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

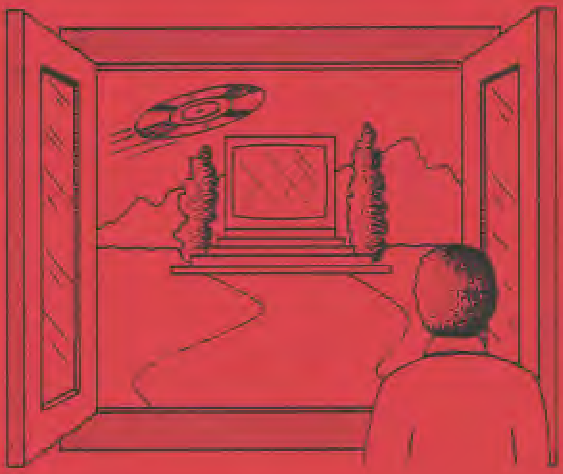
LOGO EXCHANGE

December/January 1990–1991

Volume 9 Number 4

International Society for Technology in Education

ISTE Publications



LOGO EXCHANGE

Volume 9 Number 4 Journal of the ISTE Special Interest Group for Logo-Using Educators December/January 1990–1991

Publisher
International Society for
Technology in Education
Dave Moursund, Executive Officer
Anita Best, Managing Editor
Talbot Bielefeldt, Associate Editor
Mark Horney, SIG Coordinator
Lynda Ferguson, Advertising Coordinator
Ian Byington & Tracye May, Production

Logo Exchange is the journal of the International Society for Technology in Education Special Interest Group for Logo-using Educators (SIGLogo), published monthly September through May by ISTE, University of Oregon, 1787 Agate Street, Eugene, OR 97403-1923, USA; 503/346-4414. This publication was produced using Aldus *PageMaker®*.

POSTMASTER: Send address changes to Logo Exchange, U of O, 1787 Agate St., Eugene, OR 97403-1923. Second-class postage paid at Eugene OR. USPS #000-554.

Contents

ISTE Membership

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

SIGLogo Membership (includes *The Logo Exchange*)

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

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

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

From the Editor

Why Logo?

In our graduate program here at the University of Oregon, I teach a required Logo course as part of the specialization in Computers in Education. Students from all over the world are enrolled in this program. This term my classes include students from Japan, Taiwan, Canada, Korea, the Netherlands, India, and, of course, the U.S. Most of the students begin the program in the fall, but some, of necessity, come at other times during the year. Since the Logo course is offered only in the Fall, both "old" and "new" students find themselves together in the same course. While Logo is not required in other courses of the program, many students (both old and new) use it as their language of choice in our Computer Science for Educators course that is taught in the spring or for developing projects in other courses.

Early in my fall course, I have students read a number of articles by and about Papert and about the Logo philosophy. Then we watch a videotape of the Nova episode from several years ago called "Talking Turtle." The videotape is filled with examples of uses of Logo with every imaginable type of child, shows Logo in use around the world, and is interspersed with observations by and discussions with Papert.

Every year I am struck again by the comments made by my students after doing their assigned readings and watching "Talking Turtle." The old students often remark that they thought that Logo was just another programming language and had no idea of the rich philosophical background associated with Logo. New students from the U.S. and Canada are usually somewhat familiar with Logo. In addition, they are usually aware of major educational movements in the United States such as cooperative learning and school reform. It doesn't take long until many of these students begin to make connections between the Logo philosophy and these current educational trends.

Yet other new students come from educational systems that are much more formal and rigid than most systems in the U. S. and Canada. They have a great deal of difficulty understanding the kind of open-ended projects that I assign. Because they have previously taken traditional courses in BASIC and Pascal, they are mystified by the lack of formal specification for computer programs They sometimes react very negatively to turtle graphics as not being serious. Early in the course they will often tell me that they want to learn to do "real": programming—not just this graphics stuff.

How exciting it is to see the changes that occur in all of my students. First they see the potential in Logo both as a programming language and as a learning environment. Gradually their views of education in general change. Just yesterday, a Taiwanese student rushed into my office to talk about a reaction paper he had written. He has a strong background in computer science and plans to get a doctorate in computer education at the University of Oregon. It is clear that he initially felt that Logo was a very interesting children's language...until he read Brian Harvey's books, volumes 2 and 3! (*Computer Science Logo Style*, MIT Press.) He said he read until 5:00 in the morning! He was so excited by the advanced concepts possible with Logo. He couldn't stop talking about data structures, finite automata, and artificial intelligence.

For me, seeing my students grow in understanding of Logo is new and exciting every fall. I am disturbed when they have learned Logo only as a programming language and am excited along with them as they begin to see the possibilities. I can't wait to read their journals each week. Logo seems to have become a delightful vehicle to open up new horizons for most of our graduate students. They begin to think about all aspects of education, both in their own classrooms and in their own countries. Once students understand Logo, it is easy for me to help them see the possibilities that technology offers beyond simple drill and practice. For me, my Logo course is the "foundations" course in Computer Education.

But what of other teachers learning about technology? In most universities, preservice teachers receive little, if any, exposure to computing. If they learn Logo, it is most likely that it will be a few weeks in a single course where they are taught many other topics, including programming in BASIC. It's such a shame that so few educators have a chance to "catch the dream" conveyed by Papert in "Talking Turtle." It seems likely that even if such teachers never used Logo with students, the mere exposure to the Logo philosophy and environment has the potential to deeply affect the way they think about education. Perhaps one of the most powerful ways to use Logo in education is in the training of teachers. Think of the impact on the classroom of tomorrow!

Sharon Yoder
ISTE
1787 Agate Street
Eugene, OR 97403
BITNET: Yoder@Oregon

Monthly Musing

Don't Close the Blinds
by Tom Lough

Have you ever read a story so poignant, so touching that you felt like crying out loud? I saw one recently that I'd like to tell you about.

In a guest editorial published in the March 1990 issue of *Microsoft Works in Education*, one of the ISTE newsletters, John Madian tells about an incident that happened to Paul Stachowski, a high school English teacher from New York, when in the third grade.

During class one day, while all his classmates were performing the same worksheet task in their ramrod-straight rows of desks, young Stachowski was gazing out the window, as third graders are wont to do on occasion. Suddenly, his attention was captured by an unusual event outside. Even though the sun was shining brightly, raindrops were falling from the sky! Amazing! Incredible! Such an impossible happening created a clear case of classic cognitive dissonance in this third-grade mind!

With no thought for the lesson plan or the day's announced instructional objectives, Paul Stachowski shot up his hand, imploring for the teacher's attention. "Teacher! Teacher!! Look outside! See? It's raining! And the sun is shining! How can that be?"

Talk about the opportunity for a teachable moment, this was it. But, alas, the teacher had quite a different agenda. Intent on keeping the students on task and under control, loath to depart from the lesson plan, perhaps fearful to encounter a situation which she knew she did not understand herself, she rushed over to the window and closed the blinds, cruelly crushing the enthusiastic curiosity of a (formerly) eager learner.

I don't know about you, but my heart went out to that young lad. Some incidents are so emotional that it is impossible to forget them; for Stachowski, the closing of the blinds was clearly in that category. I am glad that he eventually decided to share the details with others in the moving verse that was quoted in Madian's sensitive editorial.

Stachowski's initial excitement about his observation reminded me of some of the youngsters in my summer Logo workshops. They could hardly contain themselves as they discovered more and more interesting aspects of this amazing computer language.

"Wow! Look what happens when you put negative numbers with FORWARD!"

"That's nothing! I found out you can REPEAT a REPEAT!"

"Hey! Why is it doing THAT?"

Although my eyes were misty and my heart was aching for young Paul Stachowski as I put down the editorial, I felt confident that such an incident probably would not happen today in a Logo class. Logo and its accompanying philosophy of teaching makes us more sensitive to student observation and thinking, and encourages us not to be afraid to say, "I don't know. Why don't we try to find out together?"

Shouldn't we all be more willing to take advantage of the teachable moment? Let's not close the blinds on a single young mind—ever.

FD 100!

Tom Lough
Founding Editor
PO Box 394
Simsbury, CT 06070

About the Cover

This month's cover was created by Michael Swith. He developed his project in a computer literacy class taught by Helen Sarver. He used Apple Logo II to make his picture.

Helen Sarver
619 Milton Street
Alliance, OH 44601

Logo Ideas

Using LogoExpress
by Eadie Adamson

Last month when I discussed *LogoExpress* I deliberately skimmed over some of the fine points that make this new product really useful. This month it seems reasonable to focus on three areas which may be puzzling to new users: tool keys, creating and sending text files, and sending shapes pages. The documentation for *LogoExpress* will lead the more adventurous safely through mailing either kind of file. But for those who have difficulty reading between the lines or who have not played around with creating extra shapes pages or backup files, or who are uncomfortable with using tools, this column is for you!

Tools Keys

Last month I suggested that you might set up a special **mailer** page to connect with a particular host. The **setup** procedure on this page is configured to dial the correct number and give your user name and your password. I then suggested clearing the text from the front of the page and locking it. This makes it possible to simply choose that page and use it to connect to the host. However, this can cause you problems if you collect a large amount of mail. There are quite a few procedures already on the **mailer** page which take up memory. Remember that there is a limit to the amount of text on a page. There are ways around this, but why create the problem in the first place?

What Are Tools?

Tools are procedures which are loaded into the workspace but are invisible to the user. As far as the user is concerned, the tool procedures act just as primitives. Tools remain in the workspace as one changes from page to page. When the computer is turned off, the tools are no longer there and must be loaded again the next time they are to be used. Tools can be any page of procedures. Often pages to be used as tools have the word "tools" appended to the name of the page simply as a convenience. The tools feature is analogous to "burying" procedures in *Apple Logo II*. *LogoWriter* and *LogoExpress* both can use procedures as tools. Some of you may already have played around with tools with *LogoWriter*. The *LogoWriter* Primary disk has a **startup** page, for example, which loads some primary tools into the workspace. A *LogoExpress* mailer page can be loaded as tools using the tools keys.

Then ... What Are Tools Keys?

Tools keys are a new feature added to *LogoExpress*. To use them, simply select the appropriate **mailer** page on the Contents page and press the tools keys: Apple and "t" in the Apple version, Control and "t" in the IBM version. This will load the procedures on the selected page into memory so that you can use them. It will also give you a fresh unnamed page. On the right side of the bar separating the Command Center and the new page you will see the name of the tools page currently in memory. All the commands you need to dial that host are now available to you. You have the added advantage of a blank page to work with.

