power of databases lies in students creating the database themselves. Students do research on a particular topic, organize their data, and enter the information into a database to be used by themselves and others. I also agree with Seymour Papert that tools used by students (or anybody else) should be modifiable to meet their particular needs or interests. LogoWriter provides an environment in which children can create and modify their own tools. The ability to create and understand computer tools is a very powerful idea.

LogoWriter databases can be designed in a variety of ways which do not necessarily have to adhere to the rigid top-down

structure required by most commercial databases. Our next task will be to create a database on a timely topic, the signers of the United States Constitution. This database will be simple to construct yet can be expanded by students or teachers.

How to Get Started

Our database will begin by presenting the user with two choices for searching for information about a signer of the Constitution. The user may search by the state the signer represented at the Constitutional Convention or by the name of a particular signer. The first thing we must do is create a title page with the two options presented to the user. Follow these steps to do so.

- 1) Get a NEWPAGE from the CONTENTS page.
- 2) Name the page "CONSTITUTION" by typing NAMEPAGE "CONSTITUTION"
- 3) Press APPLE-U to wake up the cursor on the page and type the information you wish the user to read. Be sure to include the two options, SIGNERS & STATES. See figure 1.

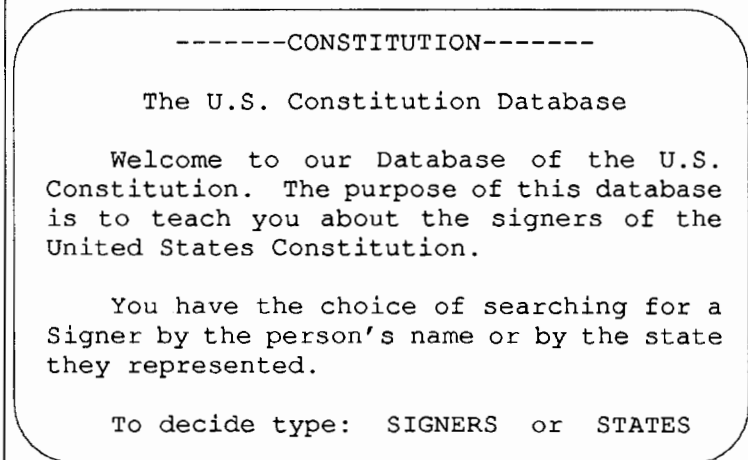


Figure 1

- 4) You probably wish to hide the turtle (HT) unless you have added some graphics to your page and wish the turtle to be visible.
- 5) When you are done entering the desired text on your page, press APPLE-F (either OPEN or CLOSED) to flip the page to the flip side.

Note: For all keystrokes which require the pressing of the open Apple key, press the CTRL-key on the IBM PC and PC Jr., and the Commodore key on the Commodore 64/128.

Every LogoWriter page has a flip-side which is used to teach LogoWriter a procedure for doing something or to leave comments or do wordprocessing (the flip-side is a wordprocessor just like the front of the page without the potential for graphics).

- 6) Write a Logo procedure on the flip-side for each option you presented on the front of your page. See figure 1a.


```

-----CONSTITUTION-----Flip Side

TO SIGNERS
GP "SIGNERS
END

TO STATES
GP "STATES
END
    
```

Figure 1a

Now we need to create a SIGNERS page and a STATES page.

- 7) Type NEWPAGE to save your current page and give you a blank one to work on.
- 8) Type NAMEPAGE "STATES
- 9) Hit APPLE-U to enter the wordprocessing mode and present some information and the options you wish the user to have. In this case, you will want to list all thirteen colonies as options. (The colonies became states upon ratification of the Constitution)
- 10) When you are done typing your information on the page, hit APPLE-F and create a Logo procedure for each of your options. This time you need to create a procedure for each of the thirteen original colonies. See figure 2.

```

-----STATES-----
The 13 Original Colonies

Choose the state whose representatives to
the Constitutional Convention you are inter-
ested in. For states with a two word title, put
a period between the words instead of a space.
Type one of the following:

CONNECTICUT          NEW. JERSEY
DELEWARE             NEW. YORK
GEORGIA              NORTH. CAROLINA
MARYLAND             PENNSYLVANIA
MASSACHUSETTS       RHODE. ISLAND
NEW. HAMPSHIRE       SOUTH. CAROLINA
                    VIRGINIA
    
```

Figure 2

NOTE: ALL procedure names in Logo must be one word. Therefore, the procedure for New Jersey must be called New.Jersey to adhere to the one word rule.

```

TO NEW.JERSEY
GP "NJ
END
    
```

A good way to name the state pages is to use their abbreviation.

11) When you are done writing your 13 procedures, hit ESC to save the page and select NEWPAGE from CONTENTS or type NEWPAGE.

12) Now create a SIGNERS page (repeat steps 7-10). See figure 3.

```

-----SIGNERS-----
The Signers of the Constitution

The following is a list of the 40 men who
signed the U.S. Constitution on September 17,
1787. Type the last name
as it appears in capital letters to choose a
signer. (Type NEXTSCREEN to see more text;
PRESCREEN to see this text.)

Abraham BALDWIN      David CARROLL
Richard BASSETT     George CLYMER
Gunning BEDFORD     Jonathan DAYTON
John BLAIR           John DICKINSON
William BLOUNT       William FEW
David BREARLY        William FITSIMONS
Jacob BROOM          Benjamin FRANKLIN
    
```

Figure 3

The SIGNERS page will have the names of the 40 men who signed the Constitution in 1787. Because there are two signers named Pinckney and Morris we must derive a scheme for naming the corresponding procedures on the flip-side of your page. One way might be to create a procedure PINCKNEY and C.C.PINCKNEY. Remember that we must instruct the user to type the correct procedure name or they will receive a friendly error message.

13) Save the SIGNERS page and begin a NEWPAGE.

14) A page now needs to be created for each of the thirteen states (see steps 7-10). On each state's page there should be a list of that state's signers of the Constitution. See Figure 4. A procedure must be written on the flip-side of the page to get a page for each signer.

```

-----NJ-----

New Jersey

The following men represented New Jersey at
the Constitutional Convention and were signers
of the United States Constitution.

Which signers of the Constitution are you
interested in learning more about?
David BREARLEY
Johnathan DAYTON
William LIVINGSTON
William PATERSON
    
```

Figure 4

A Collaborative Project

Creating a LogoWriter database is a wonderful group or class project. Once a teacher or group of students has decided on the format of the database, every student, regardless of ability, can contribute to the database.

Have each student choose one or two signers of the Constitution to research. The student must then create a Logo Writer page reporting what they learned about the particular signer of the Constitution. See figure 5.

```

-----PATERSON-----
      William Paterson

      William Paterson, born in 1745, was a member
of the New Jersey Constitutional Convention in
1776 and represented New Jersey at the Federal
Constitutional Convention in 1787.

      William Paterson served as a United States
Senator from New Jersey from 1789 - 1790 and as
governor of New Jersey from 1790 - 1793. In
1793, William Paterson was appointed as a
justice of the U.S. Supreme Court, where he
served until his death in 1806.

      Would you like to know more about his
CHILDHOOD or his term as GOVERNOR, SENATOR, or
Supreme Court JUSTICE.
  
```

Figure 5

At this point your class has created a very impressive educational tool by sharing the work load. One of the nicest aspects of this collaborative project is that once the format for the database is decided each student can independently research his/her subject and create the page(s) of data on that subject. Once each student has created their page and has it saved on the same disk with the CONSTITUTION, STATES, and SIGNERS pages the database is completed.

Potential Pitfalls

- 1) If a student names their page differently than the STATE or SIGNERS page requested, change the name of the page requested on one of the menu pages to reflect the actual name of the biographical page.
- 2) If you run out of disk space be creative. Split-up your states menus onto two disks or don't include research extensions
- 3) Pages must be named clearly. Be sure to keep careful track of the possible paths through the database.

The flexibility of this database allows a variety of extensions and embellishments to be made. If an enthusiastic student has done a large amount of research on a particular signer, the page named after the signer may present the reader with additional options.

For example:

Would you like to know about Gouverneur Morris' CHILDHOOD, POLITICAL career, civilian CAREER, or PERSONAL life?

