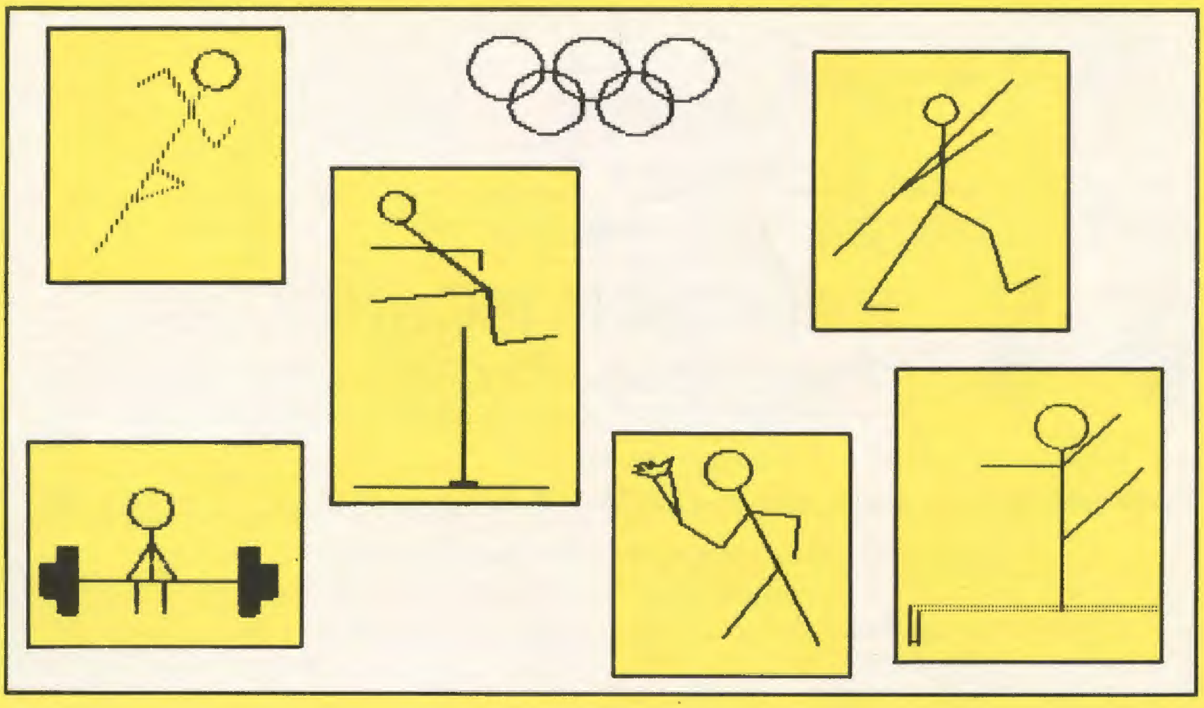
Journal of the ICCE Special Interest Group for Logo-Using Educators

LOGO EXCHANGE

OCTOBER 1988 VOLUME 7 NUMBER 2



International Council for Computers in Education

ICCE Publications

LOGO EXCHANGE

VOLUME 7 NUMBER 2 Journal of the ICCE Special Interest Group for Logo-Using Educators October 1988

Publisher
International Council for Computers in Education

Logo Exchange is the journal of the International Council for computers in Education Special Interest Goup for Logo-using Educators (SIGLogo), 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, UofO, 1787 Agate St., Eugene, OR 97403. Second-class postage paid at Eugene, OR. USPS #000-554.

CONTENTS

ICCE Membership (*includes The Computing Teacher)*

	U.S.	NON-U.S.
1 year	28.50	31.00

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. Add $18.00 for airmail shipping. VISA and Mastercard accepted.

The *Logo Exchange* is produced on a Laserwriter and Macintosh SE donated by Apple Computer, Inc.

From the Editor

Where Are We Going?

I have just spent the better part of a week teaching at Dan and Molly Watt's Logo Institute held this year at Lesley College in Cambridge, Massachusetts. As always, the atmosphere at the Institute was alive with excitement as novices and experts worked together in a rich Logo environment to expand their horizons. In (rare) spare moments, I found myself reflecting on the evolution of Logo over the past several years.

It seems like only yesterday that Logo burst unexpectedly onto the educational scene. We found ourselves reading *Mindstorms* and dreaming of a classroom of tomorrow where Logo was the panacea that would single handedly improve our educational system. Some eight years later, the educational system has not been revolutionized and educators are still looking for solutions.

But something more subtle than a revolution has happened. As hundreds of educators jumped on the Logo "bandwagon," thousands upon thousands of elementary students were exposed to Logo. Many of these young people are now in secondary schools, bringing with them a foundation in Logo programming. A substantial number of these students have enough knowledge of Logo to be able to make use of this skill with very little re-teaching.

Last year, as part of my work with Logo Computer Systems, I attended computers in education conferences all over the United States. As I talked to people in workshops, conference sessions, and over coffee, I gradually became aware of an evolution in attitude towards the use of Logo. No longer is it viewed solely as a "little kids" language. There is a more accepting attitude towards the use of Logo in junior and senior high schools. At almost every conference, I encountered enthusiastic teachers exploring the use of Logo in their junior high, high school and university classrooms.

A part of this growing acceptance of Logo at levels beyond the elementary seems to be the increasing sophistication of newer implementations of Logo. Versions of Logo for the Macintosh open up new horizons for those interested specifically in programming. Such versions offer access to more memory, greater speed, and features such as object oriented programming. Other newer versions such as Logo Writer and Logo Plus, broaden the capabilities of Logo for use in a variety of subject areas. It is now much easier to move beyond turtle graphics. Text and graphics can be combined with relative ease.

In fact, these new Logos seem to be moving more towards becoming broad based productivity tools. No longer is Logo "just" a programming language. The new Logos provide a low level desk top publishing environment, often quite inexpensively. They also have characteristics reminiscent of Hypercard (Lough, Muir). They are flexible and easy to use but there is great power in the underlying Logo programming language.

Another factor affecting the wider acceptance of Logo is the publication of a number of books that present Logo on a more sophisticated level. For example, Brian Harvey's *Computer Science Logo Style* series make it very clear that Logo isn't just for little kids. Paul Goldenberg and Wally Feurzeig's eagerly anticipated book, *Exploring Language with Logo*, allows us to play with grammar. There are certainly many other books both published and promised that address uses of Logo beyond the basics.

These observations, indicate that there are three important factors affecting the future of Logo, especially beyond the elementary level. The first is the growing number of students who have been exposed to Logo at some point in their educational careers. The second is the development of increasingly sophisticated versions of Logo on increasingly sophisticated computer systems. Finally, there is the availability of materials that can help guide us in more advanced uses of Logo. Only time will tell the effect of these two factors on the future, but perhaps we can speculate.

Is it reasonable to assume that Logo will increasingly become the vehicle through which all children will learn about computer programming? Will powerful ideas such as procedurality and debugging be seen as important enough to be included in our school curriculum? If so, is not Logo a logical tool to teach these concepts? Will skill with Logo be encouraged and nurtured so that as students move through school they become more and more proficient with some version of "HyperLogo?" Will secondary teachers be able to assume such knowledge in students? Will materials that make use of Logo in a variety of curriculum areas at all levels be published and then used in schools? Will Logo be built into the "dynabook" computer of the future that will be sitting on every student's desk? Or will the future of Logo take an unexpected turn as new software and hardware come into the picture? What will be the next addition to Glen and Gina Bull's Journal of Learner Based Tools? (Bull) And, what will be causing excitement at Dan and Molly's next Logo Institute? Stay tuned! The future is unlikely to be boring!

References

Bull, Glen. (1988) Logo Reflections. *Logo Exchange 6(9), 19.*

Goldenberg, E. Paul and Feurzeig, Wallace. (1987) *Exploring Language with Logo.* Cambridge, Massachusetts: MIT Press.

Harvey, Brian. (1985, 1986, 1987). *Computer Science Logo Style, Vol. 1 - 3.* Cambridge, Massachusetts: MIT Press.

Lough, Tom (1988). Hyperhype. *Logo Exchange 6(6)*, 2.

Lough, Tom (1988). Hyperhype II. *Logo Exchange 6(7)*, 2.

Lough, Tom (1988). Hyperhype III. *Logo Exchange 6(8)*, 3.

Muir, Robs (1988) PartialLogo on a HyperCard. *Logo Exchange 6(7) 25-26*

Sharon Burrowes Yoder
ICCE, 1787 Agate Street, Eugene, Oregon 97403
CIS: 73007,1645 BitNet: ICCE@OREGON

Monthly Musings

Incomplete Perceptions
by Tom Lough

Remember the story about the three blind people who had encountered various parts of an elephant? One was touching the trunk, another a leg, and the third the tail. Each had an idea about the nature of the beast which was quite different from those of the others!

Certainly one cannot fault the blind people for not recognizing the true nature of the elephant. Their perception of the elephant was incomplete because they were operating with information which was incomplete. Moreover, their perception is likely to remain incomplete, since no one knows everything there is to know about an elephant.

However, they could have done something to reduce the "degree of incompleteness" of their information. For example, they could have moved about and explored other parts of the elephant. They could have interviewed each other in detail to discover common attributes. Actions such as these would have provided additional information, thus helping them to gain a broader perspective of the animal.

Sometimes I am like the blind people. I find myself guilty of acting on incomplete information when, with a little additional effort, I could have obtained access to much more. Even worse, I feel that the information I have is complete (for my purposes) when in reality it usually is not so.

This same mechanism seems to be at work when people are forming their personal answers to the question, "What is Logo?" When teachers are not familiar with the full range of Logo's capabilities, it is easy for them to dismiss it as a computer language just for younger children.

This realization came home to me recently while delivering a Logo workshop to high school physics teachers. During the workshop introduction, I helped the teachers learn more about the artificial intelligence heritage of Logo, and its relationship to Lisp, a powerful and widely used artificial intelligence computer language. Activities during the workshop focused on using Logo in activities directly related to concepts of physics, such as planar motion and electrostatics.

At the beginning of the workshop, the physics teachers appeared mildly interested, since I had indicated that I had used Logo in my physics teaching regularly since 1983. As they proceeded through the activities of the workshop, however, their enthusiasm grew remarkably. By the end of the scheduled time, many teachers were exclaiming that they had no idea that Logo could do all the things they had done with it, that they did not know it was so powerful, yet so easy to use. They were surprised at the many physics applications, and began generating ideas for applications in other high school subjects they taught.

The foundation for this surprise was their previous incomplete knowledge of Logo. They said that they had always thought of Logo as a computer language just for the elementary grade students, and had no idea that it would be so useful for secondary students.

But let's go beyond the issue of using Logo in secondary schools and think about how parents, administrators, and teachers at all grade levels may have already built their opinions about Logo on incomplete information. Introductory books, one-day workshops, articles about second graders drawing squares all contribute to such an opinion. From what I can sense, this is the situation in many parts of the globe.

How can we encourage these people to take a more complete look at Logo, to get more of "the whole truth" about this fascinating powerful computer language? What can each of us do to make sure others have more complete information about what Logo is and what Logo can do?

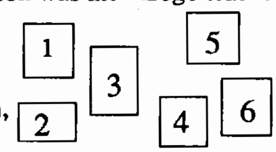
I welcome your ideas.

FD 100!

PS: Have you seen any automobile license plates with Logo slogans on them? I am currently collecting such slogans and will incorporate them into a future article. Please send me any that you know about.

Tom Lough
PO Box 5341
Charlottesville, VA 22903

Cover: These Olympic Logo projects were done by Grade Six and Seven students at Pineview Elementary School in Prince George, B.C. Canada. They used Terrapin Logo to complete the projects. Jim Swanson was their Logo teacher.



1. Jamie Hill, grade 6
2. Jamie Hill, grade 6
3. Ryan Martin & Shane Myram, grade 7
4. Jasen Florell, Jason Neumeyer & Faron Muncey, grade 7
5. Scott Ross & Chris Bowler, grade 7
6. Colleen Wanner & Daniela Mogus, grade 7

SIGLogo News

The first general meeting of the **ICCE Special Interest Group for Logo-Using Educators (SIGLogo)** was held in the Batike B Room of the Lowes Anatole Hotel, during the meetings of the National Educational Computing Conference, on Wednesday, 15 June 1988.

The acting officers were in all in attendance: Peter Rawitsch, President; Gary Stager, Vice President; and Theodore Norton, Communications Officer; so was Sharon Yoder, Editor of the *Logo Exchange*. Additionally, thirty-six persons signed the attendance sheet, although the actual number of persons present is likely to have been larger.

The agenda for the meeting was divided into three parts: (1) Welcome by Peter Rawitsch, and discussion of the proposed SIGLogo bylaws. (2) Old Business, including a reading of the membership and financial reports (Theodore Norton); and a report on the *Logo Exchange* (Sharon Yoder). (3) New Business: A discussion of possible projects for 1988-89, prepared and led by Gary Stager.

1. Peter Rawitsch opened the meeting by greeting those present. He introduced the acting members of the SIGLogo Board of Directors (see above). He then submitted the draft copy of the SIGLogo bylaws to the consideration of those present. There followed a lively discussion of the bylaws. Several changes were moved and adopted:

In Article I—Name: following the second sentence, insert the sentence: "The official publication of the special interest group will be the *Logo Exchange*."

In Article II— Purpose: In the first sentence, insert the word "legitimate" just before the word "...language..."

In re Article IV—Officers; Section 1: Election of Officers: Peter Rawitsch proposed that the four officers (President, Vice President, Communications Officer, Treasurer) be elected to two year terms. In re subsection a: Nominations will be submitted by January 1. In re subsection b: the membership will be balloted by April 1. Members will vote by May 30. Changes accepted.

At this point in the meeting, Peter Rawitsch asked for nominees for the vacant post of Treasurer. Frank Matthews, of the University of Huston, Texas, volunteered, and was approved by acclamation. (Frank can be reached at 1880 White Oak Drive #187, Houston, Texas 77009, 713-864-0768)

2. Theodore Norton read the "SIG Membership and Organization Budget" for 1988. This covered both membership and finances. Current membership in SIGLogo is estimated at 2,853 making SIGLogo is now the largest SIG in the ICCE.

Sharon Yoder reported on her editorship of the *Logo Exchange*. She commented on her assumption of editorial responsibilities, after *LX* was acquired by the ICCE. She noted changes on the journal's staff for the next year: the departure of Robs Muir, and the additons of Lesley Thyberg and Dan Watt. She indicated that *LX* would start the new academic year on schedule. She stressed that *LX* has an ongoing need for cover art as well as articles. Tom Lough congratulated Sharon for a difficult job well done. His thanks were warmly seconded by those present.

3. Gary Stager led the last part of the meeting, on the following preannounced topics: conference co-sponsorship; children's conference; Logo Learning Month; electronic bulletin board; position papers; other ideas from SIGLogo members. The following summary covers some highlights of this discussion.

Gary Stager discussed the founding of the SIGLogo, the value of ICCE affiliation, and the recent Great Lakes Logo meeting. At the latter, participants discussed regional chapters and national conferences—do we need them? Gary thought so. He is also involved in the planning of NECC '89

Suggestions for children's computer conference were put forward. Should it be a Logo conference, or a computer conference sponsored by SIGLogo? The possibility of children's telecommunications between conferences was addressed.

Tom Lough observed that there were no future plans for another MIT conference.

Several of those present spoke to the issue of telecommunications. Ricky Carter proposed the creation of a subcommittee on telecommunications. Gary Stager raised the issue of Logo bulletin boards. Tom Lough volunteered to chair the subcommittee.

Other topics that were raised included: Sponsorship of a Logo Learning Month; creation of a Logo Room; the need for a general discussion on how Logo is to be taught; and a Logo Idea Exchange.