What About Text Files?

As with tools, text files can be used with both *LogoWriter* and *LogoExpress*. The command

```
savetext "name.of.file
```

will save text, *on the side of the page you are using*, as an ASCII file. While it will not show on the *LogoWriter* Contents page it *will* show at the bottom of the *LogoExpress* Contents page.with the word **text** beside it. (Type **show filelist** on a *LogoWriter* page and you will be able to see the text files identified with a ".txt" at the end.) To load a text file, you simply type

```
loadtext "name.of.file
```

but you don't need to do this to mail a text file!

Why Use Text Files?

There is a clear advantage in using text files for telecommunications. *LogoExpress* has the command **mail** which takes two inputs. The first input is the file to send, the second is the username to whom the file will be sent.

```
mail "textfile "to.whom
```

Assume you wish to send mail to several different people. You can load *LogoExpress* with the appropriate mailer tools and then write your letters one at a time, saving them as text. Then use ct to clear your page to write the next letter. Then you can log in and mail the series of pages. For example,

```
login
mail "letter1 "michaelt
mail "letter2 "brian
hangup
```

If you save a collection of letters and then forget what you named them, type

```
show filelist
```

You will see all the file names on your *LogoExpress* disk. You can also look at the Contents page. Below the list of pages, each of which has "page" on the right side of the column, you will find a list of all the text files with the word "text" on the right column.

Files from Other Word Processors

You can also send text files created with *LogoWriter* or files from other word processing programs which have been saved as text. *AppleWorks* has a special way of doing this via the print option. The public domain word processor *FredWriter* automatically saves files as ASCII files. Check your particular word processing program to determine if you can also save files as "text only." These files can be sent via *LogoExpress*.

Sending *LogoWriter* pages is done with the same **mail** command you use to send text files. The only difference is that the *input* for the page is the name of the *LogoWriter* page.

Then What's the Mail Page For?

Remember that *LogoExpress* saves your mail on a page called **mail** and at the same time puts text of your messages on the page you are using. The **mail** page is a temporary page. The next time you receive mail, the previous **mail** page is cleared and the new mail is saved on the **mail** page. Working with a new page simplifies the process of saving mail you want to keep.

If *LogoWriter* pages have been sent to you, the mail message will indicate "page" rather than "text" and you will need to check the Contents page for the *LogoWriter* page you received. Remember that you can look at the page with *LogoExpress*, but you can only make changes with *LogoWriter*. Apple pages will be visible only on Apple versions, and likewise for IBM, although with both you will be able to read the text which may include procedures.

Once your mail has been received or sent, you may name your page and save it. Sometimes you may simply wish to leave without saving it. The **leavepage** command works with both *LogoWriter* and *LogoExpress*, clearing the page and returning to Contents.

If you clear your page too soon, you can get your mail back with the command:

```
loadtext "mail
```

The **mail** page will remain unchanged until the next time you collect mail. If you **login** and receive no mail, the **mail** page will still contain the last mail you received.

Sending Shapes

What about the **shapes** page? Often *LogoWriter* projects use special shapes. Sending the page without the special shapes is rather useless, since one needs to have the shapes as well.

For quite some time I had been using *LogoExpress* quite easily to send text messages and *LogoWriter* pages that did not require special shapes. Recently, however, I needed to send the shapes page as well. Here's the sequence I followed to send a shapes page:

Load *LogoExpress*.
Load the appropriate mailer tools.
Put the *LogoWriter* disk in the drive instead of the *LogoExpress* disk.
Make a copy of the shapes page:

```
copyfile "shapes "project.shapes
```

The renamed shapes page will not show up on the directory. More on that later. Now I was ready to send the shapes page along with the project page. Here is the script to do that (substitute appropriate names for my general ones here):

```
login mail "project "michaelt
mail "project.shapes "michaelt
hangup
```

On The Receiving End...

The person who receives a **shapes** file has a little work to do. First, it won't show on the *LogoExpress* Contents page. If you are in doubt that it is there, go to any page and type

```
show filelist
```

In the list you should see a page with "shapes" at the end.

Getting the New Files into *LogoWriter*

Load *LogoWriter*. Then, assuming you are using two drives (see below for what you do if you have only a single drive), insert your *LogoExpress* disk with the files you just received in the second drive. Use the **copyfile** command to copy this file—in this example, **project.shapes**—to your *LogoWriter* disk. First, if you don't know the prefix of your *LogoWriter* disk, do this:

```
show online
```

You should see two lists, one for *LogoExpress* ("lex")and the other for your disk. Let's assume it is named "logowr." Now type this:

```
    copyfile  "/lex/project.shapes
    "/logowr/project.shapes
```

This puts a file on your *LogoWriter* disk with the shapes that were sent to you.

Also copy the file containing the project that goes with the "project.shapes" file from the *LogoExpress* disk to your *LogoWriter* disk (substitute your page name for "project.shapes" in the commands above and press return).

Since you cannot access two shapes pages at the same time, you can either elect to give up the shapes on your disk (the simplest option) or create a subdirectory first if you need your *LogoWriter* shapes page.

Let's assume you are using a fresh scrapbook disk. You must first erase the shapes file from your *LogoWriter* disk:

```
    erasefile  "shapes
```

and then rename the file sent to you as the shapes file:

```
    rename  "project.shapes  "shapes
```

Note: GS and MSDOS users need to add ".shp" at the end:

```
    rename  "project.shapes  "shapes.shp
```

Then you're ready to go. Remember to type **getshapes** in the Command Center before using the new page.

Or, Put the New Files in a Subdirectory

If you prefer, create a subdirectory:

```
    createdir  "project.pages
```

and then copy the newly received files from your *LogoExpress* disk into the subdirectory on your *LogoWriter* disk using the **copyfile** command. Since a shapes page in a newly created subdirectory does not exist until it is accessed, you do not need to erase the shapes page before copying the new one onto your disk. By the way, if you have only a single drive, you will need to transfer your files by using a copy program.

A Final Note

Some of the processes outlined above may seem quite complicated. I have found that it helps to keep notes on what I have done—then I can follow them the next time, step by step. Yes, I do keep my reference manuals at hand when I'm trying something new! The appendixes in the "Introducing *LogoExpress*" booklet are quite helpful. Eventually the processes become second nature, however, and the notes are no longer necessary. Remember also that when you are using *LogoExpress* your contact at the other end can keep you informed on how things are going at that end. You can compare results and discuss the process via *LogoExpress*.

Meanwhile, happy telecommunicating! I hope to hear from some of you via *LogoExpress*.

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

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

Questions Please!

by Frank Corely (and friends)

At last, I have some answers for you. During the past two weeks, I have received answers to most of the questions in the September and October issues of *Logo Exchange*. I will repeat the unanswered questions at the end of the column, to try and jog the memories of potential respondents. I will fit as many answers in this month's column as I can, and continue with answers next month.

Before I begin, though, I have a question to a large segment of the readers out there. **Where are you?** I would like to call the bluff of a lot of Logo neophytes and veterans. At NECC '90, there was broad agreement that the needs of the new users of Logo must be served, and from this agreement this column evolved. Since the day after the meeting of SIGLogo, the total number of questions I have received, either from the working group formed that day or from readers of this column is: zero. Pardon me if my tone is frustrated, but, as I said in the very first issue of this column, I will have nothing to write if no one sends me anything. In addition, I have received a large number of answers to questions, and for these I am very grateful. But my gratitude goes to only four individuals, for they have answered all of the questions. Logo-philes: Come out, come out wherever you are!!

Here are the answers, and some repeated questions. Please send any questions or answers to my address below.

Answers to September Questions

1. **The question of sharing the enthusiasm** is best answered by fellow teachers, but Terrapin has produced a nice little booklet called *Why Use Logo?* which answers this question very well—and is not restricted to particular versions of the language. Simply contact them for a copy.

2. **SETH and SETPOS/SETXY**. Even some first graders can use these to great advantage. When teaching **SETH,** use a "turtle compass" made of an index card or plastic transparency with headings marked on it. Teach **SETPOS/SETXY** when teaching coordinates, target games or, rectangular versus polar coordinates.

   A very elegant answer that came from two different sources: teach these topics when the students need them.

3. **Making a slide show**. The Terrapin people inform me that Dorothy Fitch has an upcoming *LX* column on this very topic.

4. **Bottom-up vs. Top-down**. Here is an answer used in third and fourth grade classes with some success:

   Draw a picture, animation, or whatever on paper first, away from the computer, and indicate any action that should occur. (The idea of planning a program away from the computer is a good one in any case, as it forces the programmer to design rather than hack.) List the procedures needed in order as much as possible. This is done away from the computer and tends to help the children visualize their programming in a top-down way.

   This answers the question for early programmers, but I would be interested in seeing someone respond to this issue for high school or college students, students preparing for an Advanced Placement course in programming or an equivalent college course. I believe that this was the intent of the original questioner.

5. **Checking Input, Running a Procedure.** Suppose we want the user to type an answer to a question, and the correct answer to the question is **TESTWORD.** Suppose that the procedure to run if the answer is correct is **RIGHT.PROC** and the procedure to run if the answer is incorrect is **WRONG.PROC.** These lines of code will preform that task in *LogoWriter*.

```
PRINT [What is the answer to this
   question?]
MAKE "ANSWER (FIRST READLIST)
IF (EQUAL? :ANSWER "TESTWORD)
   [RIGHT.PROC] [WRONG.PROC]
```

   or

```
PRINT [What is the answer to this
   question?]
IF (EQUAL? (FIRST READLIST)
   "TESTWORD) [RIGHT.PROC]
   [WRONG.PROC]
```

6. **RIGHT vs. FORWARD.** I think that the answer to the question concerning **SETH** and **SETPOS/**

Questions Please! - continued

SETXY is perfectly appropriate, and it came to me from Eadie Adamson: teach this when they need it. In all seriousness, without dodging the question at all, this is a perfectly appropriate answer and fully consistent with the Logo educational philosophy. Let the students experiment with these commands. Allow them the opportunity to guess, and guess incorrectly. Use trial and error. My students usually think that RT moves them to the right. They nearly always try it. It does not take them long to figure this difference out. The effective way to teach this topic is to let the students teach themselves.

I would like to repeat two questions from the October issue that were asked of me with some urgency, and I would like to see responses as soon as possible. I have gotten a fair number of answers to questions concerning Terrapin Logo, but the *LogoWriter* experts have been slow to respond. Please, help us out here.