The flip-side of the page should have procedures that look like:

```

TO CHILDHOOD
GP "CHILD.G.M
END
  
```

Additional pages should be imaginatively named as above so that there will not be duplication of page names.

If a signer's biography branches into more detailed records (pages) there should be an option procedure to return the user back to the last set of choices (menu). One way to do so would be to create a procedure for example on the CHILD.G.M page like the following:

```

TO MENU
GP "G.MORRIS
END
  
```

This page should return us to the first page pertaining to Gouverneur Morris, G.MORRIS.

It might be a good idea to have the following procedure on the flip-side of every page in your database:

```

TO START
GP "CONSTITUTION
END
  
```

4) If you chain pages too deeply, you may get an OUT OF SPACE message. This is because LogoWriter has to "remember" all of the procedure calls that led to the most recent choice. The solution is to minimize the length of the "path" through the procedures.

Conclusion

The ideas presented in this column are adaptable to a wide variety of curriculum areas. We have seen that adventure stories can be used in language arts and databases for social studies. How about creating a database for autobiographies of class members or current events or weather? Use your imagination!



Gary S. Stager is Director of Training, Network for Action in Microcomputer Education, 12 Locust Place Wayne, NJ 07470. CIS #73306,2446, Applelink address K0331.

Logo LinX

by Judi Harris

Leaf It to Logo

"Nature, to be commanded, must be obeyed."
—Francis Bacon, 1620

Many *Logo Exchange* readers can now look out their windows to see forests of red, orange, yellow, green and brown. Nature provides a timely opportunity in the fall to glimpse some of its mathematical and scientific wonders, hidden in the multi-colored carpet that showers and blankets the earth.

Discussions of photosynthesis can help students to understand why the leaves turn color. Leaf collections and trips to the woods can assist the identification and classification of tree species. Water-color techniques can be used to capture the beauty on paper. Information can be gathered and shared about the many ways that people use wood. After school, trees can provide a high and friendly perch, a rough leaning post, even "home base" in a game of tag.

And, as you may have already guessed, trees and leaves can be commanded—in Logo, of course. In the process, some fascinating mathematical patterns emerge.



*"....There's a leaf on the twig,
And a twig on the branch,
And a branch on the limb,
And a limb on the tree,
And a tree in the hole,
And a hole in the ground,
And the green grass grows all around, all around
And the green grass grows all around."*

—as sung by Pete Seeger

Nature provides many branching structures for us to see, from the veins on a leaf to the limbs on a tree. But, as Bacon says, nature must be obeyed to be commanded. And, to be obeyed, Nature's patterns must be understood.

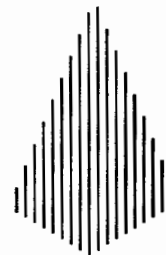
Vascular plants, such as lycopods, ferns, and trees, branch in basically two ways. Lycopods and some ferns branch dichotomously, as each new shoot subdivides or dichotomizes into two approximately equal apices. Each apice then grows for a time, and may dichotomize once again. The majority of vascular plants, though, branch laterally, in which the main shoot develops branches from buds in the leaf axils. Branch shoots then branch and rebranch in similar patterns.

A New Leaf

Leaves have vein configurations that also form branching structures, and leaf shapes accommodate venation. Veins are hollow tubules that transport starch made in the leaves to the stem, and down to the roots through phloem tubes. Veins also transport water, taken in by the roots, up to the leaves through xylem tubes.

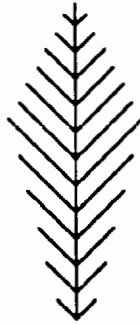
There are three basic vein patterns in the leaves of deciduous trees: parallel, pinnate, and palmate. As an example of how mathematical patterns can be pinpointed in nature, then explored and generalized with Logo, let us examine leaf vein structures with the turtle.

Parallel venation, when drawn with the turtle, looks like this.



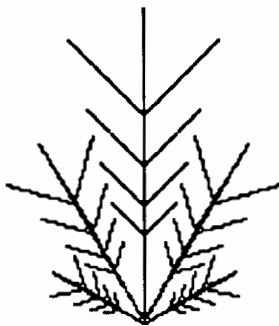
Parallel veins appear on flowering plants that are monocotyledons. A monocotyledon is a plant that has only one cotyledon, or seed leaf. The cotyledon may store food for the young plant before it can begin to produce true leaves, which take over food production.

Pinnate venation, when approximated with turtle graphics, looks like this.



Veins, lobes (sections of leaves) and leaves themselves can be arranged in this pattern. The segments of a bird's feather also have a pinnate structure: opposite or alternate from one another along a common axis.

Palmate veins look something like this.



The structure refers to leaves, lobes, or veins radiating from a single point.

Be-leaf it or Not

Look closely at these diagrams. There is a pattern to the patterns. (And the pattern is mirrored in that statement!) The Encyclopedia Britannica defines "branching process" as:

"stochastic process in which an object can generate other objects, each of which can generate more objects, and so on." (Micropaedia, 2:231)

Sound like recursion? Of course! Nature is infinitely recursive. Recursion, then, can be used to generate the vein patterns discussed above.

In this code, used to draw parallel vein patterns, a copy of the procedure is made in the computer's memory for each vein on the left half of the leaf.

```
TO PARALLEL :LENGTH :INCREMENT :LIMIT
IF :LENGTH > :LIMIT [STOP]
```

```
PU
BK 2
RT 90
FD 3
LT 90
PD
FD :LENGTH + :INCREMENT
BK :LENGTH + :INCREMENT
PARALLEL :LENGTH + 8 :INCREMENT + 2 :LIMIT
PU
FD 2
RT 90
FD 3 L
T 90
PD
FD :LENGTH + :INCREMENT
BK :LENGTH + :INCREMENT
END
```

The veins increase by 8 turtle steps plus an increment in length for each copy. Once the :LIMIT is exceeded, the multiple copies of the procedure are completed in reverse order of creation, with descending vein lengths. The recursive call marks the midpoint of the vein structure, or its line of symmetry.

(The inputs used for the sample leaf were: PARALLEL 5 3 75.)

This procedure makes multiple "v" shapes in increasing size,

```
TO PINNATE :LENGTH :LIMIT
IF :LENGTH > :LIMIT [STOP]
LT 45
FD :LENGTH
BK :LENGTH
RT 90
FD :LENGTH
BK :LENGTH
LT 45
FD 10
PINNATE :LENGTH + 5 :LIMIT
LT 45
FD :LENGTH - 5
BK :LENGTH - 5
RT 90
FD :LENGTH - 5
BK :LENGTH - 5
LT 45
FD 10
END
```

with a 10 turtle-step separation as the veins reach toward the tip of the leaf. Once the :LIMIT is exceeded, the other half of the leaf is completed. Unlike the PARALLEL vein pattern, where the line of symmetry is lateral (along the middle vein,) the line of symmetry for the PINNATE leaf is horizontal.

(The inputs used for the sample leaf were: PINNATE 10 35.)

Finally, these commands make "vees" of increasing size, arranged from one radiating point. Note that this is the only tail-end recursive procedure in the set.

```

TO PALMATE :LENGTH :LIMIT
IF :LENGTH > :LIMIT [STOP]
VEE :LENGTH :LIMIT
BK :LENGTH
LT 60
VEE :LENGTH * 0.33 :LIMIT
BK :LENGTH * 0.33
RT 30
VEE :LENGTH * 0.66 :LIMIT
BK :LENGTH * 0.66
RT 60
VEE :LENGTH * 0.66 :LIMIT
BK :LENGTH * 0.66
RT 30
VEE :LENGTH * 0.33 :LIMIT
BK :LENGTH * 0.33
LT 60
PALMATE :LENGTH * 1.33 :LIMIT
END

```

```

TO VEE :LENGTH :LIMIT
FD :LENGTH
BK :LENGTH / 3
LT 45
FD :LENGTH / 3
BK :LENGTH / 3
RT 90
FD :LENGTH / 3
BK :LENGTH / 3
LT 45
FD :LENGTH / 3
END

```

(The inputs used for the sample leaf were: PALMATE 50 150.)

Goethe's Guidelines

"Nature goes her own way, and all that to us seems an exception is really according to order."

—Goethe (1824)

I'll never forget the day when an excited 6-year-old came running to me during recess, eyes bright, with a beautiful red leaf in her hand. She held it up for me to see, using a nearby tree as the backdrop.