Sharon Yoder raised the possibility of a special Conference Issue of *LX*, much like the conference issue of *The Computing Teacher*. Tom Lough observed that work needs to go forward on a Logo Language Standard.

Dan Watt spoke to his concern with microworld development and exchange. Should we set up a Board of Editors for microworlds? He emphasized the importance of documentation standards. Sharon Yoder affirmed that ICCE would be interested in hearing more about these proposals.

International Logo activities were discussed, together with various Logo trips abroad, international communications, and a Logo Academy.

The meeting was brought to an informal close.

Theodore Norton, Communications Officer
3 Pine Tree Circle, Pelham, MA 01002

Logo Ideas

A Logo Message Center
by Eadie Adamson

One of the first events of the school year for many of us is something called "Parents' Night" or "Parent Evening" or "Open House." Last year I wanted to do something to encourage parents to visit the new computer lab in our school. Since the evening is planned as a method of communication between parents and teachers about the curriculum in general, I thought that a little experiment in communication using LogoWriter might be fun. I invited the parents to come to the lab and leave a "secret" message on the computer for their child, a message which would be printed and delivered as a surprise the next day. It seemed a perfect way to show off LogoWriter.

The program itself (see listing below) was extremely simple. Since I had a clear idea of what I wanted it to do, it took a very short time to write. Starting instructions are given on the screen. The program is invoked by typing WRITE and pressing Return. Then the program guides the user through the stages of leaving a message. When completed, the message is stored on a separate page so that each child can have several messages. Finally, the program returns to the main message page, ready for the next correspondent.

Since this particular Parents' Night involved only fourth, fifth and sixth grade parents I set up one computer for each grade so that the messages would not have to be sorted. I was not prepared for the huge success of this idea! Our lab is directly across from the library where the main reception was being held. When I arrived in the lab, the room was already jammed with parents waiting their turn to leave a message. More people were in the lab than in the library! They were fascinated by the idea of leaving a message and had turned the room into a mini-version of a Logo class, with the "experts" helping out those having difficulty. People were having a wonderful time sharing their momentary successes. I sat down with one very nervous father who confessed to me that this was the first time he had even touched a computer! Parents discovered the importance of following directions carefully, since there was no elaborate error-trapping included. I was amused to hear some accusing the computer or the program of being "stupid" when something went wrong, not unlike the reaction of many beginning students!

I have refined the MESSAGES program slightly since last year by breaking one long procedure into a more readable series of shorter procedures. This year I will have more computers running the program so that there is less time to wait. Last year several parents came back a number of times before they had a chance to leave their message. In addition, I have worked out a second procedure which prints all the message pages on the disk automatically. Since we are using French LogoWriter with the sixth grade, I will probably set up a French message center also.

The MESSAGES idea could transfer to classroom use or could be used as a mini-lesson in writing and communication. One class could write messages for another class. The classes could also retrieve their own messages and write replies. This could serve as preparation for an online penpal project.

Technical notes on the MESSAGE CENTER:

Since we use LogoWriter 2.0 for the Apple, I took advantage of being able to LOCK the message page itself. If you don't yet have this version, you can write a STARTUP procedure which prints a welcome and beginning instructions on the screen. Any procedure named STARTUP on a LogoWriter page will be automatically run as soon as the page is accessed, so your instructions will appear each time someone goes to the MESSAGES page.

Here is the initial screen I set up on my MESSAGES page:

```
locked---------Messages----------------
This is the LogoWriter message center.
To leave a message:
      Type WRITE.  Press RETURN.
FOLLOW DIRECTIONS AT THE BOTTOM OF THE
SCREEN CAREFULLY!

LogoWriter will collect your message
and your name.   When you finish you'll
see LogoWriter save your message and
then return to this page.
```

When someone types WRITE, GETNAME, the first subprocedure, asks for the first and last name of the person to be sent the message. The last name will later be used as the page name for storing the student's messages.

GETMESSAGE does several things: clears the screen (remember it's a locked page, so any changes will not be saved on the page), moves to the top of the page, begins the letter with "Dear" and the first name, moves the cursor down, tabs and waits for the person to type the message. When

Logo Ideas -- continued

Return is pressed, the person is asked to give the name to be signed on the letter. "Love," and then the person's name is printed on the page.

SAVEMESSAGE clears the text from the screen so that it appears to the user that the message is being moved to another page. Actually all the information has been stored in the variables NAME, MESSAGE, and FROM. SAVEMESSAGE then checks whether a message has already been left for that person:

```
IFELSE MEMBER? LAST :NAME PAGELIST
```

If so, it gets that page which was named with the last name.

```
GETPAGE LAST :NAME
```

If this is a new person, SAVEMESSAGE gets a new page and names it with the last name.

```
NEWPAGE NAMEPAGE LAST :NAME
```

I didn't need to check for duplicate last names, since I used separate computers for each class. It might be a good challenge for **you** to add that feature!

SAVEMESSAGE hides the turtle, then moves to the bottom of the page and prints a blank line. This ensures that the messages will be saved in order. If a message already exists, the second message goes under the first, at the bottom of the page. On a new page the bottom of thc page is the same as the top since there is no text on the new page. SAVEMESSAGE then uses the information collected to put a letter on the page, leaves a blank line and a few stars as end markers, and moves back to the MESSAGES page for the next writer.

Printing the Messages

Printing all those messages could be a long and tedious task. How could one procedure be instructed to move from page to page, printing each page as it moves along? Recursion doesn't seem to help us much here. One's first thought might be to use multiple GETPAGE commands, followed by SSPACE PRINTTEXT80. That will work, but there's an easier way (of course!)

Remember that if a procedure is to go from page to page, performing tasks as it goes along, all thc ncccssary information must be contained in that procedure or its subprocedures. If a subprocedure is called from the procedure, it must be present on the page or somewhere in memory as a tool procedure.

The problem then is this: How can you get a list of pages, run through them in order, print each page and then move to the next page, all from just one procedure? In addition, how can you eliminate the pages you don't want to print (e.g., the MESSAGES page, the PRINT page if you make it a separate page, the HELP page if you left that on the disk)? Think about the problem a bit before you read on!

I named my printing procedure PRT. Though short, it's a little complex because everything must be contained in this single procedure in order for it to move from page to page and continue running.

First, store the PAGELIST in the global variable, PAGES. Store the COUNT of the pages as HOWMANY.

```
NAME PAGELIST "PAGES
NAME COUNT :PAGES "HOWMANY
```

Next you need to repeat this process HOWMANY times to print all of the pages, but you want to check each page name **before** it is printed. If it's one you don't want, you want to eliminate it and move on to the next page. Which page are you concerned about each time? The FIRST of the list stored in PAGES.

Our REPEAT begins this way:

```
IF EQUAL? FIRST :PAGES "MESSAGES
   [NAME BUTFIRST :PAGES "PAGES]
```

This eliminates printing the MESSAGES page.

I stored my printing procedure on another page, which meant I had to be sure that page was not printed as well. You don't *need* to do that and in fact it is better not to, but I also wanted to be sure that everything worked before I combined things:

```
IF EQUAL? FIRST :PAGES "PRINT
   [NAME BUTFIRST :PAGES "PAGES]
```

Any other pages to be skipped need to be checked and eliminated in a similar fashion in this part of the PRT procedure.

One last check was necessary to eliminate an error message and is similar to the kind of stop rule needed in many recursive procedures. We know we are cutting out at least one page from the list of pages already counted. The list of pages will be empty before REPEAT :HOWMANY has finished. Here's what to say:

```
IF EQUAL? COUNT :PAGES 0 [STOP]
```

Now having passed all these tests on the first REPEAT :HOWMANY, my procedure needed to be told to get the page:

```
GETPAGE FIRST :PAGES
```

To make it easier to distribute messages later, I wanted to put the last name at the top of the page. The FIRST of the list in PAGES will be a last name, the name of the page, so the commands look like this:

```
TOP
PRINT FIRST :PAGES
PRINT []
```

Next, print!

```
SSPACE
PRINTTEXT80
```

I also needed to eliminate the page just printed from the list, so that the last part of REPEAT :HOWMANY reads:

```
NAME BUTFIRST :PAGES "PAGES
```

The REPEAT :HOWMANY line will continue until there are no more page names stored in PAGES.

I hope you have fun with this project too. Just remember, if you decide to try it at a Parents' Night, to be prepared to be swamped with people anxious to try it out. Have as many computers available as possible. Put signs on the computers to sort out the groups if necessary.

The MESSAGE CENTER procedures:

```
TO WRITE
GETNAME
GETMESSAGE
GETSIGNATURE
SAVEMESSAGE
END

TO GETNAME
CC
CT
TYPE [Please type the FIRST and LAST NAME of]
TYPE CHAR 13
TYPE [the person to receive this message.]
TYPE CHAR 13
TYPE [Press RETURN after the name.]
TYPE CHAR 13
NAME READLISTCC "NAME
END

TO GETMESSAGE
TYPE [Now type your message. DO NOT PRESS RETURN UNTIL
YOU ARE FINISHED!]
TYPE CHAR 13
PRINT (SENTENCE "Dear WORD FIRST :NAME ", )
PRINT "
TAB
NAME READLIST "MESSAGE
END

TO GETSIGNATURE
TYPE [Now type your name (Ex: Mom or Dad) and press
    RETURN.]
TYPE CHAR 13
PRINT "
REPEAT 3 [TAB]
PRINT "Love,\
PRINT "
REPEAT 3 [TAB]
NAME READLIST "FROM
END

TO SAVEMESSAGE
CC
CT
IFELSE MEMBER? LAST :NAME PAGELIST
    [GETPAGE LAST :NAME] [NEWPAGE NAMEPAGE LAST :NAME]
HT
BOTTOM
PRINT []
PRINT (SENTENCE "Dear WORD FIRST :NAME ",)
PRINT []
TAB PRINT :MESSAGE
PRINT []
REPEAT 3  [TAB]
INSERT "Love,
REPEAT 2 [PRINT []]
REPEAT 3 [TAB]
PRINT FIRST :FROM
PRINT []
REPEAT 3 [TAB]
PRINT [* * * * *]
CC
GETPAGE "MESSAGES
END
```

The Printing Procedure:

```
TO PRT
NAME PAGELIST "PAGES
NAME COUNT :PAGES "HOWMANY
REPEAT :HOWMANY
    [IF EQUAL? FIRST :PAGES "MESSAGES
            [NAME BUTFIRST :PAGES "PAGES]
      IF EQUAL? FIRST :PAGES "MESSAGE.BU
            [NAME BUTFIRST :PAGES "PAGES]
      IF EQUAL? FIRST :PAGES "PRINT
            [NAME BUTFIRST :PAGES "PAGES]
      IF EQUAL? COUNT :PAGES 0  [STOP]
      GETPAGE FIRST :PAGES
      TOP
      PRINT FIRST :PAGES
      PRINT []
      SSPACE
      PRINTTEXT80
      NAME BUTFIRST :PAGES "PAGES ]
END
```

Eadie Adamson
Allen Stevenson School
132 East 78th Street
New York, New York 10021

Stager's Stuff

A Microful of Monkeys
by Gary Stager

At the 1987 West Coast Logo Conference Seymour Papert shared a discussion he had at Project Headlight at Boston's Hennigan School. Dr. Papert and his associates mused at how unfortunate it was that the kids go home at 3 PM and the computers sit around alone all night with nothing to do. Since computers can't play baseball or watch cartoons, it doesn't seem fair that the students have to do homework and the computers get to goof off. What kind of homework can we devise to keep our school microcomputers off the street?

One such idea suggested by Dr. Papert was based on the familiar tale, "if there were an infinite number of monkeys, sitting at an infinite number of typewriters - they would eventually type the complete works of Shakespeare." This hypothesis would be nearly impossible to prove due to the low probability of Shakespeare's work being recreated at random. In fact, two of the goals of the Logo project featured in this article are to explore probability and randomness.

Using a few simple Logo procedures we are able to create a laboratory for experimenting with word formation, probability, randomness, spelling, and vocabulary. Analytic pursuits like science and math become inseparable from language. We will use the scientific method for our journey into the world of linguistics, the science of language. In an NLXUAL Challenges article by Robs Muir the author convincingly articulates this idea:

> What is science anyway?...Let me propose that science can be done in any part of the curriculum. Science isn't just a subject such as chemistry or physics; science is a method of inquiry that transcends curricular boundaries. It is just that we are unaccustomed to doing science in an English class. (Muir 1986)

In order to make our experimental data manageable we should severely restrict the goal of our experiment. Instead of waiting the centuries necessary to recreate Macbeth randomly, our goal will be for Logo to create the word "CAT" by random selection. Even the odds of creating the word "CAT" are 1 in 17,576 (26^3). If it takes only one second to randomly create a three-letter word, it could take five hours or more for the word "CAT" to appear. That's a lot of homework for our poor micro!

Our Laboratory

This entire project will run in any version of Logo. (Replace INSERT with TYPE and in Apple or IBM Logo, replace EQUAL? with EQUALP.)

Experiment # 1

The CREATE.WORDS procedure is our monkey, the beaker in which we stir the elements in our experiment.

```
TO CREATE.WORDS :COUNT
MAKE "NEW.WORD
    (WORD LETTER LETTER LETTER)
IF EQUAL? :NEW.WORD "CAT
    [PRINT SENTENCE :NEW.WORD :COUNT
     STOP]
INSERT WORD :NEW.WORD CHAR 32
CREATE.WORDS :COUNT + 1
END

TO LETTER
OUTPUT PICK [A B C D E F G H I J K L
             M N O P Q R S T U V W X Y Z]
END

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

The procedure takes a numerical input, usually the number 1, creates a random word from our collection of elements (letters) and checks to see if the word created is the word "CAT." If the word created is "cat", the user is told how many tries it took to create "CAT" and the procedure stops. If "CAT" is not formed, the word that was created is printed and the experiment continues.

LETTER randomly outputs one of the letters contained in the list. The list in LETTER represents our experimental pool. The PICK procedure has been used in numerous other Logo projects and articles. As you may remember, PICK takes a word or list as input and randomly outputs an item in that word or list.

<u>The Goal of Our Experiment</u>

The goal of our experiment is relatively simple -- to make the CREATE.WORDS procedure "smarter." This means two things.

1. The word CAT will be created in fewer tries.
2. More of the other words, while still randomly generated, will look less like gibberish and more English or English-like.

Throughout the remainder of this article I will suggest ways to modify the rules for creating new words and improve the sample of letters Logo chooses from. We will begin with the simplest of rules and pool of letters and incrementally make our experiment more personal and sophisticated.

Experiment #1: Getting Started

In this experiment, students should be encouraged to record the number of random words created and the number of English words they find in the results. Experiment #1 creates some pretty strange looking words. What was the last time you saw a word beginning with QZ? I found that my students and I were constantly looking up words in the dictionary that appeared to be orthographically[1] plausible. In the process we were able to improve our vocabulary.

How can we avoid awkward combinations of letters? In my results from Experiment #1, Logo created 80 words and 2 of them were English words. 2 out of 80 represents 2.5% of all words generated were recognizable English words (coincidentally, men's names.) What was your experimental result? What was your class average? Were your results close to the class average? Why not graph the class results? How about a contest to see who can find the most real words in their sample of 1000 words?

Type CREATE.WORDS 1 to begin the experiment. In this experiment three-letter words are randomly assembled by randomly chosen letters of the alphabet. The following is a sample of the results I achieved.

Results of Experiment #1

CDS	JSW	YLC	GDO	DLI	AZE	IKC	QNJ	WNA	PKC	BOJ
CEM	ZEJ	HJJ	WMR	WYI	JUG	EWB	JOI	ZOY	IVW	PVH
SEZ	FXR	DHB	BEF	QZU	AWW	ORQ	JSF	AHD	**DOM**	LYU
OOI	ZSG	NNI	YLP	OQI	PBV	MHM	IEV	RYO	KKJ	THH
GUW	UPX	AQB	MXI	DLG	BRE	TMN	YHT	EWM	BTN	WAI
AGR	PNL	RME	SBZ	POG	PCC	MCJ	MLO	QZQ	AXI	VCE
VIV	GLP	OZL	IDQ	WGR	VST	MLB	MEK	**DON**	MGV	WDE
BDE	PZG	CKR								

NOTE: Due to space limitations I have taken some liberties with including the results of my experiments. The English words, including proper nouns, generated are in bold; commonly recognized initials are in bold and italicized; abbreviations are underlined. Where I had to truncate my experimental results I tacked on some more of the other "real" words which were created.

Experiment #2: Vanna Speaks:
The Wheel of Fortune Theory

While exploring this microworld with some fifth graders I asked, "How can we improve the quality ("Englishness") of the words created by the CREATE.WORDS procedure without getting rid of the infrequently used letters, such as Q, X, Z?" As a hint I asked, "How do you improve your chances of winning the lottery?" Some kids suggested that you would buy more tickets. We can increase our chances of generating better words without removing any other letters from our sample by adding what the kids called the "Wheel of Fortune Letters" to the list in the LETTER procedure. I have added the Wheel of Fortune Letters (a e i o u l m n r s t) to the LETTER procedure in small letters so that we can observe how they are used in our new words. You may even wish to add two extra sets of vowels to the LETTER procedure. As you can see in the following example, approximately 9% of the words created in Experiment #2 were English words. Note how many proper nouns and common abbreviations appeared.

Modify the LETTER procedure as follows:

```
TO LETTER
OUTPUT PICK [ A B C D E F G H I J K L
              M N O P Q R S T U V W X Y Z
              a e i o u l m n r s t]
END
```

Results of Experiment #2

mUi	ssN	orV	PYm	DnD	aQl	oYU	VGO	TTr	lCU	RYo
laV	Kol	iYt	CXT	NSo	ZUH	**mOe**	QYE	EeT	JtF	**Nat**
tLO	lAF	oPM	riP	NQa	rDF	BmA	MOQ	**oFF**	MYn	Cll
QWW	mZM	iET	EHJ	Itl	moa	lLJ	sZi	JWs	AEs	Hom
asm	Rlo	WnX	Kln	iiS	sON	JiY	GTs	Krn	ZrP	oOV
Qlt	QiX	Alr	MQL	eLl	toF	lAC	sFi	**MeG**	AVS	UUs
AHI	miP	Wer	oli	DQe	GiS	AnH	IBs	RLH	sKe	mnV
RlO	ORC	QFN	FoM	aPa	nYi	iYs	ZYe	UiQ	DZa	***Aaa***
oae	rUA	Att	RWs	ReP	rXJ	iZV	oer	FHi	ZtF	**nET**
OoA	oVN	ELs	APS	**One**	WoQ	FLn	MBB	enS	CXW	Hrt
HBL	**lOG**	mDn	eSH	aHT	onT	Mie	PCD	lrs	***iBM***	iTi
iZG	**err**	PsA	iMa	CGV	INA	DQn	BKe	**rUN**	mmo	LZE
YnD	HGa	imL	otA	**BeV**	aVK	omP	rLA	JHo	mWZ	...

miD MAl ***oas Doa*** **sat nOt siD YAM ten neO saP AIM** GoV DIV ReV ...

Throughout this activity our "junior linguists" or "word scientists" should be encouraged to experiment, take risks, explore, and make hypotheses. My hypothesis is that we can create many more "good" words by imposing the rule that the three-letter words be assembled in consonant-vowel-consonant format. My hunch wasn't bad. The percentage of English words created in proportion to the number of words generated has risen to 16%.

Seymour Papert heard chuckles from the conference audience when he recounted the story of how he and some students at Project Headlight were waiting for the word "CAT" to be created and suddenly the word "DOG" appeared. Dr. Papert said that everyone watching the computer sighed a breath of relief because they felt as if they were getting closer. In my experiment (below) the word "DOG" also appeared. While it's impossible to count on "DOG" being generated, it is fascinating to ask students the question, "If DOG was the next word created would we be any closer to CAT being created?" The discussion that will undoubtedly follow will challenge the students' understanding of

Stager's Suff -- continued

their own thinking processes and may even lead to a discussion of such superstitions as, a "lucky day", or "I'm on a roll."

Experiment #3

Change the following line in CREATE.WORDS:

```
MAKE "NEW.WORD
    (WORD LETTER LETTER LETTER)
```

to:

```
MAKE "NEW.WORD
    (WORD CONSONANT VOWEL CONSONANT)
```

Add the following two procedures:

```
TO VOWEL
OUTPUT PICK [A E I O U]
END

TO CONSONANT
OUTPUT PICK [B C D F G H J K L M N P
   Q R S T V W X Y Z]
END
```

and type CREATE.WORDS 1

Results of Experiment #3

TUK QEH CAC **HEN** DOD QEX FEC **NIM** LIS
QOW YAR HOL KAF **HUT** PIZ CIY **BOG** YIM LUN BOC
NEN PEV GUD **GAS** RUQ JID QIH DEK REL FIQ **DID**
CAK **LAP** **WIM** MEX QUT LEH COH **HUB** VOL MEM VEB
REW BEY **BUG** KIG PIF **DOG** KAV ZEW SOR QUB **LAW**
NEB KIK RIH SOJ QUR TIG ZOM GOF DUR HEV RUZ
KOY **BON** XUX ZUB WEK **DAM** XUP WUZ WAC **NIM** XIX
LOH ZIC **TAN** MIB VEX YID QEN JAC LOS FUC WOY
ZUN FEQ JAC RAX CEX HUS WIV *SOS* XOT XOM GOX
YUP HOF DIV DOH LOS **MAR** **SAC** PEC LAQ DEL ROX
DAG BES DAF QIP BOR BEK WIR TUC FOH MIN KUQ
LUD QIR QUY YUQ JOP NUR POG KIR SEY **ROB** ZIK
TUT YIZ **TEN** PUZ GUS JEF KIB GEY **FOX** ... **GAP** **TIN**
HEN HON **SAT PAM DIM PIT MIT RAG JON TOM FAD** *ROM* **FAN** *POW*
VAT GEL VOW *PAC* **DAN KIM MAD WIN TAD JIM RUG RIG CAP RAY**
GOT DAB HIT NAY PEN HUB FOB FIT GUN DIP MEN SUB DOM BUN
LAG FUR BUY PUD DAY PIN DAN GUM GAS DIG TAP BIC TED LET
MOM YEN RIP SIP PAT HOT YES COW BIG MAY HEP SAD WAS JON
TOM FAD *ROM* **FAN** *POW* **JAY**

Experiment #4: A Bridge Between Language and Mathematics

In Experiment #4 we will substantially reduce our letter pool so that we can shift our focus from orthography, spelling, and vocabulary to averages, probability, and combinatorics. We will once again use the same CREATE.WORDS procedure as we did in Experiments #1-3. This time however, our pool of letters in LETTER will be restricted to [C A T].

```
TO LETTER
OUTPUT PICK [ C A T ]
END
```

Type REPEAT 1000 [CREATE.WORDS 1 PRINT [ ]] to observe how our data behaves since the pool of letters has been reduced. Notice the range of experimental results which appear - from 2 tries to 91.

Results of Experiment #4

TTC TTC ATA ACC ACT TTT AAA TCA AAA TCT TCT
AAA TTT TTC TAC CCT CCC TTC TTT ACT TAT CCT
ACT ATA TTA ACC ACA TAA CCA **CAT 30**
TAA **CAT 2**
CTC **CAT 2**
CCC AAA TAC ATA AAA CTC ACT CTC ATA AAT ACC
CCC AAA TCT ATT CAA **CAT 17**
TCA CAC ACA CTA ATC CTC CAC CAA CTC CCC ATC
AAT AAC CTT TAC TAA TTA CCA ACC ATT TCC TCA
CCA CCA ATC CAA AAC AAT CCC AAA TAA TAC AAC
AAA ATT CTA TTC AAT AAT CTT CAC AAA CCT CAC
CCA AAA AAC CCA CCT CTT CTT ATT CCA CAA AAC
TCC TCA CTA CTC CTT ACA ATC ACA CAC ACT ACT
TAA CTA TCT ACT TCT ACC CTA ACC CTA TTA TCA
ATC AAA TAT TAC TAT TCA AAA ACT TAT TAC TCC
TTA TTA **CAT 91...**

Collecting and Analyzing Data

Given a set of three letters, [C A T], how many possible permutations of these letters are possible? Mathematics teaches us that there are 3 x 3 x 3 or 27 possible unique combinations of the three letters. If you are like me and don't trust mathematicians you can use Logo to find out how many actual possibilities there are. Too often mathematics and mathematicians ask how many words can be created rather than show what those words actually are. By teaching Logo to add each newly created word to a list, we can count the number of possibilities and study what those words are by allowing the CREATE.WORDS procedure to run for a significant period of time. Eventually, three letters have to be assembled in every possible way.

Let's use Logo to help us collect and interpret the results of our experiment.

Change the CREATE.WORDS procedure to:

```
TO CREATE.WORDS :COUNT
MAKE "NEW.WORD
    (WORD LETTER LETTER LETTER)
IF NOT MEMBER?
    :NEW.WORD :POSSIBLE.WORDS
    [MAKE "POSSIBLE.WORDS
     LPUT :NEW.WORD :POSSIBLE.WORDS]
```

```
IF EQUAL? :NEW.WORD "CAT
   [PRINT SENTENCE
    :NEW.WORD :COUNT STOP]
INSERT WORD :NEW.WORD CHAR 32
CREATE.WORDS :COUNT + 1
END
```

Add the following new superprocedure:

```
TO POSSIBLE
MAKE "POSSIBLE.WORDS [ ]
CREATE.WORDS 1
END
```

1. Type POSSIBLE to begin.
2. After a minute or two stop the procedure with your equivalent of CTRL-G or -S.
3. Type SHOW :POSSIBLE.WORDS to see the words that can be possibly created by CREATE.WORDS.
4. Type SHOW COUNT :POSSIBLE.WORDS to see how many possible combinations of words created by our experiment. Did Logo tell you that there are 27 possible permutations?

Suggestions for Further Inquires

1. Run one of the experiments (or your own) over night and dump the created words to a dribble file or printer.
2. To save time, paper, and disk space modify the CREATE.WORDS procedure so that the randomly created words are not printed. All the procedure needs to do is record how many trials were necessary for "CAT" (or your word) to be generated:

```
TO START.EXPERIMENT
MAKE "TRIALS [ ]
REPEAT 9999 [ CREATE.WORDS 1 ]
END

TO CREATE.WORDS :COUNT
MAKE "NEW.WORD
   (WORD LETTER LETTER LETTER)
IF EQUAL? :NEW.WORD "CAT
   [ MAKE "TRIALS
     LPUT :COUNT :TRIALS STOP ]
CREATE.WORDS :COUNT + 1
END

TO AVERAGE :LIST
OUTPUT (SUM :LIST 0) / (COUNT :LIST)
END

TO SUM :LIST :TOTAL
IF EMPTY? :LIST [OUTPUT :TOTAL ]
OUTPUT SUM (BUTFIRST :LIST)
   (:TOTAL + FIRST :LIST)
END
```

The following line will display the average number of trials necessary to generate "CAT":