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

4. I am using Terrapin *Logo PLUS*, and I have a data disk on which my files are stored. When the data disk gets full or nearly full, and I try to read a file from it, I get a disk error. Is this a ProDOS error or a Logo error? How do I make a full disk work?

My thanks to the answerers: Jan Bird, David McClees, Dorothy Fitch, and Brian Harvey. To the rest of you: Is this how Abigail van Buren got her start? I hope to hear from all of you turtle lonely hearts soon. Until then, I hope that all of your questions are answered. If they were interesting questions, send them to me anyway, along with the answers! See you next month.

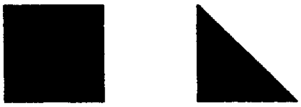
Frank J. Corley
St. Louis Priory School
500 South Mason Road
St. Louis MO 63141

Beginner's Corner

More Quilts!
by Dorothy Fitch

Last month, we explored a traditional quilt project for your class. This month, we take a look at quilts that are created by rotating simple figures. These procedures were inspired by the September 1990 *Arithmetic Teacher* article "Symmetry in American Folk Art."

An amazing number of interesting and dramatic quilts can be created using just two basic designs: a square and a half-square that is filled on one side of its diagonal. The half-square can be rotated in 90° turns for a total of four different orientations.



To explore these quilt designs, first load Logo, then enter the following procedures:

The BORDER procedure erases the screen and draws the border of your quilt.

```
TO BORDER
DRAW         ; your version may use CLEARSCREEN
               or CLEARGRAPHICS
PENUP
SETXY -60 60      ; your version may use SETPOS
                    [-120 120]
PENDOWN
SETXY 60 60       ; see SETPOS comment above
                    for the following lines
SETXY 60 (-60)
SETXY -60 (-60)
SETXY -60 60
SETXY -60 30
END
```

SQ draws a filled square. (If your version of Logo does not include a FILL command, see the alternate procedures given later in this column.)

```
TO SQ
HIDETURTLE
REPEAT 4 [FORWARD 30 RIGHT 90]
PENUP
RIGHT 45
FORWARD 5
PENDOWN
FILL
PENUP
BACK 5
LEFT 45
PENDOWN
MOVE
SHOWTURTLE
END
```

R draws half a square, with the area on one side of its diagonal filled. The input number determines the number of 90° rotations. Compare R 0, R 1, R 2, and R 3.

```
TO R :NUMBER
PENUP
REPEAT :NUMBER [FORWARD 30 RIGHT 90]
PENDOWN
HALFSQ
PENUP
REPEAT 4 - :NUMBER [FORWARD 30
   RIGHT 90]
MOVE
END
```

MOVE simply moves the turtle to the next square position to the right. Use MOVE when you want an unfilled square in your design.

```
TO MOVE
PENUP
RIGHT 90
FORWARD 30
LEFT 90
PENDOWN
END
```

NEXT moves the turtle down to the beginning of the next row of squares.

```
TO NEXT
PENUP
SETXY -60 YCOR - 30   ; or use SETPOS
                        LIST -60 YCOR -
                        30
PENDOWN
END
```

The rest of these procedures are used by those above.

Beginner's Corner - continued

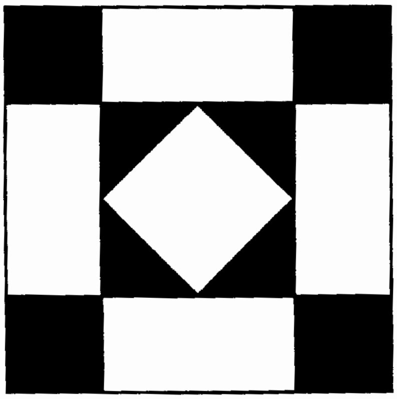
```
TO HALFSQ
HIDETURTLE
FORWARD 30
RIGHT 135
FORWARD 42.4264
RIGHT 135
FORWARD 30
RIGHT 90
FILL.TRI
SHOWTURTLE
END

TO FILL.TRI
PENUP
RIGHT 45
FORWARD 5
PENDOWN
FILL
PENUP
BACK 5
LEFT 45
PENDOWN
END
```

If your version of Logo doesn't have a FILL command, use the following procedures for SQ and HALFSQ instead of those listed above and add the TRI procedure:

```
TO SQ
REPEAT 15 [FORWARD 30 RIGHT 90
   FORWARD 1 RIGHT 90 FORWARD 30 LEFT
   90 FORWARD 1 LEFT 90]
END

TO HALFSQ
HIDETURTLE
TRI 30
PENUP
LEFT 90
FORWARD 30
RIGHT 90
PENDOWN
SHOWTURTLE
END

TO TRI :NUMBER
IF :NUMBER = 0 THEN STOP
FORWARD :NUMBER
BACK :NUMBER
RIGHT 90
FORWARD 1
LEFT 90
TRI :NUMBER - 1
END
```

Test your procedures by typing the following commands below. Each line will complete a row of four boxes on the screen and move the turtle to the beginning of the next row. When you are done, you should see the design below. Note that the colors in this design are the opposite of what you will see on the computer screen. The black areas here are filled in; the white areas are left empty. The colors on your screen will be reversed. You may want to experiment with creating both "positive" and "negative" images.

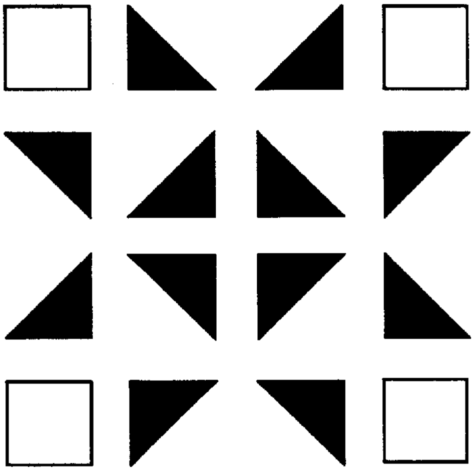
```
BORDER

SQ
MOVE
MOVE
SQ
NEXT

MOVE
R 1
R 2
MOVE
NEXT

MOVE
R 0
R 3
MOVE
NEXT

SQ
MOVE
MOVE
SQ
```



Each of the designs on the page at the end of this article is made up of 16 separate squares and half-squares. To recreate a design, envision it in exploded form, as in the picture below. Seeing the individual components will help you figure out how to draw it. Practice using R 0, R 1, R 2, and R 3 so that you know which orientation each command will produce. It may help to draw and label each figure on a piece of paper as a quick reference guide.

For example, here is the set of commands you would type to create this design.

```
BORDER

MOVE
R 0
R 3
MOVE
NEXT

R 2
R 3
R 0
R 1
NEXT

R 3
R 2
R 1
R 0
NEXT

MOVE
R 1
R 2
MOVE
```



Use the square and half-square building block procedures to copy the sample designs on the following page and create original quilt designs. You may find it easier to design quilts off the computer using 2-inch squares. Each student can make 16 half-squares, colored on one side of the diagonal, and at least 4 squares that are colored on one side and blank on the other. By rotating the half-squares and using the full squares, they can create all the sample designs and those of their own invention.

As part of your math lesson, ask your students these questions about their quilt designs:

- Does your quilt have a fold line or axis of symmetry? Can you fold your quilt in half so that one side of the design matches the other?
- Does your quilt have rotational symmetry? In how many positions can you turn the quilt and have it look the same as the original?

. Here's a game to play using paper or computer quilt squares. Have one student create a quilt design behind a divider or where the second student can't see it. The first student then describes the quilt to the student on the other side of the divider, who has a set of paper quilt squares or a computer. The second student follows the instructions and descriptions of the first student and tries to recreate the quilt design. This activity encourages the exchange of information that is complete and accurate, spatial awareness involving rotation and symmetry, and attention to detail.

Happy quilting!

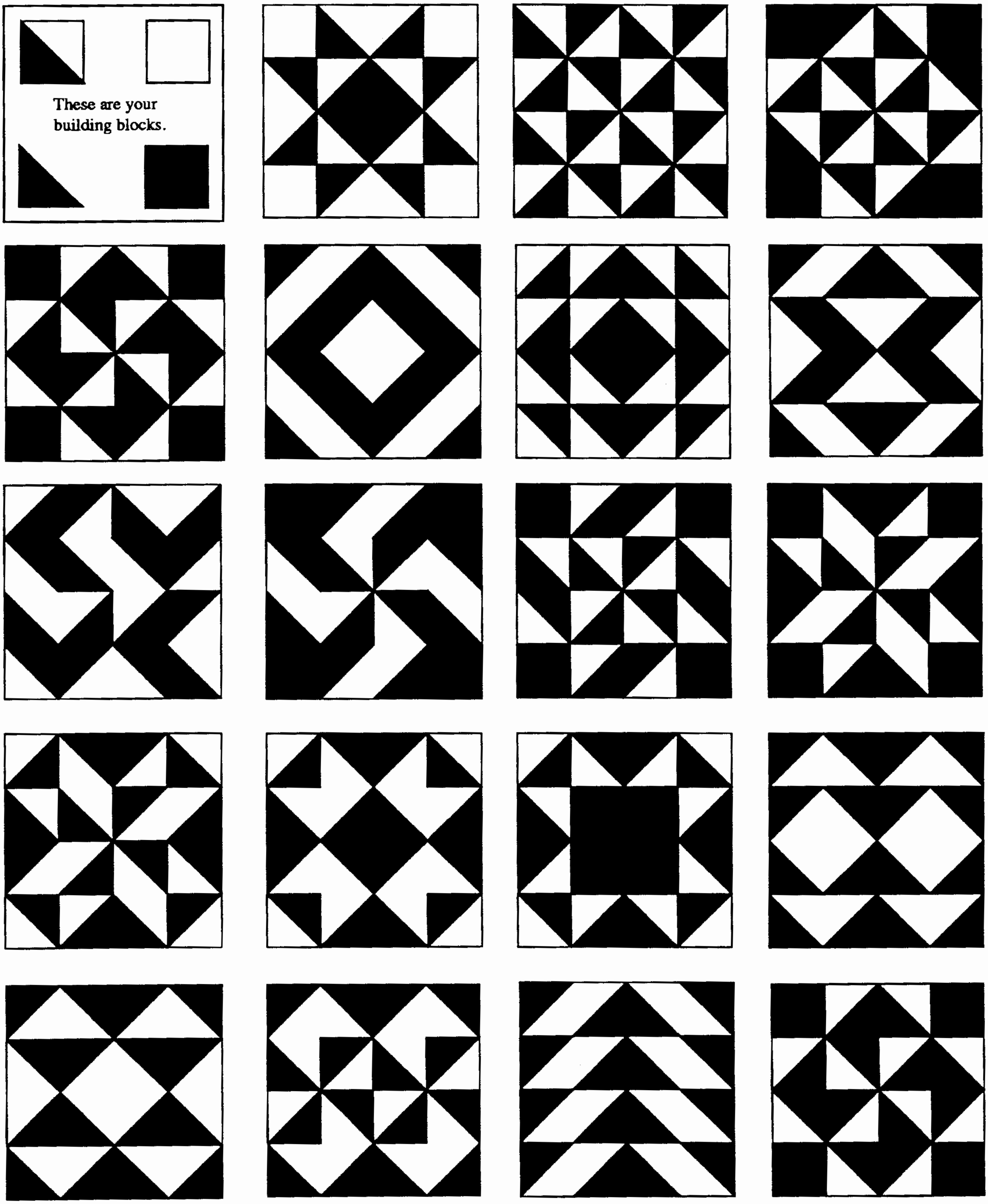
References

Zaslavsky, C. (1990). Symmetry in American folk art. *Arithemetic Teacher*, *38*(1), 6-12.