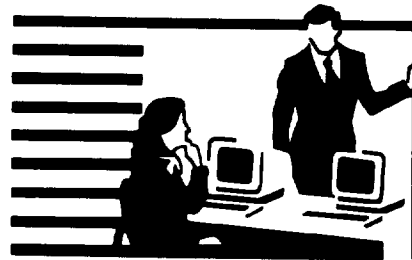
"See?" she exclaimed. "The branches on the leaf are like the ones on the tree."

I have her to thank for revealing a recursive series of mysterious natural patterns, that extend from the leaves, to the branches, to the roots of a powerful idea.

Judi Harris was an elementary school computer use facilitator, graduate education instructor, and computer consultant for a number of public and private schools in Pennsylvania. She is now a doctoral student in education at the University of Virginia. Her CompuServe number is 75116,1207.



Professional Development Workshops For Your School District



ICCE offers you a skilled cadre of presenters on topics of immediate interest to educators. And we will design workshops to meet your special needs! Examples of workshops offered include:

- Large Group:** *Problem Solving Across the Curriculum*
Long Range Planning for Computers in Schools
Brains, Computers, and Education
Integrating Computers into Mathematics Instruction
- Small Group:** *Using Databases and Spreadsheets in Social Studies*
Trainer's Workshop: Keyboard Success
AppleWorks Trainer's Workshop
Using LogoWriter in the Curriculum
Microsoft Works for the Mac
Telecommunications
Two Utility Programs for the Teacher's Toolkit
Teaching Pascal
Teaching Writing with Computers

If you would like to see workshops conducted in your area, give us a call! Keep in touch with the constant changes happening in educational technology.

For further information, prices, and dates, contact Keith Wetzel, ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403; ph. 503/686-4414.



MathWorlds

edited by

A. J. (Sandy) Dawson

The question is often raised, more so in the recent past it seems, as to whether Logo will survive. After spending two weeks in Montreal this July attending, first, the third annual Logo and Mathematics Education conference (LME 3), and, second, the eleventh annual Psychology and Mathematics Education (PME XI), where sessions on Logo were very much in evidence; and after spending numerous hours during August signing new members onto the CLIME electronic network, I have no doubts as the health of the Logo and mathematics education community. Perhaps a few details about each of these endeavors would be appropriate to lead off my first column for the '87-'88 edition of the newlook, ICCE-sponsored Logo Exchange.

In the May edition of **Logo Exchange** an announcement was made of the publication of the first **CLIME** newsletter. For those of you who may not have heard of CLIME, it is the Council for Logo in Mathematics Education. Since that announcement, a second newsletter has appeared and it seems that CLIME is well along the road to seeking affiliation with the National Council of Teachers of Mathematics (NCTM). The membership in the organization has passed the 150 mark and is growing daily. A message sent to all members on the CLIME electronic bulletin board operated through Simon Fraser University now takes sixty-five seconds to process! As reported in the second newsletter, membership in CLIME is drawn from 34 states as well as Canada, Mexico, Puerto Rico, and Ireland. The idea for the CLIME group arose from a group discussion only eighteen months ago. Tremendous progress in the formation of the group has occurred during that brief period. Ihor Charischak, Reinhold Wappler, George Bright, and others have been instrumental in creating an organization and a newsletter which speaks directly to those whose primary interest is the teaching and learning of mathematics and the role which Logo can play in that process. Articles in the second newsletter include Al Cuoco talking about the impact the introduction Logo has had on his teaching style, but he notes that he is most taken with the openings Logo provides to students for self-generated original mathematics; Phil Lewis sharing a personal reflection on the breadth and applicability of Logo to algebra, and the realization of its impact on both teaching and learning; Richard Binswanger pointing out the concrete nature of the Logo experience for kids, and why children need strong encouragement to trust and use knowledge so gained in their regular curricular mathematics; and many more pithy articles! Membership in CLIME is obviously well worth the \$5. Write to CLIME, c/o Ihor Charischak, 445 Broadway, Hastings-on-Hudson, NY 10706. Hopefully, CLIMers will be guest editing a column or two of **MathWorlds** in the near future.

The Logo and Mathematics Education group (LME) though centred on the relationship between Logo and mathematics has a theoretical orientation in contrast to the more practical focus of the CLIME group. Like the CLIME group, LME arose from a discussion amongst Uri Leron, Celia Hoyles, Richard Noss, and Joel Hillel at the Logo '84 meeting. What they sought in the LME group was an opportunity to share research interests and studies on the use of Logo in the teaching and learning of mathematics. The first two conferences were held in London, England under the sponsorship of the London Institute for the Study of Education. The third conference, as noted above, was held in Montreal this past July.

Indeed, the opening of the conference was dramatic as it was the evening of the July storm which closed down Montreal for nearly a day. Needless to say, many of the LME3 participants were late in arriving, including the keynote speaker, Tom Kieren. Joel Hillel did a marvellous job in totally rearranging the conference schedule to accommodate the fact that the campus of Concordia University, where the conference was held, was completely without power. The order of presentations at the conference was determined by whether or not the speaker had transparencies, and hence needed power for an overhead projector. John Mason, Tom Kieren, Uri Leron, Celia Hoyles and Richard Noss presented stimulating plenary sessions. The research reporting sessions were rich in ideas and abundant in discussion. Future columns will report specifically on some of these sessions—as soon as the Conference proceedings become available. Should you wish to obtain a copy of the proceedings, write to Dr. Joel Hillel, Mathematics Department, Concordia University, 7141 Sherbrooke Street West, Montreal, Quebec, H4B 1R6.

The PME conference has a much broader scope than Logo and mathematics covering as it does the entire range of research in the psychology of learning mathematics. It was pleasing to see, therefore, the strong representation made by Logo/mathematics educators to both the general program as well as to specific study groups within the conference. The British, Israeli and Canadian delegations were particularly strong in putting forth comprehensive research reports on Logo and mathematics.

All this activity—CLIME, LME3 and PME XI bodes well for the health and growth of the use of Logo in mathematics education. I think we can safely say that Logo is alive and well and living in mathematics classrooms around the world.

But this column is not just about events and activities which happened in the past. It is also about things Logo/mathematics teachers and researchers are doing right now!

Last spring Tom Lough sent me a couple of short pieces written by a teacher which, as Tom said, were "kind of nice." I have been waiting for an opportunity to include them in a column but space always seemed used up. This month, however, there is room and so I am very happy to have this chance to let Ron Bierema share his insights about the mathematical patterns discoverable in multiple recursion.

Mathematical Patterns of Multiple Recursion

by
Ron Bierema

Trying to predict the results of procedures with multiple recursions can be very frustrating. However, multiple recursions do have recognizable patterns. For example, try to predict the final value for TOTAL that will be printed out in the procedure BEGIN.

```
TO BEGIN
  MAKE "TOTAL 0
  MULT.RECUR 1
  PR :TOTAL
END

TO ADD.ONE
  MAKE "TOTAL :TOTAL +1
END

TO MULT.RECUR :NUM
  IF :NUM = 0 STOP
  ADD.ONE
  MULT.RECUR :NUM - 1
  ADD.ONE
  MULT.RECUR :NUM -
END
```

Since the beginning value for NUM is 1, TOTAL will only reach 2. But, suppose we increase the beginning value for NUM to 2, allowing for 2 "levels" of recursion. In this case, the TOTAL reaches 6. Table 1 summarizes the increase in the TOTAL as NUM (signifying the levels of recursion) is increased. As you may notice, TOTAL has a very interesting pattern. As the level of recursion is increased by one, the new TOTAL can be calculated by adding 1 to the previous TOTAL and multiplying this by 2. There is also a mathematical formula which generalizes this pattern. For 'n' levels of recursion the TOTAL will be $2(2^n - 1)$.

<u>LEVELS OF RECURSION</u>	<u>TOTAL</u>
1	2
2	6
3	14
4	30

Table 1

There are, however, more patterns to be discovered here. If you add one more recursion to the main procedure (i.e. another line of ADD.ON and MULT.RECUR :NUM -1) and vary the levels of recursion, you will obtain the results of Table 2. Again there is

another pattern here. As the level of recursion is increased by one, the new TOTAL can be calculated by adding 1 to the previous TOTAL and multiplying this result by 3. The mathematical formula here for "n" levels of recursion is $3(3^n - 1)$. Compare this to the first formula!