```
SHOW AVERAGE :TRIALS
```

3. Change the rules for creating words by making the CREATE.WORDS procedure "smarter" so that letters can not be used twice in a word or awkward combinations of letters don't occur.
4. Modify the CREATE.WORDS procedure to generate words with more than three letters.
5. Graph the number of trials or the ratio of English words to words created.

Send me examples of how you and your students modified these experiments.

Footnote

[1] Orthography is the term used to describe how a word is spelled. By looking at a word's spelling we are often able to determine if the word is a "real" word. Every language has distinctive rules of orthography.

References

Bull, Glen and Paula Cochran. (October 1985). Logo Word Chemistry. *The National Logo Exchange*, Charlottesville, Virginia.

Goldenberg, E. Paul and Wallace Feurzig. (1986). *Exploring Language with Logo*. Cambridge, Massachusetts: MIT Press.

Muir, Robs. (January 1986). It's Only Words. *The National Logo Exchange*, Charlottesville, Virginia.

Papert, Seymour. (1987). *Keynote Address*. East Coast Logo Conference, Arlington, Virginia.

Stager, Gary S. (1988). *The Literate Turtle - Novel Approaches for Using LogoWriter as a Tool for Exploring Language Arts*. Great Lakes/East Coast Logo Conference, Cleveland Ohio.

Gary Stager
5 Eastside Avenue
Wanaque, NJ 07465
CIS: 73306,2446
Applelink: K0331

The Emerald Isle Shines Again

The 1988 International Computer Problem Solving Contest Elementary Logo Results

by Donald T. Piele

In 1986, an Elementary Logo Division was added to the International Computer Problem Solving Contest (ICPSC). Last year, we expanded the Logo contest to include the same three age groups used in the Open Division: Elementary (grades 4-6), Junior (grades 7-9) and Senior (grades 10 -12). On the last Saturday in April, teams gather at local contest sites and attempt to solve five problems in two hours using Logo. This year an estimated 1,000 students from the United States and a few foreign countries participated in the Logo Division. This month, the results of the Elementary Logo Division are presented. The problems from last spring's contest will appear in next month's *LX*. The Junior and Elementary divisions will be presented in coming months.

Elementary Division Winners (Logo)

One of the great pleasures associated with the International Computer Problem Solving Contest is researching background information about the winning teams. Each year, I ask the top team's contest director or advisor to send me information that I can use when I write about the results for publication. The stuff I get is fascinating. I am amazed how often the winning team is made up of students who *don't* spend all of their waking hours in front of a computer. Most have a wide range of interests outside the world of computing. No one, in recent memory, typifies this breadth of interest better than this year's winning team in the Elementary Logo Division. Her name is Anne Chazarreta , age 12, from Scoil an Spioraid Naoimh School, Bishoptown, Cork, Ireland. Yes, the winning team consisted of only one girl! And it is the second year in a row that an Irish team took top honors.

Instead of piecing together a story about Anne, I would like to share with you the full letter I received from Michael D. Moynihan, Chairman of the Computer Education Society of Ireland. He writes,

> There was great excitement in Scoil an Spiorald Naoimh when it was learned that Anne Chazarreta was ranked first in the world in the 1988 International Computer Problem Solving Contest (Elementary Logo Division). This prestigious contest is organized annually by the University of Wisconsin-Parkside and is open to children of the same age group all over the world. Anne is the first person from Cork to win. Indeed it is worth noting that for the second year running an Irish child has won (John Farragher from Limerick was the 1987 winner).
>
> Anne is the only child of Jose and Philomena Chazarreta of Woodbrook, Bishopstown. Her parents are justly proud of her achievement. Her school Principal, Mrs Joan Toomey and class teacher Mrs Mary Fleming are equally proud of the honour that her achievement bestows on the school. Mrs Fleming describes Anne as a gentle unassuming person who is most co-operative in school and achieves a lot in her own quiet way. She is also a perfectionist with an excellent logical mind.
>
> For the past few years experimental courses in LOGO for mathematically talented children have been conducted at various centers throughout Ireland. These courses are co-ordinated by Dr. Sean Close, Mathematics Department, St. Patrick's College of Education, Drumcondra, Dublin. The Cork center is at Colaiste an Spioraid Naoimh, Bishopstown and is directed by Declan Donovan and myself. Anne was one of our students.
>
> Throughout the twenty week course Anne proved herself to be one of the more able students and quickly came to grips with Logo and displayed problem solving skills which amazed even her tutors. In April, she was ready to enter the ICPSC at the Cork site sponsored by the Computer Education Society of Ireland (CESI). Anne solved all five problems in one and a half hours and spent the remaining time testing her solutions. We knew that she had solved them all correctly, but we did not expect to be so highly ranked.
>
> Anne's other main interests are music and reading. She has been learning to play the piano for four years and also likes to play the guitar. She also was a member of the Irish National Children's Choir in 1987. Anne enjoys reading the classics and was recently taken by Joan Lingard's novels about the Northern Ireland situation and the Protestant vs. Catholic clashes. An all-rounder, in April Anne was presented a special award by the Lord Mayor of Cork for an essay on her environment. She has had her own computer (a BBC Micro) at home since Christmas 1986.
>
> Anne is in her final few months in Primary school. In September, she commences her Secon-

dary education at Bishopstown Community School.

The Computer Education Society of Ireland has been active for many years in the promotion of and use of computers in schools. In recent years, it has given special encouragement to students at the primary level."

Anne Chazarreta

he Importance of Style

This was our second year of offering a Logo contest in the Elementary Division. It could not have been done without the assistance of Sharon Yoder, editor of the *Logo Exchange* and professor at the University of Oregon. Sharon was responsible for creating the Logo problems, solutions, and for ranking the best teams. She remarked that style was often the determining factor and that bad style consists of:

Lots of unrelated statements on a single line;
Extremely long procedures;
Modularity not used to its fullest;
Procedures without meaningful names;
Procedures includedwhich weren't part of the final program.

She also observed that some teams were ranked lower because they they apparently viewed the first problem as quite easy and thus did not take the few extra minutes to polish theif work.

1989 Contest

The 1989 ICPSC will be held on Saturday, April 29, 1989 with Friday, April 28 and Monday, May 1 as alternate contest dates. *Compute It!*, the newsletter of the ICPSC, includes more information on the event and how your school or school district can become a contest site. For a free copy, write to me at the address below.

Donald T. Piele
ICPSC
P.O. Box 085664
Racine, WI 53408

1988 Elementary Logo Division Rankings

Rank	Team	School	City, State	Director	Advisor
First	Anne Chazarreta	Scoil an Spioraid Naoimh	Bishopstown, Cork, Ireland	Michael D. Moynihan	
Second	Bryan Dodds	Falcon Heights El	Roseville, MN	Mike Amidon	Rita Christianson
	Joel Nordell				
Third	David Frink	Ligon Middle School	Raleigh, NC	Cathay Smith	
	Charles Powell				
Fourth	Mark James	Scoil an Spioraid Naoimh	Bishopstown, Cork, Ireland	Michael D. Moynihan	
Fifth	John-Paul Corkery	St. Josephs	Bishopstown, Cork, Ireland	Michael D. Moynihan	
Sixth	John Sullivan	Scoil Barra	Bishopstown, Cork, Ireland	Michael D. Moynihan	
Seventh	Adrian Young	Park Terrace El	Spring Lake Park, MN	Mike Amidon	Mari Good
	Damon Durand				
	Jamie Groat				
Eighth	David May	Academic Resource Center	Tallahassee, FL	Robert Bruggner	
	Erik Brooks				
	Rod Brown				
Nineth	Daniel Wolfe	Roeper City & Country	Bloomfield Hills, MI	Terry Rudman	
	Jonathon Sheppard				
	Eddie Lee				
Tenth	Geraldine Hurley	Scoil an Spioraid Naoimh	Bishopstown, Cork, Ireland	Michael D. Moynihan	
Also solved five problems but arrived too late to be included in rankings.					
	Graham Donohoe	St. Patrick's College	Drumcondra, Dublin, Ireland	Dr. Sean Close	
	Alan Ayling				
	Lisa Flynn	Holy Faith	Drumcondra, Dublin, Ireland	Dr. Sean Close	
	Jennifer Carey	Holy Child			
	Tom Hill	Kings Hospital			
	Eoin Curran	St. Patrick's College	Drumcondra, Dublin, Ireland	Dr. Sean Close	
	Timothy Deegan	St. Patrick's College	Drumcondra, Dublin, Ireland	Dr. Sean Close	
	Ruth Brennan	St. Patrick's College			

Logo LinX

The Stranger, the Better
by Judi Harris

"The metaphor is perhaps one of [humanity's] most fruitful potentialities. Its efficacy verges on magic, and it seems [to be] a tool for creation...."
—Jose Ortega Y Gasset, 1925

Metaphors are fertile seeds for germinating creative writing. Consider, for example, what your students' essays might contain if they were asked to address questions like these:

```
HOW IS A SAILBOAT LIKE A PAINTBRUSH?
WHICH IS NOISIER:  WISHING OR ACTING?
HOW IS RACISM LIKE WEAVING?
WHICH IS LESS TRUSTWORTHY: MEASURING OR SEEING?
```

In 1968, Synectics, Inc., an independent educational research group, published an excellent set of classroom materials that encouraged the conscious use of metaphor to stimulate creative thinking, and subsequently, original writing. They called the series "Making It Strange," explaining that analogies and metaphors that make the familiar seem strange are powerful catalysts for unique perspectives from which to write.

The Greek-derived term "synectics" refers to joining two superficially irrelevant ideas. Synectics are the bases of metaphor. Four "Making It Strange" workbooks provide classes with guided activities that use different types of analogies as story starters. With a few simple Logo procedures, your students can generate individualized metaphors to inspire their creative writing. And unlike consumable instructional supplies, this type of Logo output is rarely repetitive and never exhausted.

Choices by Chance

The structure of an analogy is simple:

"How is ________ like __________ ?"

Or, in Logo:

```
PRINT ( SENTENCE [HOW IS] choice#1
        [LIKE] choice#2 "? )
```

Choices can be nouns or verbs; objects or ideas. Infinite numbers of analogies can be produced by using the RANDOM primitive with just a few question templates.

The well known PICK tool:

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

can be used within NOUN, VERB, and IDEA procedures to output choices for analogy templates.

```
TO NOUN
OUTPUT PICK [[A BALLOON] [A FROG] [A PAINTBRUSH]
    [A MAILBOX] [A DISHWASHER] [A QUILT] [A TRASH
    COMPACTOR] [A CARBERATOR] [A MODEM] [A BOOK]
    [A SPORTS CAR] [A PANDA] [A MUSIC VIDEO] [A
    NOTEBOOK] [AN ASH TRAY] [A HAMMER] [A POND] [A
    SAILBOAT] [A COMPUTER PROGRAM] [A TURTLE] [AN
    AIR CONDITIONER] [A TREE] [A SEASHELL] [AN
    ELEVATOR] [A SHOPPING MALL] [A BUMBLEBEE] [A
    CRAB] [A TULIP] [A COBRA] [A MINNOW] [A
    MIRROR]]
END

TO VERB
OUTPUT PICK [SKIING SWIMMING LAUGHING STUDYING
    TYPING THINKING PLAYING ACTING SCRATCHING
    [BEING EMBARRASSED] PAINTING JOGGING SHOUT-
    ING COUNTING DISCUSSING TESTIFYING DISCOVER-
    ING SLEEPING COOKING KISSING EATING MEASURING
    REVISING WISHING GLIDING STRETCHING DANCING
    GIGGLING SWEATING PASTING TRAVELING WEAVING
    SEEING SEWING POUNDING]
END

TO IDEA
OUTPUT PICK [DEMOCRACY TRUTH HOPE LOVE HATE FACISM
    ORDER LEGALITY IMMORALITY RELIGION GOVERN-
    MENT POLITICS GENEOLOGY ACADEMIA SEXISM
    RACISM ENLIGHTENMENT CULTURE MORALITY [THE
    WORK ETHIC] [AFFIRMATIVE ACTION] INDEPEND-
    ENCE LIBERTY FREEDOM INTELLIGENCE FRIENDSHIP
    INTEGRITY RESPONSIBILITY AWARENESS]
END
```

When constructed as procedures, these lists are easy to amend according to student interests, current topics of study, or new vocabulary.

Procedures which output lists of these categories:

```
TO CATEGORIES
OUTPUT [ [NOUN] [VERB] [IDEA] ]
END

TO CATEGORIES2
OUTPUT [ [NOUN] [IDEA] ]
END
```

can be used to randomly order choice types within question templates. For example:

HOW IS (a noun) LIKE (another noun) ?
HOW IS (verb, ending with -ing) LIKE (a noun) ?
HOW IS (a noun) LIKE (an idea) ?

Productive Procedures

Production of these metaphors is a matter of simple synectics. In Logo, that is accomplished through random concatenation.

```
TO ANALOGY
OUTPUT ( SENTENCE [HOW IS]
        ( RUN PICK CATEGORIES ) [LIKE]
        ( RUN PICK CATEGORIES ) "? )
END
```

To use this procedure, type:

```
PRINT ANALOGY
```

The computer may print:

```
HOW IS A TREE LIKE A MAILBOX?
HOW IS COOKING LIKE KISSING?
HOW IS HATE LIKE A DISHWASHER?
```

Other analogy types can be generated easily with similar procedure structures.

```
TO COMPARE
PRINT ( SENTENCE [WHICH IS] COMPARISON ": )
PRINT ( SENTENCE ( RUN PICK CATEGORIES ) [OR]
      ( RUN PICK CATEGORIES ) "? )
END

TO COMPARISON
OUTPUT PICK [NOISIER HEAVIER THINNER [MORE
    IMPORTANT] SLOWER DEEPER HEALTHIER [MORE COL-
    ORFUL] BRIGHTER SMARTER [MORE HESITANT] [MORE
    POPULAR] [MORE SENSITIVE] [MORE EXPENSIVE]
    [MORE USEFUL] [MORE COMFORTABLE] [LESS
    TRUSTWORTHY] [LESS DEPENDENT] NEEDIER [LESS
    REMOTE] TASTIER [MORE PLUSH] [MORE LOVABLE]
    [LESS DANGEROUS] SHINIER [LESS FRAGILE] ]
END
```

To see an example of this breed of writing idea, type:

```
COMPARE
```

The computer may return:

```
WHICH IS MORE HESITANT:  RELIGION
   OR TRUTH?
WHICH IS MORE COLORFUL:  LOVE
   OR LIBERTY?
```

Action-based metaphors can be generated with this procedure:

```
TO DESCRIBE
PRINT ( SENTENCE "DESCRIBE
      ( RUN PICK CATEGORIES2 ) VERB ". )
END
```

Type:

```
DESCRIBE
```

and the computer may show you:

```
DESCRIBE A MAILBOX BEING EMBARRASSED.
DESCRIBE INTEGRITY GIGGLING.
```

Assisting (Not Replacing) Thinking

The Logo turtle is often described as an "object to think with." Simple, carefully crafted Logo tool procedures can also stimulate thinking. The power of Logo-catalyzed metaphor work lies in the computer's patient compliance with the requests of one student who generates 40 analogies before finding one that captures her imagination and inspires a divergent story idea, or those of another who insists upon changing the contents of the word lists before using the procedures to find his essay topic. These young authors would probably be impatient with class-wide uniform writing assignments, and their teacher probably does not have enough time to devise and distribute individualized story starters.

Thoreau said,

> All perception of truth is the detection of an analogy. (1851)

Logo can be used to randomly synthesize analogies, but only humans can recognize interesting metaphors in computer output and portray them creatively.

The message is clear: it makes more sense to let computers *generate*, and people *create*.

References:

Syntectics, Inc. (1968). *Making it strange*. Vol. 3. New York: Harper & Row.

Syntectics, Inc. (1968). *Making it strange*. Vol. 4. New York: Harper & Row.

Syntectics, Inc. (1968). *Making it strange: A new design for creative thinking and writing, teacher's manual.* New York: Harper & Row.

Judi Harris
621F Madison Avenue
Charlottesville, VA 22903
CIS: 75116,1207
BitNet: jbh7c@Virginia

Logo Connections

Math Connections

by Glen L. Bull and Gina L. Bull

One of Seymour Papert's visions which has intrigued parents and educators is his description of Mathland. Papert notes that even after years in the language lab, children often don't learn to speak a foreign language well — yet an American child who spends a summer in France may learn to speak the language like a native. Papert points out that this demonstrates that children have the inherent capacity to learn foreign languages — it is just a matter of finding the proper learning environment. Papert proposed use of Logo to construct a Mathland. Mathland would be the mathematical equivalent of France — a place in which children could learn mathematics easily and naturally.

The reason that French is easier to learn in France is that the language is used as a tool to meet the needs of the user. A command of the language enables the learner to order a meal, make friends, or find directions. In the language lab, a command of the language enables the learner to earn a grade at the end of the semester, but serves no immediate need. There's a big difference between learning about French and using French to accomplish another goal.

There's also a big difference between learning about math and using math to accomplish a goal which the learner wishes to achieve. That's where Papert's Mathland comes in. The trick is to create an environment in which the learner will need to use numbers in the process of achieving another goal. Turtle graphics were developed in an attempt to create components of a Mathland. Many children have received hours of pleasure from creating drawings by directing the turtle.

However, turtle graphics are not a panacea. Children have different learning styles, and some children do not find turtle graphics engaging. Also, as paint programs which make use of graphics tablets and mice became available, more ways of creating drawings became available. A house can be created on the graphics tablet in a fraction of the time it is created with the turtle, and without the use of any numbers at all. Of course, graphics can be created with Logo that would be laborious to develop with a graphics tablet, but often children never reach that level of expertise, or go beyond "turtling around" with FORWARD 10 and BACK 10.

LEGO-Logo adds new dimensions to Papert's Mathland. Some children who are not attracted by the two-dimensional confines of the computer screen are intrigued by the possibility of controlling a three-dimensional object. When a Logo program is developed to control a LEGO robot or truck constructed by a child, numbers are sure to be involved.

In the last column, we reported on a young child who constructed a fan with LEGO parts. The fan soon evolved into an airplane, complete with propeller. At first Stephen was content to learn the words ON and OFF to control the propeller. (ON turned the propeller on, and OFF turned it off.) However, LEGO-Logo permits a motor to be run at seven different speeds. Stephen soon asked for a way to control the speed of the motor.

The formal way to control the speed of a motor in LEGO-Logo is the word SETPOWER followed by a number between 1 and 7. To set the motor to its fastest speed, the command "SETPOWER 7" would be typed. However, this command did not seem appropriate for a child of Stephen's age. We had shown Stephen how to control lights on the wings of the plane with the commands "RED LIGHT" and "GREEN LIGHT", since the words RED and GREEN are part of his daily vocabulary. However, "SETPOWER" is unlikely to appear in contexts other than Lego-Logo.

On the spur of the moment we created a command called "FAST". The command FAST was used to make the propeller go faster. Each time FAST was typed, the speed of the propeller increased. We were not sure what to do when the propeller reached its fastest speed (power level 7). Finally we decided to reset the level back to 1 (the slowest speed) after level 7 was reached. The Logo procedure looked like this:

```
TO FAST
IF :SPEED > 7 [MAKE "SPEED 0]
MAKE "SPEED :SPEED + 1
SETPOWER :SPEED
END
```

Stephen soon learned to type the word FAST, motivated by a desire to control the propeller. He made the propeller go faster and faster as quickly as he could type the word and press RETURN. On the seventh time he was startled when the propeller slowed down instead of going faster. At first he was afraid that he had broken the motor. However, we explained to him that when the motor reached its fastest speed, the next FAST caused the motor to return to speed 1. Once he understood that this was how the command worked, he happily began increasing the speed of the propeller again.

Since control is such an important issue, Stephen wanted to be the one who made the propeller speed return to 1. He soon found a way of doing this. Just before he typed "FAST" for the seventh time, Stephen typed "OFF". This allowed him to be the one to shut down the propeller, instead of the computer. In passing, we would note that a procedure called FAST which sometimes makes the motor go slow probably isn't a good idea, even though we suspect that it is good preparation for the real world. However, this quirk of the FAST procedure had a serendipitous result.

Because Stephen knew that the FAST command worked for six times before it wore out, he began counting the number of times which were left:

"FOUR — now I have two times left."
"FIVE — now I have one times left."

We were surprised. About a month before we had attempted to interest Stephen in the concept of subtraction.

If you have a row of three beads, and take one away, how many beads are left?

Stephen wasn't sure. More importantly, he couldn't see why anyone would care how many beads were left. He was more interested in using them as marbles.

Possibly in a month's time Stephen had now matured so that he was developmentally ready for the concept of subtraction. However, we believe that the important factor was the fact that when Stephen was playing with the LEGO-Logo airplane, subtraction allowed him to control the propeller more effectively. He was using numbers to achieve a goal that mattered to him. Connecting the computer to the motor which turned the propeller (instead of providing a hand control) made it more likely that numbers would be useful in controlling the motor.

Not every child will want to build LEGO toys and control them with the computer. However, the addition of robotics to the turtle graphics and list processing capabilities of the language will increase the chances that each child will find something of interest to them personally. This episode also suggests that LEGO robotics can provide worthwhile experiences for younger children as well as older ones.

Postscript

At the start of the column we suggested that some children may prefer to use a mouse or a graphics tablet for construction of illustrations instead of the Logo turtle. It is not always necessary to choose between a graphics program or Logo. It is possible to load a picture created with a graphics tablet into Logo, or export a Logo picture to a graphics program for finishing touches (See Logo Ideas in September 1988 *LX*). The two programs working together can be more powerful than either alone. For example, once a Logo procedure for a HOUSE is created, it can be used with the REPEAT statement to draw an entire village. A graphics program can then be used to add grass and other artistic features that may be difficult to create with Logo.

This approach is seldom used in the classroom, in part because the Logo manuals provided by manufacturers often say very little about how to transfer Logo graphics to other programs, or how to take advantage of the vast libraries of "clip art" pictures created for programs such as Print Shop. The process is often straightforward — but that's a subject for another "Logo Connections" column.

Glen is a professor in the University of Virginia's Curry School of Education. His BitNet address is GLB2B@ VIRGINIA. Gina is a programmer analyst for the University of Virginia Department of Computer Science. By day she works in a Unix environment; by night in a Logo environment. Her BitNet address is RLBOP@VIRGINIA.

Glen and Gina Buill
Curry School of Education
Ruffner Hall
University of Virginia
Charlottesville, VA 22903

MathWorlds

edited by A. J. (Sandy) Dawson

I feel reasonably confident that most readers will have heard of the Game of Life, and perhaps even seen Brian Silverman's wonderful manifestation of it written in Logo. Why then, you might ask, does this column contain an article about the Game of Life? I wondered the same thing when Jim Noyes called me about it. I was intrigued enough by what he said that I suggested he mail me a disk and a printout of his article. He did, and when I finally found a quiet moment last summer I sat down and watched the Game of Life in a new way. The thing which Jim has added is color for each generation of the objects, so you can watch cells be born and age and with each generation they survive they change color. I was quite amazed to see that on the whole, cells don't live very long at all. In viewing many iterations of the game, I only saw one cell which reached an age of ten at which point it stabilized. Yes, I know, if one creates certain designs which stabilize then all cells which age infinitely. But I am talking about configurations which don't either die out or stabilize. Anyway, I hope you enjoy Jim's Life with Logo.

Life with Logo or Colorizing the Game of Life

by Jim Noyes

Introduction

One of the better-known computer simulations is the Game of Life, devised by John Conway. This is a game that simulates *biological organism growth* based only upon an initial pattern selected by the player and three rules that determine the game. Many programs have been written to implement Life and two years ago the author decided to implement Life with IBM Logo. This was done using "text screen graphics" in order to be able to represent the *ages* of the individual organisms through the use of a variety of *colors*. As each organism's age changes, so would its color. The many references to Life and its various evolving patterns typically show each living organism as a black dot so that these patterns of black dots appear homogeneous. Many have become so well studied that they have been given names based upon their shapes. However, when color is used a beautiful heterogeny may be observed and this allows the Life patterns to be viewed in a new way and allows the Life patterns to be viewed in a new way and allows Life to now be studied from a different perspective. It is an ideal simulation for both adults and children to explore because of its simple rules, easy input and colorful output patterns.

The Game of Life

Life is played on an infinite two-dimensional grid of square cells. A computer is not necessary to do this (e.g., it can be done a checkerboard), but doing it without a coputer is a very time consuming and error-prone process. Initially the player identifies certain cells to have life at Generation 0. This is done by placing a symbol, such as a dot, in each appropriate cell. All of the others are assumed not to have life (dead cells) and are shown as empty. Each cell can have up to eight living cell neighbors because that's the number of adjacent cells. The rules are entirely based upon the number of neighbors surrounding an individual cell. The rules are as follows:

1. **Birth:** An empty (dead) cell that has exactly three neighbors in the current generation will have a live organism in the next generation.
2. **Survival:** An organism with either two or three neighbors in the current generation will survive into the next generation.
3. **Death:** An organism with fewer than two neighbors or more than three neighbors in the current generation will die (respectively from exposure or overcrowding) in the next generation.

This simulation can be accomplished by first checking all living cells and counting their neighbors. A living cell is marked for death if it has other than two or three neighbors. Dead (empty) cells, next to the living cells, also have their neighbors counted. If a dead cell has exactly three neighbors then it is marked for birth. finally, each marked cell is set as live (e.g., displayed with a symbol) or dead (displayed as blank) as appropriate. This defines the next generation. This process can continue for as many generations as the player wished or until no living cells remain. For additional details about the Game of Life, consult references 1, 2, or 4 at the end of this article.

Implementing Life with Logo

Normally, one tries to implement the Game of Life in the most efficient way possible because if there are a lot of living cells, the computation of a new generation can be quite a time consuming process. This was not a consideration here. The author's program, called ColorLIFE was designed to display only those organisms that will fit within a text screen having 21 rows and 38 columns. Organisms outside that area are computed, but not shown. Using IBM Logo character symbols makes viewing much easier, especially for children. This also permits a wider range of colors to be used. Symbols such as a heart or a smiley-face (done with Control-C or Control-B) can also be used for additional variety. The procedures for ColorLIFE are outlined below in terms of their dependencies and some of them are listed at the end of this article. LIFE is the main procedure.

LIFE: sets game up and loops through processes.
  INITIALIZE: Lets player define the initial pattern.
    STARTUP: allows player to define Life symbol
      DIRECTION: provides directions if needed
    SETSCREEN: defines screen (board) limits and colors.
  CLEARLINES: clears the status lines on the screen

DISPLAYBOARD: displays board for the current generation
GETRC: gets the current age of a given cell.
COLORCELL: assigns and displays the symbol in color
COUNTNEIGHBORS: counts the number of live neighbors
GETRC: gets the current age of a given cell
PUTRC: puts the current age in the cell
BIRTH.SURVIVE.DEATH: marks and updates the appropriate cells
CLEARBOARD: clears the board of its current pattern

A key idea in this implementation is to use a Logo *property list* to identify the living cells. A two-dimensional "infinite" array is needed because each cell position is uniquely specified by a row and column number. Hence the name of each item is the property list is "created" by making a Logo word using four parts: the letter R, the integer row number, the letter C and the integer column number. The property associated with this name is a two=element list consisting of its status and the number of neighbors it has. Its status is: B (marked for birth), D (Marked for death) or n (an integer representing its age and indicating survival.) The state of the board is contained in this property list. For example,

```
R12C25 [D 1]
```

indicates that the cell at row 12 and column 25 of the board has the status of D (marked for death) because it has only one neighbor in the current generation. It will be removed from the property list and hence not displayed in the next generation. If another is given by

```
R10C20 [2 3]
```

then it is at row 10 and column 20 with an age of 2 and with three neighbors. It will stay in the property list and be displayed in the next generation as a red symbol (see below.) GETRC and PUTRC respectively get and put things in the property list. For additional information about property lists, see reference C at the end of this article.

ColorLIFE uses a white text screen and foreground colors as show below. The age determines which foreground color is used.

Age	Color	
0	Black	
1	Brown	
2	Red	If the age is
3	Light Red	10 or older then
4	Light Magenta	the color is
5	Light Blue	high-intensity
6	Cyan	white
7	Green	
8	Light Green	
9	Yellow	

Even more colors are possible using the text screen while only three pen colors are possible if the graphics screen is used. This program has been used by both children and adults in a variety of settings and almost always stimulates of lot of interest. If additional information is needed or a copy of the program is desired, please contact the author at the address given at the end of this article. A stamped, self-addressed 5 1/2 floppy disk mailer and a blank disk are necessary if a copy of the program is requested.

Examples

Since color cannot be shown in this article, each display will be shown in two forms. The first will contain the life symbol as would normally be displayed, but without color. The second will have the same pattern, but will show a digit in place of the corresponding symbol. The digit indicates the age of the symbol. The digit 0 represents age zero (birth).

1. For the symbol "*", the following inputs define Generation 0:

```
 9 18
 9 19          ***          000
 9 20           ***          000
10 19              *           0
10 20
10 21
11 22    Generation 0       Cell Ages
```

```
      *            0
     *  *         1  0
     *  *         0  1
       **           00
 Generation 1 Cell Ages
```

```
     ***          200
      * **          0 20
       **           11

 Generation 2 Cell Ages
```

```
      *            0
     ****         3110
     *   *        0   1
       ***          220

 Generation 3 Cell Ages
```

This generates a sequence of patterns that become stable at Generation 15. A pattern is stable when the living cells continue to age, but no more die and no more are born.

2. The well-know R-Pentomino will be shown next using the same symbol. As generations advance it produces a wide distribution of other Life forms. We will only show a few selected generations up to Generation 20. Since many organisms are produced, we will not attempt to line up the

MathWorlds -- continued

row and column positions for each generation as we did in the previous example.

```
 9 19              **               00
 9 20             **               00
10 18              *                0
10 19
11 19
     Generation 0        Cell Ages
```

```
        *          0
      ** **       10 20
      *  *        3  3
       **          45

   Generation 5 Cell Ages
```

```
       **          10
      **          02
       **          18
        ***         000
          *           2
          *           0

  Generation 10 Cell Ages
```

```
    *   ***          0   010
   **     **        13    31
  **     **        02    00
   * *   * *         1 0   0 0
    **    ****        12   1002
     *   *   **        0   3   91
     *   * **              2  70
         ***               010

  Generation 20 Cell Ages
```

Despite the fact that Generation 20 has been reached, no cell organism has yet reached an age of ten generations. In fact, the oldest two of ages 7 and 9 die from overcrowding in the next generation. In Generation 21, the oldest living cell is only four years of age. The R-Pentomino produces one of the most beautiful and unpredictable color displays. So what appears as a homogeneous pattern on the left is clearly heterogeneous and can be further studies for sub-patterns and age groupings. This is what makes ColorLIFE so interesting.

References

Conway, John H. (1982) "What is Life? in E. Berlekamp, J. Conway and R. Guy, *Winning Ways for Your Mathematical Plays* (Vol 2)., Academic Press.

Gardner, Martin. (1983) *Wheels, Life and Other Mathematical Amusements*, W. H. Freeman.

Harvey, Brian (1986) *Computer Science Logo Style, Volume 1: Intermediate Programming*, Cambridge: MIT Press.

Poundstone, William, (1985) *The Recursive Universe*, William Morrow & Co.

Partial Program Listing

```
TO LIFE
; [WRITTEN BY : J. L. NOYES, JULY 1986]
(LOCAL "BGNDCOLOR "FRGDLIST "FRGDCOLOR "LIFECOLOR
"SYMBOL "BLANK "TIME "BOARD "SHOWAGE "GENSTOP "LLOOP)
MAKE "TIME 0
INITIALIZE
CLEARLINES
TYPE [Enter Generation No. to Stop \ ]
MAKE "GENSTOP READWORD
CLEARLINES
LABEL "LLOOP
COUNTNEIGHBORS
BIRTH.SURVIVE,DEATH
MAKE "TIME :TIME + 1
(PRINT "ColorLIFE: "Current "Generation
    "No. :TIME)
DISPLAYBOARD
IF :TIME < :GENSTOP [GO "LLOOP]
TYPE [Enter Generation No. to Stop \ ]
MAKE "GENSTOP READWORD
IF :GENSTOP > :TIME [GO "LLOOP]
PRINT [Enter Y if LIFE list is to be output]
SETTC [7 0]
IF READWORD = "Y [CT SHOW PLIST "BOARD]
CLEARBOARD
END

TO PUTRC :ROW :COL
(LOCAL "COORDS "SLIST "STATUS "NBRS)
MAKE "COORDS (WORD "R :ROW "C :COL)
MAKE "SLIST GPROP "BOARD :COORDS
IF EMPTYP :SLIST [PPROP "BOARD :COORDS LIST "B 1]
    [MAKE "STATUS ITEM 1 :SLIST MAKE "NBRS
    ITEM 2 :SLIST PPROP "BOARD :COORDS
    LIST :STATUS (:NBRS + 1)]
END