Dorothy Fitch has been the Director of Product Development at Terrapin since 1987. She first became involved in educational technology in 1981 when the school where she taught music received its first computer. Since that time, she has consulted with schools to provide inservice training, curriculum development and software customization; taught a number of college courses; and directed a computer classroom for teachers and students. She has also coauthored Kinderlogo, a single keystroke Logo curriculum for young learners, and created the Logo Data Toolkit. Through her work at Terrapin, she has presented workshops at many local, regional and national conferences, edited several of Terrapin's curriculum materials, and coordinated the development of Logo PLUS. She can be contacted at:

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

Beginner's Corner - continued



Easy Map Drawing with LogoWriter

by Francisco Quesada

One of the things that has impressed me most about LogoWriter is the great amplitude and versatility it offers the individual as a working environment. You can pass from doing graphics to making computations, from list processing to writing text, without ever having the sensation that you are exiting an option and entering a new one. You are always *there*, in the same page, no matter if it means the Front of the page, the Flip side or the Command Center.

Particularly remarkable is how narrow the gap has become between the *editing* mode and the *executing* mode of a procedure. For a person who started in the world of computers through LogoWriter, which is the case of many teachers in my country Costa Rica (see below), the existence of two such modes is sometimes even ignored. (I have witnessed this fact often) .

Persons having some familiarity with other computer languages (especially those which require compilation) or with former Logo versions, might well appreciate the high degree of uniformity among the different options within LogoWriter. Perhaps we owe this comfortable sensation of unity to the great idea of working inside a page.

However, this sensation of unity is really more than just that: It is a real issue. Let us consider as an example the possibility of including the operation in your procedures, that consists of asking if you are in the Front of the page or on the Flip side, without the necessity of any sophisticated commands. All you have to do is to appeal to the primitive FRONT?.

I would like to illustrate with an example of the great versatility of LogoWriter, at least as I perceive it. I want to present it as an example of a procedure that can become a practical tool. I also want to present this particular example as an idea or production that reveals the many possibilities of the LogoWriter environment. I will also include a pair of suggestions about how this particular tool could be used in the classroom.

Creating the Map Tool

We know that making irregular figures, like a map of a geographical region, a human portrait, or an animal shape, and carrying it to the flip side in the form of a procedure might be something that can take considerable time and effort to draw. A great deal of trial and error seems to be the basic material involved in performing such a task using a computer.

The procedure I present here, which I have named MAP, is a short procedure that will greatly reduce the energy required in this line of work.

What Does Procedure MAP do?

If you are interested in drawing some very irregular figures, like the map of the American continent, you could proceed by carefully trying to move the turtle small distances. Maps have borders that often have no straight edges. These borders can be approximated very successfully by a sequence of instructions, where the turtle walks a few steps, and then turns a particular angle, and so on.

We certainly know that there exists a given sequence of instructions that will produce the map. What we are looking for is an easy way to create that sequence of instructions so we can carry them to the flip side of the page and put them into a procedure. This will also allow us to use the map in various projects or as a component of a major project.

One of the main obstacles faced when working in such a project is having to erase a wrong instruction. Even if you succeed in erasing the wrong piece of line properly, the wrong instruction that produced it, and those that erase the wrong piece, accumulate in the Command Center, confounding the list of instructions that you wanted to bring to the Flip side. The very nature of such experimentation leads inevitably to this type of problem.

How MAP Makes Things Easier

As a result of the following three factors, the procedure MAP increases the efficiency of drawing a map:

1. It decreases the chance of making mistakes (drawing wrong lines).

2. It automatically collects the list of instructions that reproduces the line drawn by the turtle and automatically carries it to the Flip side of the page.

3. It also cleans the list of undesired instructions derived from the erasing processes, and this is done automatically when wrong lines are erased.

Before the procedure MAP is called, you should bring the turtle to the position on the screen where the map should start and head the turtle in the right direction. The procedure MAP collects various smaller lists of 25 instructions each, rather than collecting at once the entire list that reproduces the whole map. Each time the procedure MAP has accumulated a set of 25 instructions (a clean piece of the final list), it automatically turns the page over, writes these instructions on the flip side, Easy Map Drawing with LogoWriter - continued

returns to the front of the page and exits the procedure. In order to continue working, you have to call the procedure MAP again.

Once the drawing or the map is finished, all the instructions that neatly reproduce it are on the flip side, arranged in sets of 25 instructions. All that is needed then is to label them with a procedure name.

Moving the Turtle in the Procedure MAP

The procedure MAP allows you to move the turtle by small amounts of 5 steps: **forward 5** is the unit of motion. The unit of turn, on the other hand, is given by either **right 15** or **left 15.** (*) When you call the procedure MAP, the turtle disappears from the screen. As long as you are inside the procedure, you can control the turtle's three possible basic movements (**forward 5, right 15,** or **left 15**) by pressing one the the keys: F, R, or L (do not press Enter or Return). If the last **forward** 5 instruction has produced an undesired effect, you can delete it by pressing the D key.

Two additional rotations have been added in order to facilitate the task: riGht and lefT 90-degree turns can be performed by pressing the keys G and T respectively.

The possibility of moving the turtle backwards has been deliberately eliminated. It can be the cause of confusion and can lead to some difficult situations that are better avoided (for example, having to drive the turtle exactly over a line previously drawn).

Commands to Use Within the Procedure MAP

Pressing F makes the turtle move 5 steps forward.
Pressing R makes the turtle turn 15 degrees to the right.
Pressing L makes the turtle turn 15 degrees to the left.
Pressing G makes the turtle turn 90 degrees to the right.
Pressing T makes the turtle turn 90 degrees to the left.
Pressing D erases the effect of the last F (forward 5) command, both from screen and from the list of instructions which has been collected.

There is no need to press the ENTER key after you have pressed any of the keys F, R, L, G, T or D.

The Procedure MAP

```
to map
name pos "startingpos
name heading "starthd
ht
pd
startdrawing [] 0
end

to startdrawing  :list :counter
if :counter = 25 [copylist stop]
name readchar "x
if :x = "f  [forward 5  (name sentence :list
    [forward 5] "newlist)]
if :x = "r  [right 15  (name sentence :list [right
    15] "newlist)]
if :x = "l  [left 15  (name setnence :list [left
    15] "newlist)]
if :x = "g  [right 90  (name sentence :list [rt
    90] "newlist)]
if :x = "t  [left 90  (name sentence :list [lt 90]
    "newlist)]
if :x = "d  [if (last butlast :newlist) = "fd [pe
    back 5 pd  name butlast butlast :newlist
    "newlist]]
start drawing :newlist :counter + 1]
end

to copylist
flip
bottom
print []
print [pu]
(print [setpos] sentence :startingpos)
(print [seth] :starthd)
print [pd]
print.list :newlist
flip
end

to print.list :commands
if empty? :commands [stop]
print first :commands
print.list butfirst :commands
end
```

What to Do If You Are Interested in Less Than 25 Instructions

It can happen that a small piece of a map, which needs to be drawn separately, takes less than 25 instructions to be performed. In such a case, there are two possible easy solutions in order to have the corresponding list of instructions carried to theFlip side.

1. Once the piece of line is finished, keep pressing a neutral key (Y or U for instance) as many times as necessary, until the automatic flipping of the page occurs.

2. Stop the procedure MAP in the ordinary way, pressing CTRL-A, then write COPYLIST (in the Command Center) and press ENTER. This will run the subprocedure named COPYLIST which accounts for carrying the accumulated instructions to the flip side.

Using the Procedure MAP As A Tool in the Classroom

The idea of creating interesting and accurate maps or drawings can increase the motivation of a child in developing a particular project. Studying the provinces or states of a country usually means collecting information about its economic, social, and geographical aspects, among other things. If children have a tool that permits them to draw maps accurately and without an unreasonable amount of effort, this will help them (and the teacher) create more and better projects in geography or social studies.

If provinces or states are distributed among students or groups of students, a class activity can be undertaken by integrating the pages created by each group into a whole-class geography project. Another idea is to let students take the same map as the basis for further work. Different groups might add to a map using more graphics and text showing different aspects of the selected region—important rivers or roads, main agricultural or industrial products, and so on.

One efficient way of making use of the procedure MAP is to carefully copy the selected map or drawing on tracing paper, fix this copy in front of the screen, and then follow the lines with the aid of the procedure MAP.

Nicaragua
Caribbean Sea
Costa Rica
Panama
Pacific Ocean

If more accuracy is desired when drawing a map, this can be achieved by changing the right or left turn of 15 degrees to a smaller angle (for instance 10 degrees). All that is needed is to change every occurrence of the number 15 by the number 10 in the procedure. Making the step (**forward** 5) smaller will also produce a gain in accuracy. The counterpart of a decrease in either of these quantities would be an increase (usually unnecessary) in the number of instructions of the final procedure. Taking 5 for the the step and 15 for the turn will give a very good result in most cases.

The procedure MAP presented here was written in IBM LogoWriter, version 2.0 by LSCI, 1988. A similar Spanish version is being used in the Costa Rican Logo Project.

A Brief Description of the Costa Rican Project

Costa Rica, a country with a population of 2.5 million people, has decided to bring computers into elementary public schools through a broad program based primarily on LogoWriter.