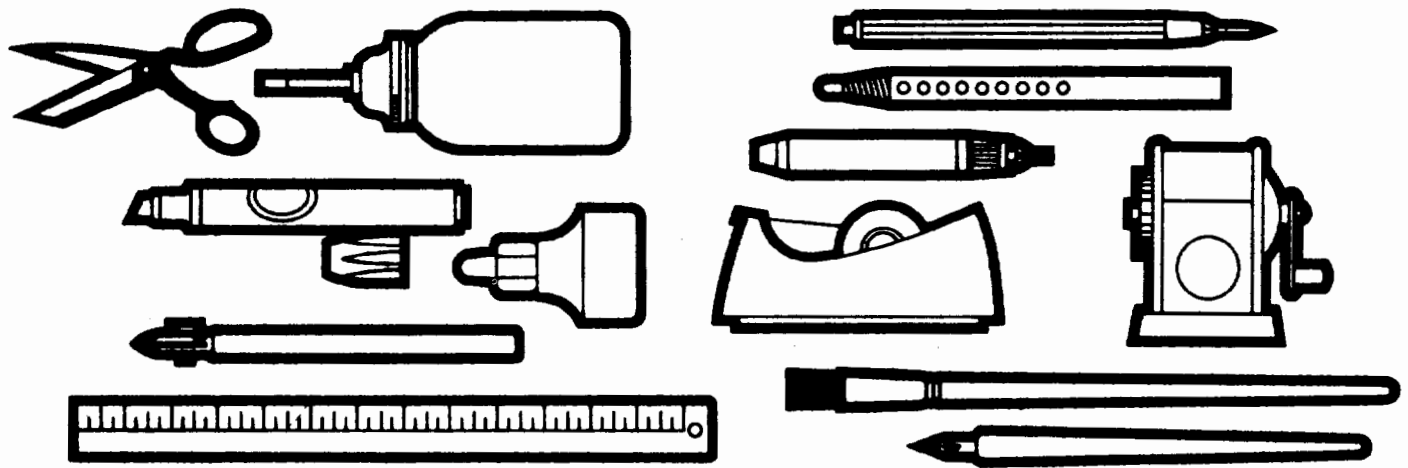
<u>LEVELS OF RECURSION</u>	<u>TOTAL</u>
1	3
2	12
3	39
4	120

Table 2

To satisfy your curiosity, after adding another recursion to the main procedure, the formula for "n" levels is $4(4^n - 1)$. In general, then, for "m" recursions at the n^{th} level of recursion, the TOTAL will be $m(m^n - 1)$. Now do you agree that multiple recursions have patterns?

I want to thank Ron Bierema (422 Edgemill Rd, Glendale, PA 19038) for sending this article to us. And I want to encourage other readers to send along their insights and reports on their experiences in teaching Logo so that these might be shared with others. Join CLIME and you can send me your articles electronically from anywhere in North America with no charge to you except perhaps a local or very short long distance phone call. The CLIME bulletin board, which I run, is open free of charge to any CLIME member, so join up and then send me your contribution to MathWorlds. ▲

A. J. (Sandy) Dawson is a member of the Faculty of Education at Simon Fraser University in Vancouver, British Columbia. His Compuserve number is 76475,1315. He can also be reached electronically as Sandy_Dawson@SFU.BITNET or Sandy_Dawson%SFU@um.cc.umich.ed



CUSTOMIZING YOUR LOGO ENVIRONMENT

by
Glen L. Bull and Gina L. Bull

This column is about Logo tools. The idea of software tools is a common one in computer science. In computer science a software tool is a piece of code that can be used in many different applications. Reusability of code is a hot topic in computer science because reinvention of code to do the same task is expensive. A hierarchy of tools also makes it possible to work at a more abstract level. Thus the user can focus on the end goal rather than the minutiae of implementation details.

The procedure SQUARE is a classic Logo tool. Everyone uses this procedure, and it is generally written in the same way across many different applications. The procedure SQUARE makes it easier to write the superprocedure HOUSE, because the number of commands in the superprocedure is decreased. The modularity afforded by software tools also makes it easier to isolate bugs. Once SQUARE has been tested and used in many different applications, you can be reasonably sure that the bug lies elsewhere.

Operating Systems and Microworlds

A tool which must be loaded when the need for it arises is much less likely to be used than one which is always available. Searching for a tool file in the catalog of a disk, or, worse yet, on several disks will dampen or extinguish the spark which created the desire to use the tool. Elevating a tool to the status of a built-in instruction by guaranteeing its availability is a way of customizing the Logo environment.

An operating system is a collection of programs, often "invisible" to the user that provide various capabilities to the users. Many operating systems permit the user to customize the working environment by means of files which are automatically executed at the beginning of each work session. In some cases it is possible to add customized commands (tools) at the same time. For example, in UNIX, an operating system used in many university environments, a file with a name such as "PROFILE" is executed when the user logs on. This file generally contains a user's definitions of useful tools

and gives values to certain "environment variables." In MS-DOS (the operating system used for IBM PCs and IBM-compatible computers) there is also a file which automatically executes a batch of commands when the computer is turned on. This file is called the AUTOEXEC.BAT file. Apple's current operating system, after loading the PRODOS file and a file with the suffix ".SYSTEM," automatically runs a file named STARTUP. In a sense, a customized operating environment becomes a microworld for the user.

Startup Files

Some dialects of Logo also allow the user to create a file which is automatically loaded along with Logo. For example, in LCS versions of Logo, this file is called STARTUP. What sorts of things might you want to include in your Logo environment? Suppose you are trying to encourage students to use color names. Possibly you would like them to type:

```
?SETPC BLUE
```

instead of typing this:

```
?SETPC 5
```

This is easy enough to arrange. The following procedures will permit your students to use color names (instead of numbers) if you are using a version of Logo on the Apple:

```
TO BLACK      TO WHITE
OP 0          OP 1
END          END

TO GREEN      TO PURPLE
OP 2          OP 3
END          END

TO ORANGE     TO BLUE
OP 4          OP 5
END          END
```

If these procedures are stored in a file (possibly COLOR.FILE) the color names can be loaded in at any time. However, if you are working with a class of third-graders, the step of typing LOAD "COLOR.FILE" may be beyond them. The teacher could load the file for them, of course, but this is not necessary. If the procedures are saved in a file with the name STARTUP, the procedures will automatically be loaded with Logo.

With older versions of Logo, tools loaded through the STARTUP file were not as useful, because memory restrictions made it necessary to limit the number of procedures in memory. With newer versions of Logo, it is not necessary to be so frugal.

The BURY Command

Loading in the startup tools with the STARTUP file does not solve all of the problems of configuring an environment. One of the most obvious problems is that the startup tools will be saved in the student's files. Also the first time the student types ERALL to clear the workspace, the startup tools will disappear. Finally, any reasonable number of startup tools will clutter up the workspace and interfere with examination of procedures created by the student. If the student creates SQUARE, TRIANGLE, and HOUSE, and typing POTS produces this:

```
BLUE
SQUARE
ORANGE
PURPLE
HOUSE
BLACK
TRIANGLE
WHITE
GREEN
```

it can be difficult to pick out the student's procedures from the startup tools.

There is a command, available in Apple Logo II and LCSII Logo II, which takes us a step further toward a customized operating environment. The command is BURYALL. This command "buries" all the procedures in the workspace. When procedures are buried, they are not erased by ERALL. Buried procedures are not saved by the SAVE command, and they do not appear when POTS is typed. In other words, buried procedures behave almost as though they were built-in commands.

(Note: The concept of "burying" procedures is available in many LCSII versions of Logo. For example, Apple Logo and IBM Logo contain the primitive BURY to bury "packages" of procedures, and LogoWriter allows you to create "tools" which behave like buried procedures.)

The STARTUP Variable

The STARTUP file provides a means of automatically loading tools when Logo is started. The BURYALL command makes these automatically loaded tools behave like built-in commands such as FORWARD or BACK. It would be possible to manually type BURYALL just after Logo is loaded to accomplish this. However, this would require the teacher to stand by as Logo loaded, and defeat

the purpose of automatically-loading tools. The STARTUP variable provides a way of automatically burying the tools loaded from the STARTUP file.