TO GETRC :RC
(LOCAL "T "R "AGAIN "OUT)
MAKE "T BUTFIRST :RC
MAKE " FIRST :T
LABEL "AGAIN
MAKE "T BUTFIRST :T
IF FIRST :T = "C [GO "OUT]
MAKE "R WORD :R FIRST :T
GO "AGAIN
LABEL "OUT
OUTPUT LIST :R BUTFIRST :T
END
```

I wish to thank Jim for sending his material to me. Won't other out there do likewise? If you don't have access to E-mail and wish to find out whether something you have written might be suitable for MathWorlds, give me a call—collect if you like—at 604-291-4326. I'd be pleased to talk with you about your ideas for a column.

Jim Noyes is an Assistant Professor of Computer Science at Wittenberg University in Springfield, Ohio. His address is Wittenbery University, Department of Mathematics and Computer Science, Box 720, Springfield, Ohio 45501

A. J. (Sandy) Dawson is a member of the Faculty of Education at Simon Fraser University in Vancouver, B. C., Canada. His CompuServe number is 76475,1315. He can also be reached electronically as UserDaws@SFU.BITNET

LogoWriter and ASCII Files

by Eadie Adamson

With Apple LogoWriter version 2.0, you can exchange files between LogoWriter and such wordprocessing programs as AppleWorks, AppleWriter, Bank Street Writer, Magic Slate, or even Apple Logo II. The files must be saved as ASCII or text files in ProDos. If your program saves DOS files, you must convert them to ProDos.

How to Load ASCII (Text) Files into LogoWriter

Follow this process to load any ProDos ASCII files:

1. Load LogoWriter in Drive A. Get a NEWPAGE.

2. Put the disk with text or ASCII filein the second drive.

3. Type:
```
SETDISK "B
```
Then type:
```
LOADTEXT "file.to.load
```

4. When the text appears on the screen, type
```
SETDISK "A
```

5. NAME your page:
```
NP "pagename
```

When the name you typed appears at the top of the screen, your text has been converted to a LogoWriter page!

How to Save LogoWriter Files as Text Files

On a LogoWriter page, you may save either the front or the back as an ASCII file. (If you need the text on both sides, paste it temporarily on one side; type RESTORE after saving the file as text — to get the page back as you want it for LogoWriter.)

1. Flip to the side of the page you wish to save as a text file.

2. Type:
```
SAVETEXT "name for file
```

3. See if you have saved it:
```
SHOW FILELIST
```

FILELIST will return a list of all pages, marked **lwp** for LogoWriter page, **lws** for SHAPES pages, **txt** for text files. Your new ASCII file should be in the list.

Note: you cannot edit a LogoWriter page of text, then save it again as text over a previously saved text page. Instead, either save with a new name (SAVETEXT "newstuff), append a number to the end of the name (SAVETEXT "MY.TEXT2), or erase the previously saved page.

How to Erase a Text File from a LogoWriter Disk

To remove a text file from a LogoWriter disk, type:
```
ERASEFILE "name.of.file
```

ERASEFILE can be used to erase a SHAPES page, a LogoWriter page, or a text file.

Note: You can also save a text file directly onto your wordprocessing disk. Simply put it in the second drive, load your LogoWriter page, type SETDISK "B, and then type SAVETEXT "name of file .

Eadie Adamson
Allen Stevenson School
132 East 78th Street
New York, New York 10021

Little Kids and Logo

Oh, Shucks!
by Leslie F. Thyberg

This month's column features a few corny suggestions, as well as some more practical ideas that can be the kernels from which you can reap a well-managed classroom of creative and independent thinkers.

As a primary level classroom teacher I made every effort to integrate the students' learning rather than compartmentalize it by subject area. Consistency of this sort lends itself to clarity and cognitive growth. Thus, I chose to teach problem solving and programming in the same manner as I did teaching students to develop and extend their writing skills — which was to use the principles of the writing process. It was my experience that there is no set order to the stages of the writing or thinking process. The process is like children themselves — highly idiosyncratic, situation dependent, and non-linear in progression.

There are identifiable ingredients involved in the process approach. It was my classroom practice to treat these stages or ingredients as applicable to all content areas, particularly writing and Logo. I chose to label topic choice and rehearsal as part of what is usually referred to as the "pre-writing" stage. Similarly, Bransford's IDEAL Problem Solving Model (1984) is a five-step process. I called my approach the *Composite Process Model.* Its components are: topic choice (identifying the problem), rehearsal (defining the problem), composing (exploring possible strategies), reading (acting on a selected strategy or plan), and revision (looking at the effects). The following is a brief description of how I chose to use it in a classroom context.

1. Topic Choice and Identifying the Problem

Whether writing an essay or creating a Logo project, both begin with the selection of a topic. Topic ideas are provided in the Turtle Center and at the Writing Center. Topics may overlap. A topic may be embedded in a previous project. The child may choose to collaborate with a peer or select a topic from a brainstorming session or a list labelled "future topics" that he or she keeps tucked away in a Turtle folder or notebook. This ability to identify a problem is essential for effective problem solving, program generation, and writing. This month's *corny* ideas are an example of thematic or center oriented study.

2. Rehearsal and Defining the Problem

Rehearsal refers to the conscious or unconscious preparation writers and programmers make for what is to follow. Daydreaming, sketching, working with manipulatives, playing turtle are all part of the rehearsal process. Rehearsal can include what Dan and Molly Watt describe as playful exploration (Watt, 1986). Messing about and experimenting are part of the process of creating. In the world of writing this is the pre-writing experience. That process preceding the actual writing should be of such a nature, be so compelling that the student can't help but want to write. This notion is equally applicable in the developing of a Logo culture or a Logo learning environment. Drill and practice and convergent style tasks (everybody make a square using FORWARD 40) and fill-in-the-blank worksheets are not as likely to bring about broad-reaching results.

In my classroom, both the Turtle Center and the Writing Center had the process steps posted in plain view for all to see. These centers were designed to provide stimulus for users, particularly if the child desired some focus for a project they were wanting to try. For a writer or a young programmer, a journal is a crucial ingredient in focusing the author's thoughts. Whatever students deem valuable can be stored inside. Theories, patterns, new ideas, designs, bugs, can all be kept "for reflection, remembering, and refinement" (Watt, 1986). The journal provides the teacher with a glimpse into the student's mind — or what Weir (1987) refers to as "an empirical window." Journals should not be graded or corrected (Rifkin, 1983; Watt, 1986).

3. Composing and Exploring

Composing includes selection of information and mechanics in order to assemble parts into a whole. In the world of writing there are many techniques employed to generate a story. Some writers might engage in "free writing" (Elbow, 1973) or stream of consciousness writing. This is akin to what we might call the antiplanner, the improviser, or the bottom-up style programmer. Still others might follow an outline or a model of some previously written story — perhaps in the style of another author — much like the language experience approach used in the following story and language patterns. The most important contributions the teacher can make in this phase are

1. Model using Logo. You should be seen developing your own applications. Good teachers of writing are themselves writers. So too, with Logo. A good Logo teacher should be a good Logo user.
2. Serve as coach, collaborator, guide, and facilitator — not the focal point of the activity but helping the child to be focused. (Watt, 1986) One obvious and generally successful technique is teaching the child to break the problem into smaller, more manageable subproblems or subprocedures. Another is to teach the child to word backward. Still another is to simplify the project and then embellish it rather than starting off at too high a frustration level.

4. Reading and Acting

Reading the written text is part of the post-writing process. In the programming process, reading refers either to actually seeing one's program run or stepping through the program to hunt down errors and/or to check the appropriateness of the program's content. Many students working with Logo tend to get in over their heads very quickly. This is not a negative thing. When a student stumbles into a surprise (recursion, simple typographical errors, etc.) this becomes an optimum occasion for informal teaching. It is here that the teacher, based on a personal knowledge of the student's needs and interests can best intercede. Using a mixture of Bransford's Model and a debugging format described by the Watt's (1986), the teacher can use meaningful metaphors and provide useful models. For example, the teacher can create situations and opportunities for the student to describe the problem, or recreate the bug, or re-experience the bug in another way by playing turtle or computer or sketching the desired outcome graphically, or by making a procedure tree. (See last month's column Revise the orchard to have autumn-leafed trees with possible debugging strategies to help students increase their yield and harvest better results.)

5. Revision and Looking at the effects

Revision, literally "seeing again," can range from a simple adjustment for a small bug to making a major addition, deletion, and/or a reorganization of a set of subprocedures. If students never participate in the evaluation of the results, they will fail to learn — to accommodate and assimilate new information — to be *real* and *active* learners.

Your children can be taught to adopt this approach through a developmental process which is broken into three phases: an introductory phase, an implementation phase, and an integration phase. In the first phase it is the teacher's responsibility to guide the student orally through each step of such strategies as IDEAL or the writing process. In the second phase the teacher's responsibility is to refer the student to the model, giving him/her the main responsibility of working through the steps. The third phase is the one in which the student can independently initiate use of the model. As a part of the training process with the children, posters are made (preferably by the students) which are visible throughout the classroom, listing such items as IDEAL, the steps of the writing process, and enumerating potential debugging strategies. Early fall is the perfect time to begin introducing these strategies to your students.

The following scenario typifies an application for the *Composite Process Model* to a classroom context. In the introductory phase the teacher might begin an introduction to a creative writing class by reminding the students that they will be using IDEAL to generate a group Logo project. The teacher could ask what the first step of IDEAL is and pause, giving the students an opportunity to reflect. Hopefully, the child that responds will explain that the first step of IDEAL is to identify the problem. The problem at hand is not having a topic for their graphic design. The teacher might choose to ask if anyone has any ideas for topics or to create parameters by giving a thematic focus to the assignment. For example, this month's thematic study is based on the bounty of fall harvest. Kick the unit off with a visit to a nearby grain elevator, a farm, or an animal feed store. October is a good time to to rotate the learning center from September's column to become a new classroom attraction. Some possibilities are County Fair, Scarecrow Scene, the Pumpkin Patch, or the Corn Corner. Science and social studies lessons can be combined for class investigations into discovering the number of food products that we use which are made from corn. Displays of actual items or collages from magazines can be compiled. Study the three different types of corn: animal or field corn, sweet corn, and popcorn. Grow some real corn in the classroom. Pop some corn. Use the kernels of last months FIELD.TRIP tool kit to make a collection of corny jokes. As a culminating activity have a corn roast or a full-blown county fair.

Various students would then be called upon while suggestions are placed on the board. Students would ultimately divide themselves into teams (or be divided by the teacher) to work on the topics in which they were most interested. While students work in subgroups further defining their project goals or actually carrying out the assigned tasks, the teacher could conference with students needing greater focus or assistance.

Students would be seen engaged in a lot of discussion as they begin exploring their ideas at the keyboard. The teacher would remind these students to use "act" and "look" or "read" and "revise" while composing and creating. The teacher's goal is to help the group act on strategies, try out ideas, and look at the results. The remaining class session, a class meeting, or a small special interest group provide opportunity for revision and polishing.

This is one small example of using the *Composite Process Model.* Certainly, as the year progresses it is in the student' best interest to give them increasing opportunities for autonomy and decision-making. Students who need the on-going support or more direct guidance of a teacher, can and should receive it. However, appropriately applied, this model should help students become effective and empowered self-starters.

Teachers obviously need to subscribe to this approach philosophically for it to be effective. There are some presuppositions to the approach as well. Learning a relatively few reasoning techniques will enable an individual to deal effectively with most types of problems. However, learning to reason requires interaction with someone else. Very few individuals are able to critique their own reasoning. This is especially true of children. Children, therefore, need as much opportunity as possible to test the correctness of their reasoning with others. The major barrier to effective problem solving is more often psychological than intellectual. Many students are so accustomed to having someone else solve their problems, or having no "real" ones to solve, that they simply do not try to break through and solve them on their own.

Creating some basic guidelines or standards also helps enable students to engage effectively in this process. One example is a rule that I developed some years ago. This is the "Three Before Me" rule which requires children to ask for

Logo for Little Kids -- continued.

help from three peers before they ask a teacher. The two other rules I employ are: you are responsible for your own thinking and behavior; and you must be willing to help anyone who asks.

In my experience, the outcome of following this *Model* goes beyond enhancing the students' self-concept and teaching them to be effective thinkers. It creates a whole culture which is self-supporting. A few potential social outcomes are learning how to deal with people who are different, acquiring patience, learning to be respectful of others, seeing a classmate in a new light, and developing self-reliance. When working with others, each students gets many opportunities to be both the teacher and the student. This helps students realize that solving together is often more efficient and enjoyable than working by themselves. This format reduces a teacher's class load because the students are helping each other. Students and teachers alike get immediate feedback. Because it is generic and flexible, the individuality, integrity and dignity of both teacher and child remain intact.

"An economy based upon people who think for a living requires schools dedicated to the creation of environments in which students become very adept at thinking for themselves, place where they master the art of learning and acquire a strong taste for it " (Carnegie Task Force on Teaching, 1986, p. 45). Empowering students to perceive themselves as learners and to be able to act on that perception is what I believe our role as educators to be.

References:

Bransford, J. and Stein, B. (1984) *The IDEAL problem solver: A guide for improving thinking, learning, and creativity.* New York: W. H. Freeman & Company.

Calkins, L. (1978) Children write — and their writing becomes their textbook. *Language Arts, 55,* 804-810.

dePaola, T. (1978) *The popcorn book.* New York: Scholastic, Inc.

Elbow, P. (1973) *Writing without teachers.* New York: Oxford University Press.

Rifkin, B. (1983, Jan.) Turtle folders help third graders. *National Logo Exchange.*

Thyberg, L. (1988) *A descriptive analysis of student behaviors and attitudes using Logo programming as a vehicle in the writing process.* Doctoral dissertation. University of Pittsburgh.

Watt, D. and Watt, M. (1986) *Teaching with Logo: Building blocks for learning.* Menlo Park: Addison-Wesley.

Weir, S. (1987) *Cultivating minds: A Logo casebook.* New York: Harper and Row.

I welcome your correspondence and comments. This column is after all, for you. If you have a neat idea — send it to me! I'll cite you and you can brag to all your friends that your are now famous and widely read.

Dr. Leslie F. Thyberg, 5637 Rippey Street
Pittsburgh, PA 15206, AppleLink ALS 038

Assessing Logo Learning

Procedures and Subprocedures
by Dan Watt

This is the second of nine columns based on a research project which Molly Watt and I have been carrying out with support from the national Science foundation, "Exploratory Research on Critical Aspects of Logo Learning." In this project, we collaborated with a group of experienced Logo teachers to identify critical aspects of Logo learning and group them under eight headings which were listed in September's column.

This month I will talk about what we mean by each of the subheadings included in the first of our critical aspects, **Procedures and Subprocedures**, illustrated with examples of student work. For a fuller sense of what we mean by critical aspects of Logo learning, and our rationale for an approach to assessing Logo learning, please read the September '88 column in this series.

**Procedural Thinking —
The Importance of "Mind-Sized Bites"**

Essentially, a procedure is a recipe, a step-by-step set of instructions designed to achieve a specific result. Logo allows us to give procedures any names we want and allows any procedure to call any other procedures as subprocedures to carry out any task. This is Logo's core strategy for helping us solve problems! In its simplest form, this means we can divide a large task into smaller parts, give each part its own name, and solve each small piece of a larger task, one at a time. Of course each subprocedure, once named, can call on still other subprocedures, to solve smaller parts of the problem, and so on.

In *Mindstorms*, Seymour Papert quoted a seventh grader, Robert, who had discovered the usefulness of this approach: "See, all my procedures are mind-sized bites. ... I used to get mixed up by my programs. Now I don't bite off more than I can chew." Dividing problems into mind-sized bites allows us to limit and identify the small piece of a problem that we are working on at any given time. Giving each piece an understandable name, allows us to come back to find any particular piece again when we want it, to reuse it when appropriate, and to debug larger problems more effectively, by identifying the exact subprocedure in which a bug occurs.

All of this makes a lot of sense to us as teachers — but it doesn't always make sense to students. Lots of teachers and researchers report that their students don't use procedures and subprocedures effectively in solving problems. So part of our job as Logo teachers is to see that our students learn these ideas — not as abstract principles which they are made to memorize for a test — but as useful techniques that can make their own work easier, and more understandable for themselves and others.

This last point is both subtle and important. In any creative work, there are always at least two audiences, the creator himself, and others who may be interested in both appreciating and using that work (or some of the ideas in it) for themselves. As teachers, one of our overriding tasks is to create a community of learners who can build on ideas created by others and can contribute ideas for others to use. The importance of this may be temporarily lost on a student (or on any of us) in the midst of a struggle to solve a particularly thorny problem. But part of the genius built-in to Logo is that when a Logo procedure is clearly written, with appropriately defined and clearly named subprocedures, it is readable by anyone, even by someone with less Logo knowledge than its creator. As teachers we can help students keep this in mind as they revise and complete their work. Just as writers many need many drafts to communicate their ideas as effectively as possible, most Logo programmers, even the most experienced, usually have to revise their work to make more understandable for themselves and others.

In our work with teachers, we have identified seven different levels of sophistication in using procedures and subprocedures. However, students don't always learn to use procedures in the particular order we have set out, and may not always function consistently at one level. For example, we have noticed that many students learn to write clear, well-articulated superprocedures, while still using sequential programming style for all their subprocedures. Or, a student may use one or two procedures as tools, without really understanding modular programming. Or students may learn to use tail recursion in its simplest form, before understanding much about other aspects of procedures and subprocedures.

Here are the headings we've developed for describing learning to use procedures and subprocedures. Some examples will follow.

A. Pre-procedures:

Pre-procedures involves using the *idea* of a procedure without actually creating one. For example, repeating the steps needed to draw a square, several times, without defining a square procedure. Or, using a series of commands over and over in a procedure, without defining a particular subprocedure. The learner is clearly identifying a sequence of commands that is used more than once for a particular purpose, but does not choose to (or does not know how to) define a procedure for that purpose.

B. Using REPEAT:

For some students, using REPEAT may be another form of pre-procedure. Although it does not usually involve naming, it is often the first situation in which students realize that a particular set of instructions can be used repeatedly to produce an effect more easily than by typing a long sequence of commands. For example, drawing a square by typing REPEAT 4 [FORWARD 50 RIGHT 90], in one line, rather than typing 8 separate commands. Beginning Logo students often use REPEAT to produce dramatic or unexpected effects, without necessarily noticing either the number of repeats or the sequence of commands being repeated: REPEAT 10000 [FORWARD 1234 RIGHT 4567]. This use of REPEAT does not necessarily lead to procedural thinking!

C. Sequential, undifferentiated procedures:

This involves taking a sequence of commands that produced a desired effect in direct mode, giving them a name, and typing them into the editor. This mode is characterized by long command lines, with many commands to a line, and by frequent typing errors that occur while typing commands from notes or memory. Students may use idiosyncratic names for procedures: TO A, TO ME, TO FLUFFY.

D. Use of subprocedures:

The use of subprocedures begins with giving separate names to procedures that draw different designs, or accomplish different parts of a task. Names begin to reflect purposes: To TRIANGLE, TO HEAD, TO LEFTLEG. Some students continue to work sequentially, giving separate names to each part of a drawing. Others create shapes separately, and make more complex designs by rotating and combining them. Students do not always distinguish between the parts of a procedure that draw a design, and the parts that move the turtle into place. Therefore, interface bugs (parts of designs out of place) are common when subprocedures are first used.

E. Structured programming styles:

Structured programming usually involves planning. Students decide in advance what parts will be needed for a given project, and choose names that reflect those parts. Naming becomes more thoughtful and complex. Procedures that move the turtle may be defined and named separately from procedures that draw parts of designs. A structured programmer may start with a fairly definite plan, but the plan (and naming strategies) are usually modified in the course of work on a project. Some programmers do minimal planning in advance, create a program sequentially, and then impose a structure in revising the project after it is partially or totally working.

F. Modular programming styles:

Modular programming involves the use of the same subprocedures in more than one setting. For example, the use of a circle procedure as a tool for drawing a particular design. Or the creation of a design with interchangeable parts (a face that can have different sets of features). Sometimes procedures made for one purpose may be used for another. It is wise to make modular procedures state-transparent so that they can be used in any position or for any purpose. Modular procedures with variable inputs are more flexible and powerful than procedures with fixed inputs.

Assessing Logo Learning in the Classroom -- continued

G. Procedures using recursion and conditionals:
It is not clear where these fit in the hierarchy. They have a range of sophistication of their own: simple tail recursion; recursion with conditional stop rules; recursion with incrementing variables; embedded recursion. Few students in grades 4 -6 deal with embedded recursion. I'll say a bit more about recursion in next month's column, dealing with variables.

A Note About Problem-Solving Approaches
Students' use of procedures and subprocedures are closely related to their preferred problem-solving approaches. We can talk about two major problem-solving approaches, top-down and bottom-up, although most of us use some combination of these in any actual work. Top-down styles involve starting with a specific goal, breaking a problem into parts and solving it systematically, part by part. Bottom-up styles involve building up a complex structure, bit by bit, without necessarily knowing in advance what the eventual result will be. Many students have a preference about which approach to emphasize in their problem solving, and this shows up in their Logo work. (In *The Second Self*, Sherry Turkle calls these preferences "hard mastery" and "soft mastery," and has some excellent illustrations of the power of both approaches used in Logo.) I'll have more to say about this in the column on planning and carrying out projects. For now, I'd just like to point out that it is more common to use combinations of these styles in solving a particular problem, rather than to use either of them in their purest form.

Examples of Student Work Using Procedures
1. Kathy's toothbrush (grade 4):