A school engaged in the program is supplied with a computer laboratory. Each laboratory consists of 19 IBM PS/2 25 computers arranged in a network, one IBM PS/2 50 acting as the server of the network, and one printer.

Sixty laboratories of the type described above were installed during the year 1988 throughout the nation, with priority given to schools in marginal urban zones and rural areas.

By the end of 1990, the total number of schools having a computer laboratory will be 210. By that time, it is expected that nearly 50% of the national public school population will be using computers and LogoWriter. The principal and three teachers in each school with a laboratory are trained in Logo and its use in connection with the curriculum. These three teachers become the instructors of the laboratory. When a particular group comes to the laboratory, one of these instructors will assist the class teacher, introducing new Logo primitives to the class or helping to find projects for the students. The instructors will continue to work with their groups as regular teachers. They will receive a stipend for their work as laboratory instructors.

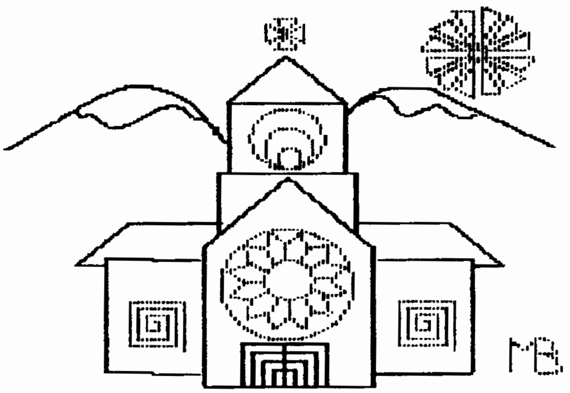
A group of tutors has been specially trained in Logo. The first group of tutors began their training at MIT under the direct guidance of Seymour Papert and some of the members of his team. These tutors are responsible for teacher training in Logo. The tutors also visit the schools regularly, being a source of pedagogical and technical support for the instructors in their daily work with the children.

Francisco Quesada has been a full time professor in the mathematics department of the Universidad de Costa Rica in San Jose, Costa Rica. Since 1988 ge has been working with the Costa Rican Logo Project as part of the collaboration of the university with the project. He is particularly interested in the teaching of science and technology.

Logo LinX

Luminescent Explorations

by Judi Harris



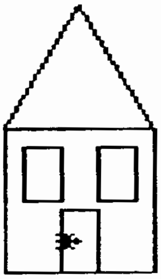
Programmed in Logo by: Marta Brann, Beaver College, Glenside, PA, February 1984.

> *Light, God's eldest daughter, is a principal beauty*
> *in a building.*
> *Thomas Fuller, 1642*

As the winter holidays approach in many parts of the world, buildings, trees, windowsills, mantlepieces, sidewalks, and street curbs become luminiferous. This, of course, is not a naturally-occurring phenomenon. Neighbors align equidistant paper bag luminaries along streets and walks. Hanukkah menorahs are kindled with candles for eight nights. Gardners trace the branches of their favorite trees with strings of lights. And, of course, Christmas celebrants often line their roofs, windows, and doors with multicolored electric bulbs. Inherent in these luminous decorations are motivating mathematical lessons on perimeter and area.

Logo Lighthouses

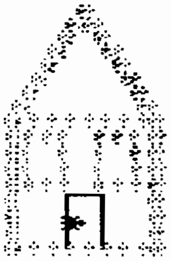
I'd be willing to wager that by this time in the school year, many of your students have drawn, filed, and forgotten houses similar to this one.



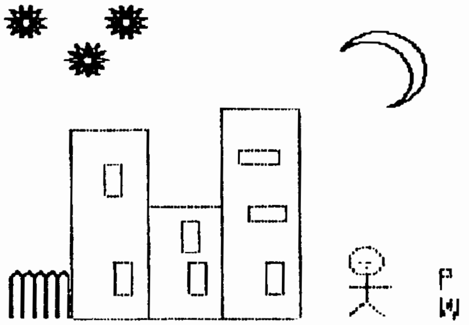
Why not challenge them to resurrect these creations, and decorate them with holiday lights? Once the pictures have been loaded from the diskette, the LogoWriter Turtle Move keys can be used to position the turtle for holiday decorating. Then a simple shape can be created to represent a luminous bulb, which, when stamped at measured intervals, will adorn your students' Logo houses for the holidays.

Igniting Instruction

Stated as such, Logo holiday lighting is an interesting beginning-level programming challenge. Tool procedures, such as the ones below (written in LogoWriter Version 2.0), can be used to help children also explore estimating perimeter and area size. Let's suppose that a student wanted to place lights around the outer perimeter of this apartment building.



Programmed in Logo by: Pat Wagner, Beaver College, Glenside, PA, February 1984.



An ESTIMATE tool procedure could be evoked, asking the student to make an estimate of the total number of steps that the turtle will have to travel to place Logo lights along the perimeter of the building. First, though, all global variable values should be reset with the RESET tool:

```
TO RESET
MAKE "PERIMETER 0
MAKE "AREA 0
MAKE "ESTIMATE 9999
END
```

Then the ESTIMATE tool would ask for the student's idea of the size of the building's perimeter.

```
TO ESTIMATE
CC
TYPE [How many turtle steps of lights do you
    want to use?]
NEXT.LINE
MAKE "LIGHT.ESTIMATE FIRST READLISTCC
END
```

Lightfooted Turtle's Feet of Lights

The student could then use the LIT.FD tool (in place of FORWARD) to place lights on the building. LIT.FD and its turtle-step input can be used by themselves, or from within a REPEAT statement, just like the FD command.

```
TO LIT.FD :DISTANCE
REPEAT :DISTANCE/10 [LIGHT.STEP]
END
```

Example: LIT.FD 50

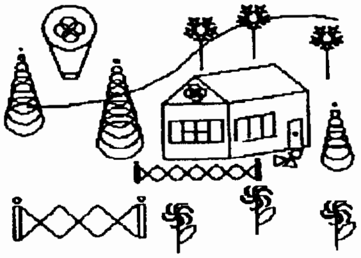
One of LIT.FD's subprocedures, LIGHT.STEP, moves the turtle forward 10 steps and stamps a light. As it places the lights, it adds 10 turtle steps for each to the growing perimeter of the shape that is forming. It also checks to see if the number of turtle steps already travelled has exceeded the student's estimate.

```
TO LIGHT.STEP
IF :LIGHT.ESTIMATE < :PERIMETER [CC TYPE
    [OOPS! You ran out of lights!] NEXT.LINE
    STOPALL]
PLACE.LIGHT
HT
SETC 0
FORWARD 10
MAKE "PERIMETER :PERIMETER + 10
PLACE.LIGHT
END

TO NEXT.LINE
TYPE CHAR 13
END

TO PLACE.LIGHT
ST
PD
SETSH 13
SETC (1 + RANDOM 5)
STAMP
END
```

How many turtle steps did *you* estimate were needed to traverse the perimeter of this building? The windows?

Enlightened Areas

Areas of polygons can be explored in terms of grids of holiday lights, placed, for example, on the front wall of this mountainside home.



Programmed in Logo by: Rebecca Sickert, Beaver College, Glenside, PA, February 1984.

The AREA tool might be helpful in this endeavor. It uses LIT.FD to traverse the wall's horizontals, repeating its trip according to the height of the area to be illuminated.

```
TO AREA :HORIZONTAL :VERTICAL
SETH 90
REPEAT :VERTICAL/10 [PD LIT.FD :HORIZONTAL
    PU BACK :HORIZONTAL SETY YCOR + 10]
MAKE "AREA :HORIZONTAL * :VERTICAL
END
```

Since the AREA procedure keeps track of the total area that has been decorated with it, the student can ask for that information at any time by typing TYPE :AREA. To use this tool for practice in estimating area, simply type RESET, then ESTIMATE before invoking AREA.

Holiday Heartlights

The legend of Hanukkah depicts light as a symbol of faith, hope, and survival. In the time of Judah Maccabee, the people's temple was destroyed. The Eternal Light, which must burn on the altar of the temple without ever being extinguished, was left with enough oil for only one day of flame. Yet, it burned for eight days while messengers travelled to get more oil. It was said to be a miracle.

The story of Christmas uses light to express promise and salvation. It includes a tale of a most brilliant star, which showed the way to the manger where a baby was born. He was said to be the true light of the world, a promise of salvation, a miracle incarnate.

May 1991 be a miraculous year of light for all of God's children, on Lightship Earth and beyond.

Judi Harris works in the Department of Teacher Education at the University of Nebraska, Omaha as an assistant professor of educational technology.

BITNET: JHarris@UNOMA1
Internet: JHarris@Zeus.unomaha.edu
CompuServe: 75116,1207

MathWorlds

Sunrise in Australia!

edited by A. J. (Sandy) Dawson

Yes, as in other parts of the world, the sun rises in Australia, but the behavior of the sun in the sky is not the Sunrise in Australia I wish to report about here this month. Rather, it is the Sunrise Project, a project of the Australian Council for Educational Research (ACER) under the direction of Liddy Nevile. This project has been under development since 1986, and today there are a number of Sunrise Classes scattered around Australia.

A review of the early working documents of the Sunrise Project gives one a flavor of the concerns that led to its creation:

"There is not an educational institution [in Australia] which considers the use of technology from the students' point of view..." (3, p. 2) And "This project is concerned to establish the ways in which information technologies can enhance the learning environments of secondary students..." (3, p. 2)

The project goals, directions, and operational rationale became clearer over the years. The concerns of the project seem to focus in three areas—the model of computing to be promoted in the schools, the role of the teacher working within that model, and the nature of the students' involvement in their own learning.