Essentially, the STARTUP variable is a list of commands in the startup file which are automatically executed when Logo is loaded. The startup variable is created like this:

```
?MAKE "STARTUP [BURYALL PR [COLOR TOOLS
LOADED. ]]
```

The startup variable is saved in the STARTUP file along with all the startup tool procedures. If there are several commands which you wish to have automatically executed, you may want to place a special SETUP procedure in the STARTUP file. The SETUP procedure could look like this:

```
TO SETUP
BURYALL
PRINT [Color Tools Loaded.]
END
```

In that case, this kind of startup of variable would be created:

```
?MAKE "STARTUP [SETUP]
```

In LogoWriter, any procedure named STARTUP is automatically run when a page is loaded. So, for LogoWriter the procedure

```
TO STARTUP
GETTOOLS "COLOR.TOOLS
PRINT [Color Tools Loaded]
END
```

could be saved on a page named STARTUP, while the page named COLOR.TOOLS would contain the color procedures.

Note that it is important to document which startup tools have been automatically loaded with Logo. Otherwise, students who have become accustomed to using a particular set of startup tools will think the computer is "broken" when they use a plain-vanilla version of Logo without startup tools. (See, for example, "Circles, Arcs, and Headaches," by Sharon Burrowes, *The National Logo Exchange*. (Vol. 3, No. 5) January 1985, p. 1-2.) The power of customizing an operating environment is so great that it almost seems as though the computer is broken if a special configuration is not available. This is true for UNIX, MS-DOS systems, and Logo.

Site Licensing

The STARTUP file, the BURYALL command, and the STARTUP variable all work together to permit the Logo environment to be customized. A fourth element permitting even greater customization has recently become available. This breakthrough was achieved not through computer science, but through a licensing plan.

Earlier this year Logo Computer Systems, Inc. (LCSII) reacquired the rights to Apple Logo II from Apple. The firm then reissued Apple Logo II (with a few improvements) as LCSII Logo II. The major breakthrough for LCSII Logo II is that a site license for it is available to schools. A site license means that LCSII Logo II is not

copy protected, and that the site purchasing it can make copies for as many computers as are available on site. (LCSI Logo II is also available in single copies and ten-packs.)

Let's see why site licensing is such a major breakthrough for customization of Logo environments. Suppose that a teacher with a lab of 10 computers wants to set up a Logo environment with poetry tools for an English class, an environment with tools for data acquisition and sensors for science class, and an automatically loading Logo INSTANT program for the first grade. It is only possible to have one startup file per disk. Thus a teacher with ten copies of Logo (one for each machine) has previously been forced to choose one set of startup tools for all three classes.

The result is inevitably a compromise. In this case one size does not fit all. A teacher who wishes a class to explore the concept of Pi by constructing circle procedures definitely does not want a built-in circle tool. However, the teacher working with a class graphing the results of a survey with a pie chart will probably regard circle tools as a means rather than an end.

With a site license, it is possible to have more copies of Logo than there are machines in the building. One set of Logo disks may be customized for science class, one set for English class, and one set for the first grade. It is even possible and desirable to have customized versions of Logo for art class and music class. Each of these Logo environments represents a different microworld.

The ability to have specialized startup files which automatically load and run is not only useful in lab situations. For example, consider a situation in which a hesitant first-grade teacher would like to try INSTANT Logo with her children. Only a couple of instructions are required to tell her to "First load Logo, and then load the INSTANT file, and then type 'INSTANT'." However, if the teacher is accustomed to commercial software which automatically loads and runs, even those three steps may be too many. With automatically loading STARTUP files and startup variables, it is possible to have an INSTANT disk which automatically loads and runs. Then all the teacher has to do is place the disk in the drive and turn the computer on. Versions of Logo which automatically load and run programs are useful for creation of demo disks and a variety of other purposes as well.

National Educational Computing Conference (NECC) '88

Loews Anatole Hotel, Dallas, TX, June 15-17, 1988. For more information contact Jim Poirot, NECC '88, Department of Computer Education and Cognitive Systems, North Texas State University, P.O. Box 13886, NT Station, Denton, Texas, 76203-3886.

Even major business software firms have shied away from site licenses, and with good reason. When large quantities of a software product are available, there is the distinct danger that some copies will find their way to sites which are not licensed. LCSI has made a courageous marketing decision to offer its products in the form of a site license. It is to be hoped that the results of this experiment will be financially as well as educationally rewarding so that other courseware developers will offer their products in this form as well. Currently the site license option is only economically advantageous for schools with large numbers of computers, but perhaps a way will be found to confer similar benefits on schools with fewer computers.

Summary

The four components discussed in this column which particularly facilitate development of customized Logo environments are:

1.
the STARTUP file
2.
the BURYALL command
3.
the STARTUP variable
4.
the Logo site license

The STARTUP file automatically loads a set of Logo tools when Logo is started, the STARTUP variable automatically runs a list of Logo procedures when Logo is started, the BURYALL commands places tool procedures in the background so that they behave like built-in Logo commands, and the Logo site license makes it possible to develop several different customized versions of Logo for a single computer.

Teaching tools are Logo procedures provided by the teacher which facilitate exploration of a particular content area — in English, mathematics, science, or social studies. The four elements described above greatly facilitate use of teaching tools, because they make it more convenient to have a specialized Logo environment in which a particular set of tools is present. ▲

Logo Picture Tools

by Judi Harris

Purpose

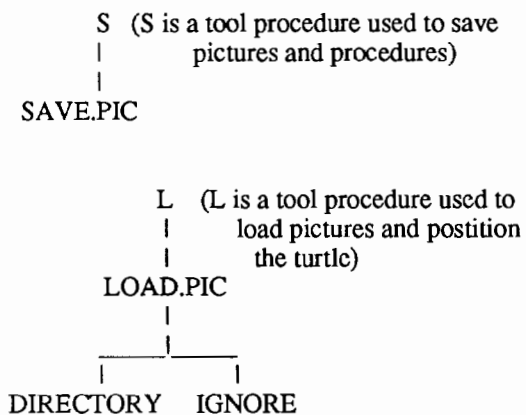
Many versions of Logo now have a primitive (such as SAVE-PIC or SAVEPIC) that allows users to save a picture drawn with turtle graphics commands. The picture file that this creates contains only the screen image. This is especially convenient to use with complex pictures that can take the turtle 20 minutes to draw. SAVEPIC(T)'s most powerful use, though, is with young children and other beginning Logo users, who have not yet learned to write procedures, but nevertheless like to save their turtle graphics work. SAVEPIC(T) allows the novice user to save the screen image to disk, and continue drawing at another time.

There is substantial inconvenience that results from the use of the command, though. When the user reloads the picture to continue drawing, the turtle appears at HOME (in the center of the screen,) with HEADING 0, no matter where it was on the screen when the picture was saved, or in what direction it was pointing. It is very frustrating and inconvenient for the user to reset the turtle to continue drawing; it is, in fact, unrealistic in many cases to suggest that a beginning student even attempt to do this. This considerably limits the control that the user has over the picture's final appearance, and discourages detailed, long-term projects. Many of the students that I have worked with simply start a new picture when they cannot reposition the turtle as they wish.

The procedures that I have written provide one solution to this problem.

Procedures

Here is a "procedure tree" that charts the PICTOOLS procedures:



The superprocedures S and L give directions to the user, allowing them to save pictures that they have drawn and load pictures that have been saved to disk. If you would rather not be prompted for a picture name, use SAVE.PIC "some.file.name or LOAD.PIC "some.file.name. Both the prompted and unprompted versions of the procedures do the following:

S or SAVE.PIC

- saves the picture currently on the screen in a disk file with the name

that you choose plus .PIC

- saves all of the procedures in memory (including the picture tool procedures) to a second disk file with the name that you chose for the picture, without the .PIC at the end
- saves the turtle's position and heading in the second disk file as two global variable values

L or LOAD.PIC

- loads the picture file from the disk with the name that you chose when you saved the picture, and displays the picture
- loads the procedure file from the disk with the same name
- repositions and resets the turtle's heading, so that it is in the same place and facing the same direction that it was when you saved the picture

Hints for Tool Use

- If you are typing the procedures on the following pages for the first time, be sure there is nothing else in the workspace, and save them to your data disk by typing SAVE "PICTOOLS .
- Remember to load the picture tools into the computer's memory at the start of each session by typing LOAD "PICTOOLS . That way, the tools are ready to save and/or load pictures as the user needs to do so.
- (IBM version only) The program expects the data diskette (where the pictures are stored) to be in drive A. If you would like to use the data diskette in drive B, type SETDISK "B before saving or loading files.