```
TO HANDLE
 FD 70
 LT 90
END
```

```
TO TOOTHBRUSH
 FD 7
 BK 7
 LT 90
 FD 4
 RT 90
 FD 7
 BK 7
 LT 90
 FD 4
 RT 90
 FD 7
 BK 7
 LT 90
 FD 5
 RT 90
 FD 7
 BK 7
 LT 90
 RT 90
 FD 7
 BK 7
 RT 90
 LT 180
 FD 4
END
```

Figure 1

Kathy's procedures illustrate *sequential, undifferentiated programming*, as well as the use of *pre-procedures*. Although she has two separately named subprocedures HANDLE and TOOTHBRUSH, both of them must be typed to draw the figure shown. Notice that she uses the sequence FD 7, BK 7, LT 90, FD 4, RT 90, repeatedly, without using either REPEAT, or a subprocedure. This is characteristic of *undifferentiated programming*, as well as a nice example of the use of *pre-procedures*, within an existing procedure. (Readers might be interested to know that Kathy had been taught and encouraged to use subprocedures and had used them effectively with her teacher's assistance in other projects. But here she shows clearly that she has not yet mastered their use.) Notice also the literal incorporation of typing errors, and errors made in direct mode, into the procedure: using FD 5 rather than FD 4 (I'm *guessing* this was an error, given the regularity of a toothbrush); using LT 90 RT 90 without FD 4 between them, and using RT 90 LT 180, without replacing them by LT 90. Once again, these are characteristic of the sequential, undifferentiated mode of Logo programming. It is not easy for Kathy to debug her work!

2. Brad's City (grade 4):

```
TO CITY
 SQUARE HT PC 5
 REPEAT 45 [FD 1 RT 90 FD 90 LT 90 FD 1 LT 90 FD 90 RT 90] HOME
 LT 90
 FD 908 RT 90
 RECTANGLE
 LT 90 FD 20
 RT 90 RECTANGLE
 FD 20
 RT 90
 FD 40 LT 90
 FD 20 LT 90
 FD 40 RT 90
 FD 20 RT 90
 FD 40 LT 90
 FD 20 LT 90
 FD 40 RT 90
 FD 20 BK 40
 BK 60 RT 90
 FD 80 FD 15
 LT 90 STREET
END

TO SQUARE
 REPEAT 4 [FD 90 RT 90]
END

TO RECTANGLE
 FD 109 RT 90
 FD 20 RT 90
 FD 109
 RT 90
 FD 20
 RT 90
END

TO STREET
 RT 90
 FD 1000
 RT 90
 PU
 FD 70
 RT 90
 PD
 FD 1000
 RT 90
 PU
 FD 35
 RT 90
 REPEAT 20 [FD 10 PU FD 10 PD]
END
```

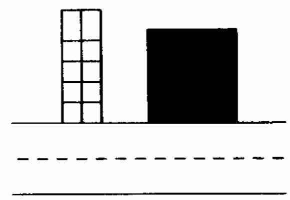
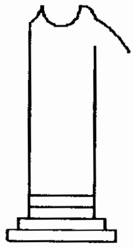


Figure 2

Brad's CITY shows an early stage of development in a series of related projects that were extended and elaborated for a period of months, while he also worked on other projects. Brad's work illustrates some use of *subprocedures* and *modularity*, mixed with *sequential programming*. For example, Brad used subprocedures to draw a SQUARE, RECTANGLE, and STREET. But he has not used subprocedures to identify either of the two houses within his CITY procedure, or to separate the parts of his pictures from the commands that place them on the screen. And he does not use a subprocedure or repeat for the repeated sequence of commands that draws the criss-crossed design on the taller building. *Modularity* is hinted at here by the re-use of the same subprocedure, RECTANGLE, in two places. Modular programming showed up a lot more in later CITYs. Brad's three uses of **REPEAT** (to draw a square, to fill in a square, and to draw a dotted line) indicate a good mastery of that command and its possibilities. (Much of Brad's other work consisted of explorations using REPEAT in many different ways.)

3. Heather's Statue of Liberty (grade 6):



```
TO SECRET
 PLATFORM
 BODY
 HEAD
 CROWN
 TORCH
 BOOK
END

TO PLATFORM
 HT PU HOME BK 119 RT 90 FD 50 LT 180 PD
 VRECT 100 10
 PU FD 10 RT 90 FD 10 LT 90 PD
 VRECT 80 10
 RT 90 FD 10 LT 90 FD 10 PD VRECT 60 20
 VRECT 60 10
END