The Sunrise Model of Computing:

> There is a widespread model of computing which exists in schools in which the technology is used to deliver information, to facilitate certain learning processes by supplementing the learners' skills or to provide motivation and interest for students. In the Sunrise Centers, the technology is to play a different role...the technology is required to present a different medium for expression of present and existing information. (3, p. 3)
>
> In the Sunrise Centers the learning environment will be designed to enhance the opportunities for construction of knowledge by using the technology as a medium for that construction. Students will use computational representations of their naïve knowledge as the base for their developmental work. By becoming familiar with their own understandings from an objective position (seeing it represented in the computational medium), they will be in the position to compare their view of knowledge with the views of others (also represented in the computational medium). This activity has been available in many classrooms in recent years in some forms. (3, p. 3)
>
> In the Sunrise Centers, the technology will be used to go further and to help the students to "test" the competing sets of knowledge...The role of the technology is to provide the medium for this process and thus support students' development of the critical awareness of their knowledge. (3, p. 3)

This leads to the Sunrise Project's view of nature of the students involvement with their own learning:

> There is an assumption that once students have come to understand that their own perceptions are not likely to be unique, or necessarily correct, and that by allowing interaction between their understandings and those of others, they can construct from their personal knowledge a more objective form of knowledge, they will then be suitably skilled in knowledge construction to be able and willing to consider the more widely accepted community knowledge. (3, p. 3)

And this in turn gives rise to what the role of the Sunrise teacher might be:

> The role of the teacher in the development of an environment in which the process described above is possible cannot be underestimated. The teacher's skills extend beyond the transmission of knowledge and skills to the ability to establish a place in which students can take risks and are willing to do so because they consider it a worthwhile exercise...The teacher must act wisely and know the difference between interfering with and prompting the students' construction of the desirable knowledge. (2, p. 3/4)

Behind the Sunrise Project lies the assumption that schools should be places where students develop as independent learners. Indeed, one Sunrise School, the Methodist Ladies' College, Kew, states that:

> Independent learning is our goal, and computers form a significant part of the environment in which that independent learning occurs.

> Our intention in the Year 7 Sunrise Class is to extend the use of the computer further; to integrate its use more fully into the day to day activities of students; and for it to be part of a freeing of students and teachers from "lock-step" approach to teaching and learning which tends to predominate in secondary schools. We are aware of how effective many primary classrooms are in promoting independent learning, in providing for individual differences and in using the computer effectively in that process. In secondary schools this has been more difficult. (1, p. 6)

How did this school deal with some of these difficulties?

> The timetable was organized to provide the students with extensive blocks of time, and the curriculum has been developed to provide a framework within which independent learning is encouraged, but which also ensures a range of core experiences common to all our Year 7 students. The teachers in the team have training in teaching Mathematics, English, Geography, History, and Biblical Studies, and it was these subjects which provided the framework for the Sunrise program. (1, p. 7)

And what was this curriculum framework?

> In constructing a curriculum framework for our Sunrise class we looked to both our own curriculum and to the guidelines for a Sunrise School as developed by ACER....[which] offer three principles:
>
> Context
> - cultural
> - social and economic
> - school
> - gender
>
> Connection
> - from one "area of knowledge" to another
> - from one experience to another
> - from one starting point of knowledge, interest or curiosity to the unknown
>
> Construction
> - active response
> - creative response
> - tangible outcomes
>
> The starting point should be the student's interest world. Each individual or group begins with an object, a question, a curiosity, and moves into further exploration and creation.
>
> The guidelines also encourage students to consider any item or issue from four viewpoints:
>
> The Naturalist's Viewpoint
> - emphasizes observation of the natural world, directly or indirectly
>
> The Humanist's Viewpoint
> - emphasizes observation of human behavior and communication, values and ethics
>
> The Engineer's Viewpoint
> - emphasizes the construction of models and simulation
>
> The Artist's Viewpoint
> - emphasizes artistic interpretation and representation in art, literature, music and drama. (1, p. 7)

A truly ambitious curriculum, would you not agree? And what do the teachers at Methodist Ladies' College feel about the effect this curriculum has had on their students?

> It is our belief that these students are making significant progress as responsible, independent learners who are comfortable with computers as tools and aids to their learning, and as a medium in which they can create and realize ideas and experiences of their own. (1, p. 9)

The Sunrise Project and MathWorlds: What is the relationship?

"How does all this relate to Logo and mathematics?" you might be asking by this time, dear reader. Well, you see, *LogoWriter* was used extensively in this particular Sunrise class, and robotic qualities of *LEGO TC logo* are to be used in the current academic year. Other Sunrise Projects are investigating the use of computers and Logo in particular with preservice primary school teachers. (2, p. 10/11) The prototypic computer based medium known as *Boxer*, developed by Andy diSessa at UC Berkeley, is a computational system that grew originally from Lisp and Logo environments. The Sunrise Project is involved with exploring how this thinking tool might be used in Sunrise schools.

The Sunrise Project and its many classrooms certainly seem to be actively exploring the use of computers as thinking tools in secondary school classrooms, and the place of Logo and Logo-like computer environments and approaches is central to the Sunrise approach.

The LME Conference

In closing this column, I would like to mention the LME 5 Conference, which is to be held in Australia the first week of April, 1991. The LME group arose from a conversation among Celia Hoyles, Richard Noss, Uri Leron, and Joel Hillel as they sat on the lawn at MIT during a break between sessions at one of the earlier Logo conferences in the mid-80s. They decided to form a group whose focus was on the use of Logo in the teaching and learning of mathematics. The first and second LME conferences were held in London, England, 1985 and 1986; LME 3 was in Montreal in 1987; and LME 4 in Jerusalem in 1989. If all goes according to plans, LME 6 will be held in Vancouver, in July 1992. Before that, at LME 5, Liddy Nevile, Director of the Sunrise Project, will host all those interested in how computers can enhance the teaching and learning of mathematics at a site in Northern Australia beside the Great Barrier Reef. Might some *LX* readers join us there? (See the notice elsewhere in *LX* for details about the conference.)

References

Baker, Ruth (1990) Computers and independent learning: A year seven project. *Sunrise Notes, 1* (1), 6-9.

Jones, A. Nevile, L. (1990) A logo environment for teacher education students. *Sunrise Notes, 1* (1), 10-11.

Nevile, L. (1990) The Sunrise Project: Early concepts. *Sunrise Notes, 1* (1), 2-5.

Sunrise Notes is the quarterly information bulletin of the ACER Sunrise Project. Liddy Nevile is the Sunrise Project Director, and is a Research Fellow of ACER. She can be reached at Sunrise Project, ACER, P. O. Box 210, Hawthorn, Victoria 3122, Australia. Her e-mail address is liddy%otto.bf.rmit.OZ.AU@munnari.oz.au .

A. J. (Sandy) Dawson is Director of the Professional Development Program and an Associate Professor in the Faculty of Education at Simon Fraser University in Vancouver, BC, Canada V5A 1S6. He can be reached through Bitnet as userDaws@SFU.BITNET

LME 5
April 1 - 5 1991
Cairns, Australia

LME5 is a conference for those interested in mathematics education research in which computers play a significant part in providing the medium for expression of the mathematics.

In the past, LME was a meeting for those who were involved in Logo and mathematics education research. Now there are a number of other programming languages that offer extensive levels of expression of mathematical ideas. For this reason, LME5 will not be so prescriptive with respect to which language one uses.

Academic program
The themes chosen for LME5 are:

- Logo in the mathematics curriculum;
- styles of learning and strategies for teaching mathematics in a programming environment;
- expressing mathematical structures in programming languages.

Dr. Richard Noss (Institute of Education at London University) will be the program chair. Papers presented will be published by ACER and available at LME5.

Timelines
Abstracts of papers will be expected in Australia at ACER by 30 November 1990 and final papers by 31 January 1991.

Registrations will be accepted in the order in which they come, but places will be reserved for those who have attended LME on a regular basis in the past.

Location
LME5 will be held at Lake Tinaroo in the hills just outside Cairns, a major tourist resort on the northern Australian coast beside the beautiful Great Barrier Reef. The venue is one hour's drive from the coast.

As Australia is very far from home for many people, we are keen to make the voyage worthwhile. We are planning to confer from April 1 - 5 and then travel with our visitors to the Great Barrier Reef for two days of diving and snorkelling. We would like overseas visitors to travel south to Sydney with us for a half-day mathematics education plenary session with teachers on Monday April 8 as a way of gaining an opportunity to see Sydney.

Climate
This is the best time of year and, as they say in Queensland, "The weather should be perfect every day!"

Financial matters
LME5 is being planned to help those for whom the distance might be financially crippling. We have chosen a venue that offers modest accommodation only and simple fare in an effort to make the costs associated with the conference as low as possible—we trust the excursion to Australia, the wonderful surroundings and the opportunity to go to the Reef will be good compensation.

The conference will be held at Genazzano, on Lake Tinaroo. The rooms are set up for two or four people. We are planning to ask LME5 participants to share a room with a friend! In this way we can accommodate 60 people, but that is the limit. Please let us know as soon as you can if you are seriously interested in attending, and when you do, perhaps you would like to nominate with whom you would like to share if you have a preference.

Expenses
The Registration fee for the Conference will be $150 (Aust.) and for accommodation and meals, $150 (Aust.) for the period Monday to Friday. This will include the round trip to the venue from Cairns.

Registration of interest
Please complete the following and return it to us—e-mail or snail mail:

Please send full registration details to:
Name:
Institution:
Address:

I am interested in presenting a paper
YES/NO
I am interested in presenting a workshop
YES/NO
I am interested in leading a work session on:

My particular area of interest is:

If mailing this form, the address is:

LME5 c/o Liddy Nevile
A.C.E.R.
P.O. Box 210
Hawthorn
Victoria 3122
Australia

A Learning Styles Logo Environment

by Mary F. Salter

Learning is an intensely personal experience. While facts may be learned during a class period, and new insights developed, it is when students work alone, doing an assignment or working on a project, that the assimilation process occurs. Just as there are individual personality types, there are individual learning styles. Consider, for a moment, using a variety of teaching styles to make learning more natural to students with different learning styles. The goal is to make learning natural, spontaneous, and personal. Learning will be maximized when students work in an environment that is comfortable and appropriate for their learning style. Teachers will be better able to create a learning styles environment if they experience one themselves.

When reflecting on the effectiveness of lectures in presenting techniques for creating a Logo environment, I wonder if my preservice teachers will able to implement a Logo environment from our discussions of the various components. I suspect that hearing isn't always believing! Rather than relying on lectures, my preservice teachers need to experience a Logo learning environment, one that accommodates differing learning styles. By experiencing this learning environment they will see the power and potential of Logo. Experiencing is believing! Assuming there are different teaching styles corresponding to the different learning styles, preservice teachers need to experience and master techniques to create a "learning styles Logo environment" in which students can explore Logo.