You are encouraged to modify and append these tools to fit your specific personal and instructional needs. I hope that they are helpful in your Logo adventures.

Judi Harris

University of VA
Curry School of Ed.
405 Emmet Street
Charlottesville, VA 22903

Logo Picture Tools: IBM

```

TO S
  CLEARTEXT
  PR [PLEASE TYPE A NAME FOR THIS PICTURE, ]
  PR [THEN PRESS < RETURN > .]
  MAKE "NAME READWORD
  IF WORDP :NAME [SAVE.PIC :NAME] [S]
  END

TO L
  CLEARTEXT
  PR [PLEASE TYPE THE NAME OF THE PICTURE]
  PR [THAT YOU WOULD LIKE TO SEE.]
  PR [IF YOU WOULD LIKE TO SEE A LIST OF THE]
  PR [PICTURES ON THIS DISK, TYPE D.]
  MAKE "NAME READWORD
  IF :NAME = "D [DIRECTORY STOP]
  IF WORDP :NAME [LOAD.PIC :NAME] [L]
  END

TO DIRECTORY
  TEXTSCREEN
  CLEARTEXT
  
```

```
PR [THIS IS A LIST OF ALL OF THE PICTURES]
PR [ON THIS DISK:]
PR []
DIR "\*.PIC
PR []
PR [PRESS ANY KEY TO GO ON.]
IGNORE RC
END
```

```
TO IGNORE :CHARACTER
END
```

```
TO SAVE.PIC :NAME
HT
FULLSCREEN
MAKE (WORD :NAME "POS) POS
MAKE (WORD :NAME "HEADING) HEADING
SAVEPIC (WORD :NAME ".PIC)
SAVE :NAME
ST
CLEARTEXT
SPLITSSCREEN
END
```

```
TO LOAD.PIC :NAME
LOAD :NAME
LOADPIC (WORD :NAME ".PIC)
PU
SETPOS THING (WORD :NAME "POS)
PD
SETHEADING THING (WORD :NAME "HEADING)
ST
CLEARTEXT
SPLITSSCREEN
END
```

Logo Picture Tools: Apple Logo II

NOTE to Apple Logo II Users: If this is the first time that you are using the picture tools on a particular data diskette, BE SURE to type in the commands, CREATEDIR "PIX and CREATEDIR "PIXDO before saving any pictures. This will create subdirectories on the data diskette for your picture and picture procedure files.

```
TO S
CLEARTEXT
PR [PLEASE TYPE A NAME FOR THIS PICTURE,]
PR [THEN PRESS < RETURN > .]
MAKE "NAME READWORD
IF WORDP :NAME [SAVE.PIC :NAME] [S]
END
```

```
TO L
CLEARTEXT
PR [PLEASE TYPE THE NAME OF THE PICTURE]
PR [THAT YOU WOULD LIKE TO SEE.]
PR [IF YOU WOULD LIKE TO SEE A LIST OF THE]
PR [PICTURES ON THIS DISK, TYPE D.]
MAKE "NAME READWORD
IF :NAME = "D [DIRECTORY STOP]
IF WORDP :NAME [LOAD.PIC :NAME] [L]
```

```
END
```

```
TO DIRECTORY
TEXTSCREEN
CLEARTEXT
PR [THIS IS A LIST OF ALL OF THE PICTURES]
PR [ON THIS DISK:]
PR []
SETPREFIX "PIX
CATALOG
SETPREFIX FIRST ONLINE
PR []
PR [PRESS ANY KEY TO GO ON.]
IGNORE RC
END
```

```
TO IGNORE :CHARACTER
END
```

```
TO SAVE.PIC :NAME
HT
FULLSCREEN
MAKE (WORD :NAME "POS) POS
MAKE (WORD :NAME "HEADING) HEADING
SETPREFIX "PIX
SAVEPIC (WORD :NAME ".PIC)
SETPREFIX FIRST ONLINE
SETPREFIX "PIXPROC
SAVE :NAME
SETPREFIX FIRST ONLINE
ST
CLEARTEXT
SPLITSSCREEN
END
```

```
TO LOAD.PIC :NAME
SETPREFIX "PIXDOC
LOAD :NAME
SETPREFIX FIRST ONLINE
SETPREFIX "PIX
LOADPIC (WORD :NAME ".PIC)
SETPREFIX FIRST ONLINE
PU
SETPOS THING (WORD :NAME "POS)
PD
SETHEADING THING (WORD :NAME "HEADING)
ST
CLEARTEXT
SPLITSSCREEN
END
```

Logo Picture Tools:" Terrapin Version 2.0 & H

```
TO S
CLEARTEXT
PR [PLEASE TYPE A NAME FOR THIS PICTURE,]
PR [THEN PRESS < RETURN > .]
MAKE "NAME FIRST REQUEST
IF WORD? :NAME THEN SAVE.PIC :NAME ELSE S
END

TO L
```



```

CLEARTEXT
PR [PLEASE TYPE THE NAME OF THE PICTURE]
PR [THAT YOU WOULD LIKE TO SEE.]
PR [IF YOU WOULD LIKE TO SEE A LIST OF THE]
PR [PICTURES ON THIS DISK, TYPE D.]
MAKE "NAME FIRST REQUEST
IF :NAME = "D THEN DIRECTORY STOP
IF WORD? :NAME THEN LOAD.PIC :NAME ELSE L
END

```

```

TO DIRECTORY
NODRAW
PR [THIS IS A LIST OF ALL OF THE FILES]
PR [ON THIS DISK.]
PR [THE PICTURE FILES END IN .PICT]
PR []
CATALOG
PR []
IGNORE RC
END

```

```

TO IGNORE :CHARACTER
END

```

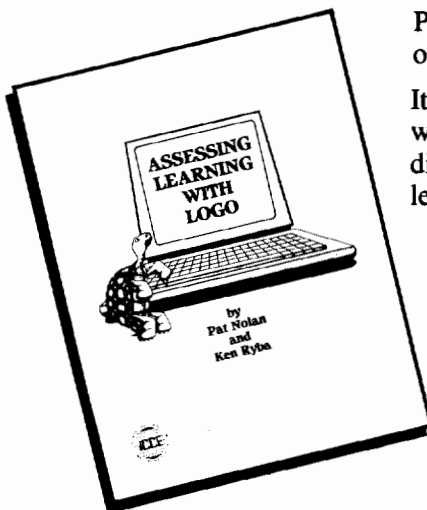
```

TO SAVE.PIC :NAME
HT
FULLSCREEN
MAKE (WORD :NAME "POS) POS
MAKE (WORD :NAME "HEADING) HEADING
SAVEPICT :NAME
SAVE :NAME
ST
CLEARTEXT
SPLITSCREEN
END

TO LOAD.PIC :NAME
LOAD :NAME
LOADPICT :NAME
PU
SETPOS THING (WORD :NAME "POS)
PD
SETHEADING THING (WORD :NAME "HEADING)
ST CLEARTEXT SPLITSCREEN
END

```

Assessing Learning With Logo



Presents the method for *Assessing Learning With Logo* at the levels of basic Turtle commands, repeats and procedures.

It contains all the necessary materials—checklists, assessment worksheets and activities—for developing coding, exploration, prediction, analysis and planning, creativity, and debugging at each level of learning Logo.

The methods and activities have been especially designed to highlight the role of the educator as a “facilitator of learning.” In this role educators guide students to reflect on their own thinking as they come into contact with powerful ideas at the beginning levels of Logo.

Single copies are \$12.50 plus \$2.50 shipping. Call now for a free catalog of ICCE publications.



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

Pen Points

Logo Book Reviews with ASTROLUG

by Ginny Nyahay

A Nudge is a small push. It's just enough to get you started on an exiting adventure. This book of Nudges is intended to give you a small push to get you started on a long series of adventures with Apple Logo." (p.iii)

Hi! My name is Ginny Nyahay and I am an Elementary School Computer Teacher. I love Logo. I find it such an exciting language to teach children. They really seem to enjoy the challenge of each Logo adventure.