TO VRECT :W :L
 REPEAT 2 [FD :W RT 90 FD :L RT 90]
END

TO BODY
 DRESS
 ARMS
END

TO ARMS
 SETH 120
 REPEAT 15 [FD 3 RT 2]
END

TO HEAD
 HAIR
 NOSE
 MOUTH
 EYES
END

TO CROWN
 BASE.OF.CROWN
 WINDOWS
END

TO TORCH
 HANDEL
 BASE
 FLAME
 HAND
END

TO BOOK
 WRITING
 HAND
 OUTLINE
END

TO DRESS
 HT RT 90 PU FD 20 PD FD 130 BK 130 LT 90 FD 60 RT 90 FD 150 RT 55
 REPEAT 20 [FD 1 LT 2] SETH 90 SETH 180
 REPEAT 19 [FD 3 LT 10] SETH 90 REPEAT 20 [FD 1 RT 4]
END
```

Figure 3

Heather and her partner, Joanie, set out to make a turtle drawing of the Statue of Liberty. The example here shows just one stage of the work in progress. (We'll come back to this project in the column on Planning and Carrying Out a Project.) The girls' superprocedure and subprocedure names illustrate the use of *top-down problem-solving*, and *structured programming* styles. *Modularity* is shown by the use of VRECT, a variable rectangle procedure borrowed from an earlier project. The separation of the command lines in PLATFORM, into lines that move the turtle with its pen up, and lines that draw parts of the picture, are a further indication of an awareness of modular programming styles. Both the PLATFORM and DRESS subprocedures, however, indicate that *sequential, undifferentiated procedures* are still very much part of Heather's and Joanie's approach to Logo. They do not seem to be thinking about another whether another reader could readily understand how their procedures work, or whether they themselves might have difficulty debugging them someday.

Some Closing Thoughts

Each of these examples, and the example of Jimmy's "yin-yang" design given last month, illustrates work in progress — typical of many children's Logo work in upper elementary grades. Each example shows some use of Logo's powerful ideas involving procedures and subprocedures, and also shows that some potentially useful ideas were ignored or only partially utilized.

I leave it to you to think about what you would suggest to Kathy, Brad, Heather and Joanie, or Jimmy, if you were their teacher. Specific decisions about how and when to intervene with each child depend on the teacher's best insights about which learnings about procedures and subprocedures (if any) would be most useful and productive in helping individual students meet their own immediate goals. She'll also have to consider which of these best meet her long range learning goals for her Logo students, and what types of teaching intervention works best for each individual student. What I have shown here is that there is a lot to be learned by taking a detailed look at student work in process. I hope you'll join the discussion by letting me and readers of this column know how you've been looking at your students' work.

References

Hillel, J. and Samurcay, R. (1985). "The Definition and Use of General Procedures by 9 Year Olds," Research Report #2, Department of Mathematics, Concordia University, Montreal, Quebec.

Leron, U. (1983). "Some Problems in Children's Logo Learning," in Hershkowitz, R. (Ed.). *Proceedings of the Seventh International Conference for the Psychology of Mathematics Education*, Weizmann Institute of Science, Israel.

MacDougall, A. (1986). "Children's Difficulties in Perceiving Structure and Using Subprocedures," *Proceedings, Second International Conference for Logo and Mathematics Education*, University of London, Institute of Education, London, U.K.

Papert, Seymour (1980). *Mindstorms*, Basic Books, New York NY. (See Chapter 4, "Languages for Computers and for People.")

Papert, S. DiSessa A., Watt, D., Weir S. (1979) *Final Report of the Bookline Logo Project*, Logo Memo # 53, MIT Logo Group, Cambridge MA.

Turkle, Sherry (1984). *The Second Self*, Simon and Shuster, New York, NY (see Chapter 3, "Child Programmers: The First Generation.")

Watt, D. (1979). "A Comparison of the Problem-Solving Styles of Two Students Learning Logo," *Proceedings, National Educational Computing Conference*, 1979; *Creative Computing*, December 1979.

Watt, M. and Watt, D. (1986). *Teaching With Logo*, Addison-Wesley, Menlo Park CA.

Daniel H. Watt
Educational Alternatives
Gregg Lake Road
Antrim, NH 03440

Logo: Search and Research

Problem Solving: Beauty and the Beast

by Douglas H. Clements

A great discovery solves a great problem but there is a grain of discovery in the solution of any problem. Your problem may be modest; but if it challenges your curiosity and brings into play your inventive faculties, and if you solve it by your own means, you may experience the tension and enjoy the triumph of discovery. Such experiences at a susceptible age may create a taste for mental work and leave their imprint on mind and character for a lifetime.

George Polya

There is beauty in these words. It is not difficult to find teachers who resonate with their beauty. When observing these teachers in action, however, it is more difficult to see specific instructional actions that are designed to develop problem solving abilities. In fact, most of us teachers have difficulty even describing what program solving is, or how we might help children master it. It's a beast to understand.

Teachers are not alone. Most people, including curriculum designers, professors of education, and psychologists, struggle to grasp the essence of this elusive ability. Teachers, however, must teach children *today*. And research has uncovered much about problem solving that is useful. This column marks the beginning of a series of research reviews on problem solving, including suggestions regarding how children's ability to solve problems might be developed. The first two columns will deal with understanding and teaching problem solving, with or without computers. Research on *Logo* and problem solving will follow. Problem solving is initially discussed separately for two reasons. The first is that it *is* difficult to understand and to teach. Discussing it in isolation allows us to tackle one problem (!) at a time. Second, it is especially important to integrate and connect computer and non–computer experiences. Therefore, a unified framework is necessary for understanding and teaching problem solving with and without Logo.

The Nature of the Beast

"What is 7 times 4?" Is this a problem? Or merely drill? What if it were couched in a verbal format such as: "Seven friends each have 4 books. How many books do they have in all?" If a student can simply multiply and say "28," has he or she solved a problem? In one sense, yes, of course. In another sense, however, there was no problem, only an application. Can this example ever be a "true" problem? Consider giving some books to a group of first graders and asking them the same question. Some may act out the situation. Some may count on their fingers. If if they make a mistake, have they not engaged in problem solving?

Solving a problem that one "already knows how to solve"—an application—is an important part of mathematical ability. But we also want to stress "true" problems. These are situations in which a person wants to reach a goal but does not know immediately what actions he or she can take to achieve it. This type of problem solving is, in Grayson Wheatley's words, the ability to "know what to do when you don't know what to do." Finally, note that 7 x 4 can never be a problem for students unless they want to know the answer (i.e., they *accept* the problem). Let's see what research in cognitive psychology has to tell us about how people learn to solve problems; to know what to do, even when they don't (immediately) know what to do.

How Do People Solve Problems?

Sternberg (1985) hypothesizes that different types of processes are carried out by separate components of people's information-processing systems. Each component has a special function, or job, which it performs. It is as if each played a role in a drama or business company. Tasks are completed through the cooperation within the company. For example, in solving the transitive inference problem (or syllogism), "Mark is taller than Pete; Bob is shorter than Pete; who is tallest?" one component needs to encode, or take in, the information by reading. Another might represent the first two statements as spatial arrays (i.e., a mental "picture" showing Mark as taller than Pete and so on). In this way, the mind is seen as a company in which a group of separate members—components—work with each other. The metacomponents—the "executives" of peoples' "mental company"—plan and evaluate all one's information processing. So, they are critical to successful problem solving. They include:

- deciding on the nature of the problem (e.g., understanding that the problem is to order the three people and find the tallest);
- selecting a representation for the problem (deciding to draw lines representing each person or to visualize them mentally);
- selecting a strategy for combining performance components relevant to the solution of the problem (deciding to order the first two people, place the third within that ordering, and determine which is the tallest); and
- monitoring solution processes (keep track: Am I answering the right question? Can I visualize everything without getting mixed up, or had I better switch tactics and write it on paper?).

Global News

Taming the Beast

In the next two columns we shall examine these meta-components in more detail and discuss teaching techniques that research has identified as effective in developing these abilities. Until then, let's enjoy another glimpse of problem solving's beautiful side, with the another passage from Polya:

> Thus, a teacher of mathematics has a great opportunity. It he fills his allotted time with drilling his students in routine operations he kills their interest, hampers their intellectual development, and misuses his opportunity. But if he challenges the curiosity of his students by setting them problems proportionate to their knowledge, and helps them to solve their problems with stimulating questions, he may give them a taste for, and some means of, independent thinking.

Reference

Sternberg, R. (1985). *Beyond IQ*. Cambridge: Cambridge University Press.

Douglas H. Clements, SUNY at Buffalo
Department of Learning and Instruction
593 Baldy Hall, Buffalo, New York 14260
CIS: 76136,2027 BitNet: INSDH@UBVMSA

In Memorium

LX has received notice of the death of FredKlotz. He died last April in Dublin, Ireland as a result of injuries from a cycle accident. Fred will be remembered by those in the Logo community as the organizer or the first European Logo Conference in 1987.

Bob Sparkes and Fred Klotz

Fred was a pioneer in the use of microcomputers in education in Ireland. Many Irish primary teachers received their introduction to computers from Fred. His professional links both in Ireland and throughout the world enriched the entire computers in education community. He will be missed.

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

Fishing with Logo in Costa Rica

by Hillel Weintraub
With assistance from Marta Santamaria Marin, Fred Martin, Nuria Jiménez Montero, and Mitch Resnick

> If you give a person a fish, you feed her for a day;
> If you teach a person how to fish, you give her the power to feed herself for a lifetime.
> *a*ncient Chinese saying

In 1985, when Oscar Arias was running for President of Costa Rica, he made a number of campaign promises. As we are now aware, Arias, this year's Nobel Peace Prize winner, is a man who tries to turn his rhetoric into reality. One of his promises was that he would try to improve the educational system in Costa Rica, and more specifically that he would introduce computer technology in the schools as one means of working towards this goal. Since campaign promises are to most politicians what vaporware is to most software manufacturers, the public tends not to pay much heed to them. But people in Costa Rica have come to expect more of Oscar Arias, and, indeed, after his election in 1986, Arias began to take immediate steps towards building his vision.

The first thing he did was to form an educational commission which announced that it was interested in getting proposals from companies for creating a program which would utilize computers in an educationally sound way. IBM approached Dr. Seymour Papert and a group was formed at MIT composed of Dr. Papert, Marilyn Schaffer, Steve Ocko, Mitch Resnick and Fred Martin, all of whom are involved in a variety of ways with Logo at the Media Lab at MIT. Thus the innovative educational project, called *Genesis,* was born.

The information in this report was provided by Mitch Resnick and reviewed by Fred Martin who also supplied helpful editing suggestions as well as photographs. Fortunately, Marta Santamaria Marin and Nuria Jiménez Montero were available to share their perspective on the training and the program thus far. The responsibility for accuracy and the unquoted opinions expressed are the author's.

The reason why the *Logo Exchange* has chosen to give so much space to reporting on this project is because we feel that there is much exemplary about it, from the top down planning and support from the government to the bottom up grass roots pride of the Costa Rican people.[1] Furthermore, as a model of cooperation between government, industry, university and public schools, this project, now called *Genesis,* has much to teach us. And finally, as an example of an

Global News -- continued

educational project which places great emphasis on the key element for long term success— the teachers— it needs to be looked at it carefully.

The group from MIT wrote a proposal which outlined a philosophical framework and a vision of how computers could be used in the classroom, and emphasized teacher training as a major factor for the success of their program. After visiting various educational institutions around the U.S., members of the Costa Rican government and Ministry of Education, who by now had formed a foundation under the support of the Agency for International Development (AID), decided to accept the proposal from the IBM and MIT group, who then proceeded to implement this program late in 1987.

Phase One

In the first phase a group of 12 educational leaders from Costa Rica came to MIT to spend two weeks working intensively with Logo and the philosophy of education which might be characterized as a constructionist, discovery-oriented approach to learning. These educators came from different fields and their home institutions ranged from the national university to public elementary schools. But what was most striking about this group was that they all had made a three year commitment to work with *Genesis*, and that the government and their institutions had given them full support. To our knowledge, that is an unprecedented depth of governmental commitment, showing that Arias and his Ministry of Education were behind this idea with actions and money as well as words.[2]

The foundation in Costa Rica made it clear from the start that this group of twelve educators, who were designated as teacher trainers, would have an extremely important role in this project. During the initial two week intensive training at MIT, the team of twelve met with the five from MIT to discuss philosophy of education in general and the philosophy of Logo in particular, and to observe the classes of children who had been using Logo in a Boston inner city school where MIT has been conducting their School of the Future program for the past three years. However, most of the time was spent in working directly with Logo, specifically LogoWriter for IBM PC's.

About half of the team from Costa Rica was able to communicate well in English, so consideration had to be taken about translating parts of the materials and discussions. This had certain advantageous aspects in that the MIT team had to be constantly thinking about how and what they were communicating, never taking for granted that they were being understood, certainly a wise attitude for any people interacting in a learning environment.

After the two week program at MIT, the twelve teacher trainers returned to Costa Rica, where they had a week to digest all the new ideas and materials to which they had been exposed. Now they could get ready for Phase Two: helping their compatriots learn about a discovery oriented approach to learning and about Logo.

Phase Two

Because the grant from IBM called for implementation of this program in stages, a set number of schools were to be be brought in each year. The total number of computers in the proposal was 4,000 IBM Personal System 2's. In addition there was to be networking of computers and the addition of educational software. By the end of the summer, 1200 of the computers had been placed in 60 schools, but even more important is the number of teachers that will have been introduced to Logo and this new style of learning and who will *continue* to be supported in their work at both a local and national level.[3]

Having a structure which supports the philosophy of a program so directly relates to the success of that program, and here, too, *Genesis* showed much foresight. In Costa Rica, many schools had expressed interest in becoming part of *Genesis*, since each participating school would be given a grant of twenty computer systems. The Foundation in Costa Rica had to make decisions about who would initially take part in the program in ways that were congruent with the philosophy of their program.

Therefore, it was made a condition for any school that wished to become part of this project to send their principal and two teachers to be trained.[4] The two teachers would then become the computer coordinators who would be responsible for the computer labs and for training other teachers in the school. The principals would be able to interact with parents and other teachers with an understanding and support of the program. Furthermore, it was held to be important that the training not be pushed into a weekend or even week-long session. The three people attending from each school were given release time to attend the intensive programs, which went from 8-4 for three weeks.[5]

One of the important considerations in planning the training sessions at MIT was to keep in mind how the Cambridge sessions would relate to the Costa Rican teacher sessions and how everything would in turn relate to the classrooms with children. After the program at MIT, the TT's (Teacher Training group) would return to Costa Rica and make the important switch in roles from those who are being taught to those who are teaching others. The philosophy of education which Logo emphasizes minimizes this distinction, pointing out that both teachers and students are always learning and that traditional ways of looking at roles more often than not get in the way of learning through discovery or construction. The MIT group put a lot of emphasis on the idea that everyone was learning together. This helped the TT's return to Costa Rica with confidence that they could "learn together" with the people they would be working with in-country.

The sessions in Costa Rica followed the same general structure as the training at MIT, but with the TT's now taking the role of the MIT people, and the MIT group, as planned, making their own roles less and less important as the training went on until the present time when the teacher training sessions are continuing without any MIT presence other than in spirit.

The issue of transfer of responsibility and leadership, had not only educational ramifications, put political and historical ones as well. Everyone, from Arias to the TT's, recognized that it was essential for the educators involved to appropriate as much as possible the value of Logo-like learning. Having this idea imposed from the outside would insure its failure, so it was fortunate that the educators in the initial group had a basic sympathy for open style, discovery oriented learning. This made the work of the MIT group, not so much one of convincing or "teaching" the TT's but creating environments which would enable them to add depth and width to their own experiences.

Two of the TT's were asked about the relationship of the MIT group and the TT's when they moved to Costa Rica. Marta Santamaria Marin said that during the first in-country program to introduce teachers to Logo, "I was the teacher and Mitch (Resnick) was my assistant." Nuria Jiménez Montero noted that "Seymour (Papert) and the others were just like a family to us. We really helped each other."

Marta and Nuria's Evaluation

It was interesting to get the evaluation of Marta Santamaria Marin and Nuria Jiménez Montero, teachers who have been involved with *Genesis* from the beginning. They felt that it was difficult to measure the children's progress in traditional ways because they had only been using Logo for a few months. "They are still playing with the machine and getting used to the computer and Logo." But no one seemed to be pushing the teachers to measure the children by quizzes or tests. The concept of meaningful learning activities is being redefined. Nuria announced. "We want to elevate the quality of education, not grade the children for the use of computers."

Instead of talking about how many commands or how much knowledge kids seemed to have, Nuria and Marta shared some stories which they felt showed the value of *Genesis*.

Nuria observed that the teachers and principals who attended the first training sessions were very excited by the new approach to learning and seemed to appropriate the ideas and adapt them to their own interests and subject areas. In addition, they expressed great pride in being part of a larger vision that Arias has been trying to bring into reality in his country by focusing on educational rather than military applications of intelligence and capital.

Marta's story: In the capital I saw a boy without any shoes. I thought, "poor kid." Then I saw his arms were paralyzed. The teacher put the keyboard of the computer on the floor and he could type with his big toes. For the first time, he could feel equal to his classmates!

They both told of some of the feelings that they could notice in children and in teachers which they considered valuable.

> Our children feel as though they are holding the future in their hands when they use the computer.
>
> Computers can help the child be better prepared for the future, and they can feel equal to kids in the U.S. and other countries.
>
> Kids who couldn't draw before, can now enjoy it. The computer is like a tool for them to do something they couldn't do before.
>
> Paired work is really benefiting the quiet kids who had little confidence.They are getting support from each other.[6]

When pressed to find some problems, Marta and Nuria pointed out that getting teachers to change some of their attitudes about "learning" was the most difficult matter. "In class, the teacher is chief! Everyone follows the teacher," is both the feeling of the teachers and the expectation of the students and parents. In the computer labs, however, teachers were encouraged to act like guides and to create space for the children to act like guides as well. This could be one important advantage to having the computers in a separate room: it is easier for teachers to change their behavior in an unfamiliar environment and hopefully, when seeing the results of different actions, some of those behaviors might be brought back into the classroom.

Another "problem" was that children tended to enjoy being and working in the labs so much that they were reluctant to return to their classroom. Of course, this would also have some effect on teachers, though it might not be so positive at first.

Marta and Nuria admitted, "Of course, there are problems. Each training session for teachers gets better and better. For example, in the early training programs, we used to give handouts to the teacher at the beginning of each session, and we noticed that they weren't paying attention during the meetings. They just wanted to copy what was on the handout and go on to something new. It was very difficult to get them to explore. Now they try things out and learn, and then we give them a handout!

Each program changed, with explanations becoming clearer and clearer. The trainers became less tight, more confident, and more active and playful. At the end of each day the teachers and their tutors got together talking about methodology and problems and these sessions also set a good tone for the idea of learning together that, it was hoped, would return to the schools.

The idea of giving kids the power to learn for themselves attracted many teachers who were attending the training

Global News -- continued

sessions. During a recent program, one teacher made the connection between what he had learned about educational philosophy and the famous Chinese proverb about the difference in giving someone a fish and teaching her how to fish. "Logo," he said, "is like a fishing rod!"

Training for Discovery Learning: A Contradiction in Terms?

The idea of "training" is somewhat problematic. Can we "train" teachers to "do Logo?" Can we "train" people to create empowering learning environments? Difficult questions with no easy answers, but ones that need to be asked nevertheless. In fact, they should be asked during the so-called training programs.

One problem with these programs is that because of time limitations, there is always pressure to learn as much as possible in as short a time as possible. This sets up the contradiction between what is being said about discovery approach to learning and how the training sessions are themselves being run. Often a lot of talk goes on about constructionism, discovery learning, but much of the underlying modeling turns out to be traditional telling. It takes a lot of time to let people figure things out for themselves and training programs which have a specific time frame are not the *best* places to do that. Unfortunately, they are sometimes the *only* places to start modeling this kind of environment.

Even more problematic is the fact that whatever teaching style gets modeled in the training program, gets taken back to the classroom, where, although the time pressure is not *so* great, still the system encourages quick-fix teaching rather than slow-cook learning. Since helping people learn by asking questions rather than giving answers is not natural to most teachers, it takes a lot of modeling and practice for them to work this into their own behavior. Teacher training programs often don't model this as they should.

These problems were raised to members of the MIT team and some of the TT's. They all agreed that it was indeed a big problem with training programs. I think that the *Genesis* project took this limitation into account and tried to lessen it as much as possible by stressing the importance of long-term support.

Marta and Nuria recognized this as a large problem, especially in the in-country programs, but they felt that the MIT staff really tried to model scaffolding questions, rather than quick answers. Marta described how Mitch would help her:

> I would call Mitch to come and help me, "Mitch, come here! I have a problem!" Mitch would ask me, "What do you want to do?" and I would say, "I want to make a square" or something. Then he would say, "Can you draw a square on a paper? Show me." and I'd do it. Then he'd point to a part of the drawing and ask me, "Do you know the commands to do this in Logo?" If I knew, I'd tell him. If I didn't, sometimes he would tell me, and sometimes he'd say I should wait a while, to just play with the ones I knew.
>
> This was this kind of "teaching" we got — someone would try to make us relate the knowledge we were seeking to something we already knew about, rather than answering right away.

Nuria added that at first she didn't have enough confidence in her own teaching so during the early training programs she tried to tell the teachers everything she knew. "I told too much; now, not so much!" she said with a laugh. "We're getting better. That's what learning is all about, isn't it?"

Notes

1. It should be noted that the Costa Rican people have an unusual involvement with education. All public education is free, but parents and schools have invested both time and money in preparing for the computerization of education by supplying furniture and air conditioning for the labs. "Parents were *so* pleased to hear the good news about their kids using computers," one of the teacher training team reported.

   Perhaps one reason for the support and excitement of so much of the adult community is that the second phase of this plan is to open the computer labs in the evenings and on the weekends to the community. The teacher training staff is now preparing programs to teach adults.
2. I learned of an only slightly comparable situation, but one which also puts great emphasis on the importance of teacher support. John Foster, the director of Microelectronics Educational Support Unit (MESU), the government educational agency responsible for advising teachers in the UK about how to make use of technology, announced in his keynote address of the World Conference on Education and Technology in Edinburgh (March, 1988) that the UK is planning to put 8.5 million pounds into creating a core of 628 "advisory teachers" who will work for two years with teachers in the schools to help them understand and utilize technology with a "particular phase of the curriculum" such as science or language arts. (This will hopefully not preclude cross curriculum work! -hw)

   This support contrasts greatly with the situation in many other countries where governmental support for teacher guidance is much less far sighted.
3. It is so important to make this distinction in our thinking, writing, andspeaking! Newspapers and journals are filled with reports and research which talk about things like how many computers are in the schools, and grants which have been given to purchase X dollars worth of equipment. In Japan, for example, almost daily headlines can be seen purporting to show the depth of governmental commitment to educational uses of computers by talking about the *numbers* of computers in schools and the size of the budgets, rather than *how* the computers are being and will be used and *how* the teachers are being prepared to use them.
4. What an ingenious stroke to insist that any schools choosing to be involved needed to send their principal! The importance of having the administration of a school closely involved with the educational innovations taking place in that school can be understood by anyone who has ever worked where such support was lacking.
5. It's important not to take for granted this seemingly minor official support in time and money. In Japan, for example, most teacher training workshops are on day affairs! In the U.S., I wonder how many teachers could get released time from their district in the middle of a school year in order to get training?
6. The average number of kids in each class is about 35. Since each lab has 20 computers this means that children work in pairs. They "always" work in pairs because the teachers feel that both students benefit more than if they work alone.