A exploration creating such a learning environment was carried out in my Computers in Education class for preservice teachers. Bernice McCarthy's 4-Mat System was used as the model for learning styles. She describes four basic types of learners:

> ...type 1, the innovative learner, perceives information concretely (sensing/feeling) and processes it reflectively (thinking); type 2, the analytic learner, perceives information abstractly (watching) and processes it reflectively; type 3, the common sense learner, perceives information abstractly and processes it actively (doing); and type 4, the dynamic learner, perceives information concretely and processes it actively. (New Mindstorms #4: Styles, MMI)

The Exploration

For this exploration my initial goals and objectives were:

1. to introduce preservice teachers to the Logo programming language, especially to turtle graphics.
2. to have students become comfortable enough with computers and with Logo so that they can continue investigating Logo on their own.
3. to introduce learning styles and Bernice McCarthy's 4-mat system.
4. to acquaint students with Tim Riordon's Logo Environment.
5. to model the type of Logo environment that these preservice teachers might someday implement in their own classes.
6. to have students reflect on their own learning/ thinking styles.

The first introduction to Logo (lesson 1) was presented in the abstract/reflective style. The lecture/demonstration method was used, utilizing a computer and overhead display panel. A large portion of the first class was lecture while only a small portion was spent on the computers. As the weeks progressed, this was reversed. With each class, the amount of lecture time decreased and the amount of lab time increased. After an introduction to the turtle move commands and after students were able to start up Logo, flip the page, and save work, then students were encouraged to work in the remaining three learning styles. Three sets of learning materials were available, corresponding to types 1, 3 and 4 of the 4-mat system. Students were free to pick a style and were encouraged to experience a topic in more than one style.

Materials for the Abstract/Active style were color coded blue. The materials consisted of disks containing sample LogoWriter pages. Students were requested to view pages to find one that interested them and that they would enjoy learning how to construct. They were then asked to study the Flipside, dissect the procedures used to create the page, and then experiment with creating something similar.

The Concrete/Active style used materials that were color coded pink. The teaching method here was to allow free exploration. In addition to the handout (describing turtle moves and special keys) of the first class, pink sheets were available that listed and briefly described Logo primitives. Each pink sheet had a title, such as "music," "words and lists," or "multiple turtles." Students were expected to read the description of the command, experiment with it, and eventually build procedures that used these primitives. The Logo reference manual was also available as a resource.

The Concrete/Reflective style materials were based on the principles of guided discovery and were color coded green. These materials included the cards and booklets from the Intermediate Logo materials published by LCSI; activity sheets from *Apple Logo in the Classroom*, published by MECC; activity sheets from *Geometry Problems for Logo Discoveries*, by Margaret Moore (Creative Publications, 1984); sheets from *LCSI Logo Link Scrapbook*, published by LCSI; and article reprints from *The Computing Teacher*, offered by ISTE.

The teaching method used for the Abstract/Reflective learning style was lecture/demonstration. All students were exposed to Logo using this style in each of the six lessons.

During the six classroom lessons, student assignments were very open-ended. Students were encouraged to explore and also were requested to create something as a result of their explorations. Students went to the lab in order to experiment with Logo outside of class time. The expectation in this course is that two hours outside of class is spent for each hour in class. A specific final (graded) assignment was made during the third class so that students would begin to think about it as they continued their explorations. Students have three weeks to complete the assignment after the classroom introduction to Logo is completed (after lesson six). This specific assignment involves either creating a lesson in which Logo is used to develop a critical thinking skill (a microworld) or creating a project using procedures that involves music, multiple turtles, or animation.

My expectations of the students and teacher were:

1. that students would enjoy Logo and creating pictures.
2. that students would have more in-class "hands-on" time than in previous years and would thus be able to work more effectively outside of class.
3. that students would learn as much or more than in previous years of the lecture-only style, since the number of Logo classes increased by two.
4. that students would spend time filling out the Journal pages, in planning, and in reflecting on discoveries, and problem-solving strategies.
5. that students would develop expertise at different rates.
6. that it would be time consuming for the teacher to prepare materials for the three styles and that materials would be added each semester that Logo is taught
7. that it would be difficult for the teacher to answer student questions, since she would be unsure of what the student had experienced in previous work.
8. that there would be less emphasis on programming style and structured techniques than in past years, due to the short time of exposure in six lessons (and fewer lectures).

Observations

As the project was carried out during the Spring 1990 semester, a shift was noticed from the fourth to the sixth lesson. Students initially preferred using the green (guided discovery) and blue (dissection) methods in early lessons and moved toward pink (free exploration) in later lessons. This trend was also noticed in each individual session. Students would often start with an activity card, faithfully type each line in the early part of class, and push it aside and just try things toward the end of the class. Students used worksheets to build their confidence and get ideas, and then felt free to create.

Students seemed willing and able to state theories from their investigations. For example, under Discoveries on the Journal sheet, theories such as the following were put forth: "the larger the number for the right turn, the more acute the angle is." Students also made note of Problem Solving Strategies: "If you don't like what happened, reverse it"; "We learned to compensate for the mistakes we made"; "I drew a circle by reasoning about past experience."

Overall, students were very open to trying something new. They enjoyed choosing their own directions for exploration. Many chose to explore music, an area not included in past years. From session to session, most planned to extend and/or add to work done in previous sessions. Students seemed reluctant to ask questions of the teacher. Several students were frustrated when they did not like the result of their last command; they wanted a way to undo the last thing they tried—an opportune time for teaching procedures. In the first few lessons there was a little more frustration in some students than I had anticipated. Each had chosen the free exploration style but was unfamiliar and uncomfortable with using a computer. The suggestion that they use the guided discovery style (green) until they were more comfortable with Logo and computers eased their frustrations. The Journal page was very helpful in making this determination since I was unaware of their frustration while walking around the room. It was nice also to be able to write the suggestion to use a different style on their Journal page and give it back at the start of the next class rather than making a comment to them in front of the whole class.

Initially, students progressed through the material at a much slower pace than in past years, when only the lecture method was used. However, a deeper understanding and enthusiasm was noticed. Their final projects were of equal or better quality. Although students did develop expertise at different rates, it was not as difficult as anticipated to answer student questions.

Students did not spend as much time as I wanted in filling out the Journal pages. At the beginning of each class, which met in the computer lab, previous Journal pages were returned and blank pages were passed out. Students were asked to spend about five minutes at the beginning and at the end of each class filling out the Journal page. Students seemed eager to get to the computers and quickly, without much thought, made goals for the day. They also were reluctant to finish work at the end of class and complete the Journal page; thus, they did not reflect upon their learning styles as much as I wished. The Journal pages, which were collected and read but not graded, were very helpful to me in gently guiding and in encouraging the students. I would not want to see a grade attached, since this diverts students' thoughts to what they think the teacher wants to read. Things might be better in the situation where the computer lab is in a separate room from the classroom. Time could be allocated in the classroom before and after the time spent in the computer lab.

I was especially pleased with the way that the Logo environment was presented. First it was described verbally and on a handout, and then it was modelled in classes. Given the limited amount of time available for Logo, presenting the concept of a Logo environment abstractly and then modelling it concretely was a very effective way to present this important topic. These students have a much better intuitive and conceptual grasp of what a Logo environment should be than do past students who were exposed to the concept only through the lecture method. With elementary and secondary students, I would discuss the Logo environment in which they will work, calling attention to the ideas of experimenting, discovering, and learning from mistakes.

It was surprising that so few students asked questions. It was explained that the teacher was a resource for information but that she would not formally present material or offer suggestions until they asked for help. Students seemed to enjoy trying to figure things out for themselves more so than in the past. Students in past semesters seemed more frustrated when things did not work out immediately and they immediately sought help (often telling me that it was the computer's fault). The atmosphere was decidedly different this semester in that students did not expect to get things correct on the first try. I don't think my expectations were any different, because I've always felt that one learns more from mistakes than from doing things right immediately; but the student's expectations of themselves seemed different. I attribute this to our discussions and modelling of the Logo environment.

The Lessons

A brief description of the six lessons follows. Recall that these lessons were used with adults who had 16-hour-a-day access to the computer lab. For use with elementary and secondary students, I would suggest that a lesson be followed by several lab sessions in which students are able to experience Logo in the various learning styles.

Lesson 1
Time frame: 45 minutes lecture and 30 minutescomputer time.
Topics: introduction to Logo; instructions for starting up LogoWriter; important keys; Turtle moves (FD, BK, LT, RT, PU, PD, CG, REPEAT); flip side; teaching the turtle tricks (procedures).
Other: Students completed an exercise sheet during the 30 minutes lab time.

Lesson 2
Time frame: 75 minute lecture.
Topics: critical thinking vocabulary; Teacher's role in Logo environment; student learning goals; learning styles; Tim Riordon's Logo environment.

Lesson 3
Time frame: 20 minutes lecture and 55 minutes computer time.
Topics: assign final project and demonstrate sample project and sample lesson; discuss Logo primitives, procedures and objects; discuss structured problem solving.
Other: During lab time, students select a learning style and continue Logo explorations.

Lesson 4
Time frame: 20 minutes lecture and 55 minutes computer time.
Topics: cut and paste from command center to flip side; words and lists; variables.
Other: During lab time students select a style and continue Logo explorations. Several students have a five-minute conference with the teacher during this time.

Lesson 5
Time frame: 15 minutes lecture and 60 minutes computer time.

Topics: turtle state (heading and position); commands and reporters. Lab time as in Lesson 4.

Lesson 6
Time frame: 15 minutes lecture and 60 minutes computer time.
Topics: control structures (REPEAT, IF, IFELSE, RUN, STOP, PAUSE). Lab time as in Lesson 4.

Finally, the dynamic nature of this learning styles Logo environment must be noted, for a true Logo environment will be ever changing and evolving. Past student projects are excellent sources for additions to the learning styles materials, as are ideas from the *Logo Exchange*. Be ever on the lookout for new methods and project ideas. Student involvement and enthusiasm are the reward.

References

(1982). Logo full speed ahead: A learner driven language. *The Computing Teacher, 10*(3), 14-63.

Apple Logo in the classroom. MN: Minnesota Educational Computing Consortium.

New Mindstorms #4, Styles. MMI print materials, pp. 7-8.

(1988-90) The LCSI Logo Link Scrapbook, Volumes 2, 3 and 4. VT: Logo Computer Systems Inc.

Joyce, B. and Weil, M. (1986). *Models of teaching*, 3rd Edition. New York, NY: Prentice Hall.

Moore, M. (1984). *Geometry problems for Logo discoveries.* CA: Creative Publications.

Riordon, Tim. (1982) Creating a Logo environment. *The Computing Teacher, 10*(3), 48.

Mary Salter
Computer Science 164
Franciscan University of Stubenville
Franciscan Way
Stubenville, OH 43952
CompuServe address: 76354,3177

Logo Connections

Digitized Images

Glen L. Bull and Gina L. Bull

In this year's columns we are discussing practical multimedia suggestions for Logo. The term "multimedia" refers to the use of several different media in combination. In each column we will review a different multimedia option that can enhance Logo.