I enjoyed *Nudges* very much. It is a workbook which has ninety-three, open-ended activities. *Nudges* was written by Glen Bull, Steve Tipps, and Timothy R. Riordon with assistance from Sharon Burrowes, Tom Lough, Jerry Short, and Joyce Tobias and published in 1985 by Holt, Rinehart and Winston. It was written for Apple Logo and Apple Logo II but is easily adaptable to Terrapin Logo and LogoWriter. (There is also an IBM version of *Nudges*, published in 1984.)

Directed and Open-Ended Activities

The book is set up as a consumable workbook. A concept is presented and there are open-ended ideas that follow. I especially like the idea of a directed activity in the beginning because it gives the students a definite place to start. The open-ended idea allows the student to start exploring, then, watch out world!

Nudges 1 - 40 focus on introductory Logo commands. At the beginning of each part, there is an explanation of the commands that will be introduced. This is a real strength of the book because you could scan the beginning section to see what commands are presented. As you go through each part, the new Logo concepts are highlighted which is great for the teacher as well as the student.

I enjoyed using Nudges 7 and 8 which introduce the Repeat command. The students try to guess what the commands will draw and then test their hypothesis on the computer. The activity would lend itself to showing the child a picture and having them create the Logo commands to go with it. I really think the children enjoyed this a great deal. Often, they came up with shapes that were totally different from any in the book. Wow, that was fun! The most exciting part was that we named the shape after the child. Can you imagine having a shape names after you? Well, *you're* not 9 years old either!!!

After creating a basic shape, Nudge 12 is challenging and fun. A child is asked to create figures based on the shape he started with. For example, after creating a square, a child is asked to create a row of squares or a set of stairs.

In Nudge 20, a picture is presented, and the student is asked to take the shapes he already knows how to create and to put them together in a procedure. I have found that in my experience, that

this seems difficult for the children in the beginning, but once accomplished makes debugging so much easier.

Art and Music Adventure

Nudges 41 - 93 contain more adventurous activities with art, music, words and lists and finally game creation. I enjoyed looking over the activities. The activity where students are asked to create a keyboard, and then have the turtle play notes using the keyboard is super! I can't wait to try that with my students this year.

Those of you who really enjoy a challenge will enjoy Nudges 83 - 88. They contain ideas for creating games using Logo. One game that looked particularly exciting was turtle flipper. In this game the computer could "flip a coin" using a random command. The activities are explained in such a clear and concise manner that even if you are not fully skilled in Logo, you can still understand them. These would be terrific to use with students! They would be exposed to more advanced concepts and also enjoy the activity.

I would recommend *Nudges* for beginning Logo Teachers as well as those a little more experienced in teaching Logo. The material seems easily adaptable for students in Kindergarten through grade eight. Teachers with advanced knowledge of Logo might not find enough challenging activities in the book. I think the book is well written and easy to follow. Keywords are bold-faced, which really helps to emphasize them. The format is good with an introduction and then a drill and practice activity followed by an open-ended activity. As a teacher, you can follow the format of the book exactly or select ideas that you like from different sections. There is even a teacher's manual available to assist you in your use of *Nudges*.

Two Pen Ups

Although I am the only person writing this review, I would give this book two pen ups. (One from me and one from the students who tried some of these activities.) I think it is an excellent source of Logo concepts as well as creative ideas. I especially like the idea that it isn't extremely wordy. As a teacher, you don't have time to look through thirty pages of text to get to the "teachable parts." It's readable style and organization make it readily adaptable to lesson planning.

Take some time and browse through *Nudges*. I am sure you will find many exciting adventures you would like to take. ASTROLUG will be back next month with a review of another Logo book. Until then, keep the turtles happy and keep reading.



Book of the Month

Each month a tool from a different Logo book will be highlighted. If we were only permitted one Logo book for our bookshelf, it would be one by Dan and Molly Watt. The question is, which one? One candidate would be *Learning with Logo*, written by Dan Watt (McGraw-Hill, 1983). In the end our choice would be *Teaching with Logo*. This book has an excellent definition of "tool":

Tool procedures are provided to help someone carry out a task that he might not otherwise be able to accomplish. A tool procedure can be used as if it is a Logo primitive. The person using it has to know how to use it, but he doesn't have to understand how it works. (Molly Watt and Daniel Watt, *Teaching with Logo*. Menlo Park, California: Addison-Wesley Publishing Company, 1986, p. 313.)

One Logo tool from this book causes the computer to wait until the user presses RETURN before going on. For example, if you wish to display two screens of text, the WAITFORUSER procedure causes the computer to wait until the user has read the first screen of text. The superprocedure might look like this:

```
TO SHOW.INSTRUCTIONS
PRINT.SCREEN1
WAITFORUSER
PRINT.SCREEN2
END
```

The procedures PRINT.SCREEN1 and PRINT.SCREEN2 each print a screen of text. The WAITFORUSER procedure causes the computer to pause between two screens.

LCSI Version of WAITFORUSER:

```
TO WAITFORUSER
PRINT [PLEASE PRESS RETURN]
PRINT READLIST
END
```

MIT Logo Version of WAITFORUSER:

```
TO WAITFORUSER
PRINT [PLEASE PRESS RETURN]
PRINT REQUEST
END
```

There are many variants of this tool. For example, we frequently use a similar procedure that requests the user to press the spacebar to proceed.

Using Apple Logo II and the Grappler

Have you been trying to figure out how to print out procedures and graphics from Apple Logo II using a Grappler interface card with a parallel printer? This problem was addressed in the September issue of *Open-Apple*, an excellent newsletter dealing with the more technical side of Apple products. For subscription information write to Tom Weishaar, Editor, *Open-Apple*, P.O. Box 7651, Overland Park, KS 66207.

Weishaar and Dennis Doms, his technical consultant, noted that the Grappler card expects a Control-Y character (instead of the "usual" Control-I) when the Pascal interface of Apple Logo is used. Thus if you send CHAR 25 (Control-Y) to the Grappler card, it will activate the printer. Then, you can use the other Grappler code letters to specify your printout. [e.g., G for graphic, D for double size, R for rotated, etc.]

They suggested a procedure similar to the following as a starting point for printing a graphic double size and rotated.

```
TO PRINT.PIC
MAKE "SLOT 1
OPEN :SLOT
SETWRITE :SLOT
TYPE CHAR 25
PRINT "GDR
CLOSE :SLOT
SETWRITE []
END
```

—from Tom Lough

Testudinal Testimony

by
Douglas H. Clements

The Atlanta–Emory Logo Project

Some Initial Groundwork: One Name (Mathematics), Two Subjects

In the words of Richard Skemp there are "...two effectively different subjects being taught under the same name 'mathematics'" (1976, p. 22). One subject, instrumental mathematics, consists of a limited number of "rules without reasons." The other, relational mathematics, is "knowing both what to do and why." It involves building up conceptual structures from which a learner can produce an unlimited number of rules to fit an unlimited set of situations. As Skemp continues, "... what constitutes mathematics is not the subject matter, but a particular kind of knowledge about it" (p. 26). From this perspective, the current mathematics curriculum is deficient because it neglects relational understanding. There is too much emphasis on instrumental understanding—formal symbolism and naming—and not enough on analysis, synthesis, problem solving—on *meaning*.

Mathematics to Skemp is a system of concepts that becomes organized at increasingly higher levels of abstractions. To learn these concepts, students need examples. Such concepts, once learned, serve as meaningful mental models for higher-level concepts. Rote, or instrumental learning, actually blocks later learning because students do not build the necessary mental structures that support higher-level concepts.

Logo and SOLO

John Olive (1986) based his early work with Logo on several beliefs, consistent with Skemp's work:

- That which is learned instrumentally has little probability of relating to other knowledge. The traditional curriculum may not provide experiences in which geometric objects and properties can be put into relationships.
- The turtle provides mental models for those geometric relationships which cause problems in the traditional curriculum, such as parallelism, perpendicularity, proportion, and angular measure.
- The turtle is a transitional object. It bridges experiences involving the body and other concrete objects with those of more abstract mathematics.
- Geometry in Logo can be manipulated. Objects such as line segments, rotations, and figures can become the products of the students' own investigations; they can be acted upon by the children. Children can be led to reflect upon this activity.

- Thus, certain Logo environments have the ingredients for building relational understanding: personal involvement in the learning experience, linkage between the experience and extant schemata, and reflection on the experience and its meaning.

Olive worked with ninth graders, two hours per day, three days per week, for six weeks (for additional information, see Barbara Elias' April 1986 column). He used a guided discovery approach. This approach supplies guidance and a certain amount of sequencing to build on existing knowledge, together with many opportunities for students to explore their own ideas. He made two major conclusions. First, progression through a sequence of activities should be based on students' individual levels of learning. The SOLO (Structure of Observed Learning Outcomes; Biggs & Collis, 1982) taxonomy provides one possible guide for determining such levels. They gauge how students approach the use of objects. These objects, in the case of Logo, might be Logo primitives, defined procedures without and with inputs, or recursive procedures. The SOLO levels are:

- Prestructural:** The object is not used when it is available and could be used, or it is used inappropriately or incorrectly.
- Unistructural:** The object is used by itself. Immediate feedback is required before any other commands can be used.
- Multistructural:** Objects are used in combination, but they are not related correctly with respect to the task.
- Relating:** The objects are related in terms of the nature of the task and the structure of the objects.
- Extended abstract:** Objects are related to create a new object which is more generalized than its parts, or a generalized procedure is used effectively to create specific objects which help accomplish the task.

Second, for many students, the sequence of lessons was too fast. There was not enough time for them to explore new programming ideas and geometric relations before new ones were introduced. This rushing may have resulted in students regressing to previous solution strategies and to a denigration of understanding. Their understanding of Logo and of geometry was generally instrumental. However, those students for whom the (relatively rapid) pace used in the study was appropriate did progress through the SOLO learning cycles. They demonstrated a shift to more abstract mode of functioning with Logo and relational understanding of many of the geometric concepts that were introduced.

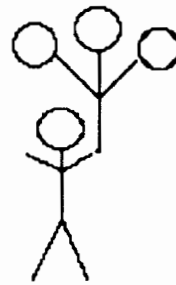
The Atlanta-Emory Logo Project (finally)

This brings us to the Atlanta-Emory Logo Project (Olive, Lankenau, & Scally, 1986). During a two-year period, several mathematics teachers from two urban high schools were given training in using Logo to teach geometric relationships. Two ninth-grade Logo classes in two schools became the main focus for the project research. These classes were taught for 16 weeks. Curriculum development was an on-going process extending over 1 1/2

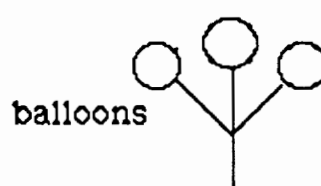
years of the project, but was not standardized. Rather, guidelines and materials which the teachers could use in their own way were offered. Both classes attempted to start from students' intuitive knowledge of motion, direction, angles, and geometric shapes. They then used Logo to build on these beginnings. To give the flavor of this development, two lessons will be described. As a brief first example, a Balloon Man was designed to help students create a top-down plan (see Fig. 1)

TOP-DOWN PLAN FOR CREATING A BALLOON MAN

GOAL: a balloon man



MAJOR PARTS:



SUB_PARTS:

single balloon



circle



lines



Bob Jensen & John Olive, Emory University

Figure 1

The second example is actually a sequence of lessons. In the pilot curriculum, gaps were found where variable inputs to procedures were introduced. The following sequence was designed to fill the gaps and help develop a relational understanding of both the use of variables in Logo procedure and the mathematical relationships which emerged from the investigation of complex polygons. The sequence begins at a point where students are comfortable with the definition of non-variable procedures for generating polygons and

with the use of REPEAT.

First, students define several regular polygons using REPEAT. Second, they complete a table to relate the number of repeats to the angle turn (i.e., the "Rule of 360"). Third, they define a series of square and triangle procedures that create different-sized shapes (e.g., SQ.20 SQ.30...). Fourth, they compare each procedure to discover what is changing in each (the input to the FD commands)

and what remains unchanged (everything else). The teacher introduces the syntax for procedures with inputs, and students generate variable procedures for their different shapes. Fifth, students discuss similarities and differences among the procedures for different polygons and the rule of 360. This leads to a more generalized form, such as:

```
TO HEX :SIZE
  REPEAT 6 [FD :SIZE RT (360 / 6)]
END
```

The only difference now is the number of repeats. Discussion elicits the idea of using a variable. Students generate a shape changing procedure (note that it is of a fixed size at this point):

```
TO FIXED.POLY :N
  REPEAT :N [FD 50 RT (360 / :N)]
END
```

Sixth, students combine the two variables:

```
TO REGULAR.POLY :N :S
  REPEAT :N [FD :S RT (360 / :N)]
END
```

Seventh, students discuss that REGULAR.POLY always produces a simple, closed, regular polygon. What would happen if we could input any angle?

```
TO APOLY :N :S :A
  REPEAT :N [FD :S RT :A]
END
```

Students explore APOLY, which can lead to an investigation of star polygons and the creation of a procedure which embodies this relationship.

```
TO MPOLY :N :S :M
  REPEAT :N :S RT (:M * 360 / :N)]
END
```

MPOLY produces a N-pointed star when M and N have no common factor and M is greater than one and less than (N-1).

Research Results

Much information on the students was collected. The first type of information was dribble files—a record of all text that is printed on the screen as the students works with Logo. Studying these files can be fascinating; it is unfortunate that space limitations prevent a presentation here. The dribble files of students' work on a final exam project were analyzed using the SOLO taxonomy. Analyses indicated that most students at school 1 worked at the Relating level at some point. Three students appeared to reach an Extended Abstract level when defining variable procedures. Programming problems included misuse of variables and misunderstanding of error messages.

Students at school 2 were more diverse. Three worked mostly at unistructural and multistructural levels with some evidence of prestructural use of variable procedures. Three were mostly at Multistructural and Relating levels, two appeared to reach an

Extended Abstract level and one worked mainly at a Multistructural SOLO level. They had similar problems as the students in school 1, but in addition two students did not define requisite procedures and two others did not use previously defined procedures as building blocks.

The most striking difference was that students from school 1 seemed to be generally more systematic in their approach to the problem and appeared to follow the directions more closely. They tended to work at the Multistructural or Relating levels.

This difference may be due to a particular instructional activity which occurred at school 2 two weeks prior to the final exam activity. The teacher presented several complex geometric ideas toward the end of the semester, without enough time for students to explore each new idea at their own level before the next one was introduced. In one activity, students typed in procedures that utilized complex trigonometric formulas that the students did not understand. This rote, or instrumental, presentation of new procedures may have contributed to their instrumental approach on the test activity. This reaffirms Olive's original finding of rote learning in students who are rushed.

Another thrust of the project is to evaluate changes in students' thinking from a framework of the van Hiele's. As discussed in previous columns, the van Hieles believe that students pass through levels of geometric thinking. The first three levels as conceptualized by the project are:

Visual: The student sees figures as entities and is unable to discern their parts.

Descriptive: The student describes the properties and relations of figures and operates upon them, but in a case specific manner only. He or she can use these properties to form definitions of the figures, but cannot yet identify class inclusions (e.g., that a square is a rectangle).

Theoretical: The student generalizes the logical relations which exist among figures and their parts and operates on those relations deductively.

Susan Scally designed interviews to assess students' level of thinking. Many of the questions dealt with the topic of angles (no strange topic to us!). Students were asked to draw an angle, then draw others, each of which is different in some way from all preceding ones. On the pre-interview, all students included irrelevant attributes in their responses. On the post-interview, three of five Logo students included no irrelevant attributes; one of three comparison students included no irrelevant attributes. As we saw in this column in previous months, some students think orientation and length of represented rays are aspects of the angle. Several students use a "protractor model" to estimate angle size. They select the scale which reads from 0° to 180° from left to right. For them, a 60° angle oriented to the right measures 120°. One mentally imposed a protractor on top of the angle and assigned a value to each ray according to its location on the left to right scale. She then labeled her angles with the resulting ordered pair.

All eight students defined an angle as two lines that meet. Some said, "at a point (or vertex)." Most also correctly identified classic representations of angles, but not exterior angles. Most marked two intersecting curved lines as angles. Overall, there was little change from pre- to post-interview.