In our local school system an earth science teacher has incorporated images of geologic samples in an instructional program that he has developed. A biology teacher has added images of biologic specimens to a program that she developed for a cell biology course. A sixth-grade class developed a Logo program describing their school that contains images of their school and the surrounding neighborhood. A disk containing the program was exchanged with a classroom at another school to create an electronic field trip.

Capturing Images

It would have been possible to write a turtle graphics program to draw these images, or use a paint program to create them. However, the teachers in each class used a video digitizer to capture an image that could be imported into Logo. A video digitizer allows the image from a video camera or a videotape to be converted to a computer image.

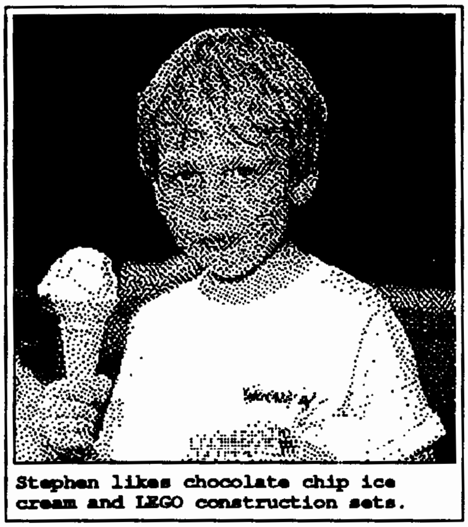
Not long ago relatively expensive peripherals were required to digitize an image. Today a video digitizer can be purchased for an Apple IIe for as little as $129.00. This option consists of a computer card that is placed in one of the slots in the computer. A videocassette recorder (VCR) or camera can be connected to one of the video inputs on the board. A computer program that is provided with the video digitizer is used to capture images from the camera or VCR. These images can then be imported into Logo.

This process requires a VCR or camera to provide a source of video images. However, most school systems have one or both of these. The low cost of video digitizers makes this one of the most economical multimedia options available. A number of firms make video digitizers, but we are most familiar with *ComputerEyes* digitizers. One advantage of this line of digitizers is that models are available for Apple, IBM, and Macintosh computers. Both a monochrome and a color version are available for each brand of computer. We have used the monochrome and color versions for the Apple computer, and the color version for IBM computers. We have found the software reliable and easy to use.

Logo Connections - continued

A Logo Yearbook

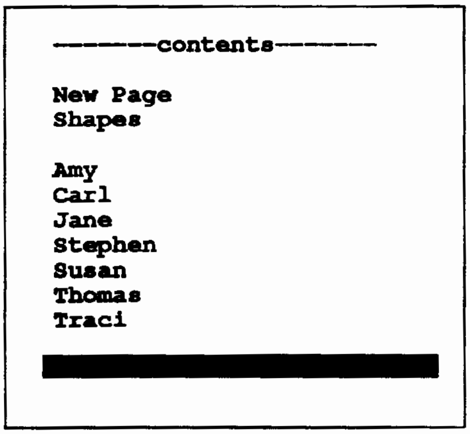
The ability to add digitized images to Logo programs opens up a wide range of classroom activities. For example, the class can make an electronic yearbook with LogoWriter. Use the digitizer to capture each child's image. Place one digitized image on each LogoWriter page, and use the student's name as the name of the page. Use the LogoWriter word processor to add notes about the student at the bottom of the page.



Stephen likes chocolate chip ice cream and LEGO construction sets.

(Long-time readers of the *Logo Exchange* will recall that the only issue for which we have failed to submit our column was the month Stephen was born. Now, as you see, both he and Logo are considerably more mature.)

The *LogoWriter* yearbook can be used for an open house or on PTO nights, or exchanged with pen pals from another class, or simply used as an electronic record. Although this type of project can be completed with any version of Logo, LogoWriter in particular lends itself to this kind of application because of its "page and book" metaphor.

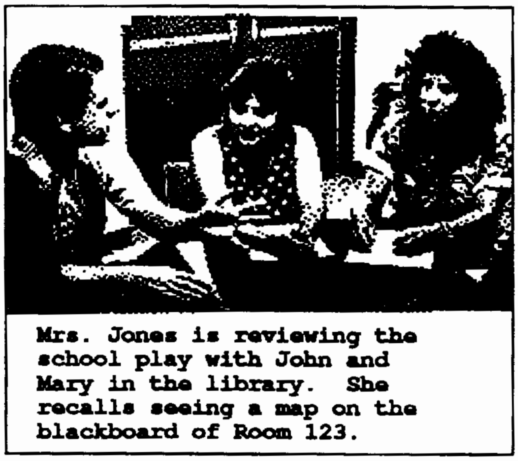


```
--------contents--------

New Page
Shapes

Amy
Carl
Jane
Stephen
Susan
Thomas
Traci
```

A Logo Adventure

Where in the World Is Carmen Sandiego? is a popular educational game in which students must track down a criminal by identifying clues associated with various regions of the world. This has proved to be such an effective learning tool that several states have even commissioned versions, such as *Where in Florida Is Carmen Sandiego?*

A video digitizer can be used to create a similar sort of Logo treasure hunt for a school. Digitized images of various rooms of the school are imported into different pages of LogoWriter. Clues to a mystery are placed in different rooms. (In this metaphor, each LogoWriter page represents a different room.) Each room also contains a hint as to where to search next. For example, the search for the school mascot has led to the media center. Mrs. Jones remembers a map on a blackboard that was not erased after history class.



Mrs. Jones is reviewing the school play with John and Mary in the library. She recalls seeing a map on the blackboard of Room 123.

This type of project can be developed using images created with turtle graphics, or imported from a paint program. However, a class often enjoys using real images from its own school.

It is possible to navigate from "room" to "room" in this LogoWriter adventure by using the **GetPage** command. For example, the LogoWriter page containing an image of Room 207 might be named "207." The command to travel to that room would then be:

```
GetPage 207
```

However, it's probably not a good idea to name pages with numbers. In some versions of LogoWriter, you cannot get such pages from the Contents Page. You must use the GetPage command. We preferred to create a procedure called "Room" that is placed on the flip side of a LogoWriter page named "Tools."

```
TO Room :Number
GetPage :Number
End
```

Procedures such as **Room** that are on the **Tools** page must be loaded at the beginning of the Logo treasure hunt by typing:

```
GETTOOLS "Tools
```

This makes it possible to use the **Room** procedure on any LogoWriter page. To travel to this location, it is now possible to type:

```
Room 207
```

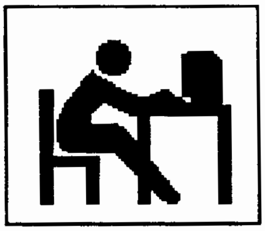
If the room has a name rather than a number, it must be preceded with a quotation mark:

```
Room "Library
```

A Logo treasure hunt can be an interesting way for newcomers to learn about a school. We think there is one important difference between this type of Logo adventure and the commercial Carmen Sandiego program. In Carmen San Diego the facts to be learned have been selected by others. In a Logo adventure, the class members can choose what they think is important to know about their school. This also can provide a means of conveying a school culture from one generation to the next.

Images in Art and Science

Digitized images can be used in other ways. Elaine Blitman, a colleague of ours at Punahou School in Hawaii, worries about the possible effect of computer graphics on the artistic abilities of future generations of children. With an image digitizer, it is possible to capture original drawings by children and import them into Logo. Artistic efforts created with crayons, water colors, or chalk on the blackboard can be captured in this way. For smaller drawings it may be necessary to add a closeup lens to the video camera. This will allow it to capture details of smaller images. For example, the following digitized drawing might be used as a symbol for the school computer lab.



Recently middle and high schools in our area have been acquiring devices which allow a video camera to be attached to a microscope. The purpose of these devices is to allow the output of the microscope to be displayed on a television monitor so that the entire class can view it at one time. However, when the output of the video camera is transmitted to the image digitizer, it can be captured and imported into Logo. Biology classes have captured images of cells, while geology classes have digitized pictures of minerals. Projects which might be envisioned with Logo could include a rogues gallery of microscopic life forms, or an electronic box of digitized minerals.

Another project created by a class involved digitization of a geologic map of the surrounding area. Going from one *LogoWriter* page to the next allowed the turtle to travel from one section of the map to another. Logo procedures for the points of the compass (North, NorthEast, etc.) were created for this project.

Digitizing Tips

The current generation of digitizers does have some limitations. Video digitizers such as the *ComputerEyes* system require about 20 seconds to digitize an entire image, so an object must remain still for that length of time before it can be successfully digitized. Some digitizers can digitize an image in 1/30th of a second, making it possible to capture a single video frame of a moving object, but these systems are considerably more expensive. We have found that more expensive VCRs with four video heads can sometimes successfully freeze a video image when the pause button is pressed so that the picture can be digitized. The less expensive VCRs more commonly found in schools often fail to produce a stable image when the pause button is pressed. As an alternative, a half-minute videotape of the same image (using a tripod to ensure the camera is steady) offers a means of capturing images for the digitizer, and avoids the necessity of using the pause button. However, this method only works for objects which are not moving.

More recently video still cameras have gained popularity. These cameras store still video images on a disk. One disk will hold about 50 shots. The "picture" captured can be displayed on a television screen or monitor. Because the image is stable, it is ideal for digitization. This provides a convenient way to capture images of moving objects. Since the camera is lightweight, it also provides a convenient way of capturing images on field trips. One of the best known of an increasing number of still video cameras is the Canon Zap camera. Its current educational price is about $500. It works well with the *ComputerEyes* digitizer and is available from Digital Vision and other suppliers.

Logo Connections - continued

Another limitation of digitizers is that the resolution of the captured image is no greater than the resolution of the computer screen used to display the image. This varies depending on the type of computer and version of Logo that is used. The display of an Apple IIe has less resolution than an IBM or a Macintosh. The display of an IBM with a CGA monitor has less resolution than an IBM with a VGA monitor.

This raises another issue. Each type of computer has several different graphics formats. The *ComputerEyes* software allows the image to be digitized in several different formats. You will need to select a format that is compatible with the version of Logo you are using. For example, if you are using an Apple II+ or IIe computer, the chances are that you should digitize pictures in the standard high resolution format (sometimes abbreviated as HiRes) employed by the Apple.

Graphics Editors

After an image has been digitized, it can be edited with a paint program such as the ones described in last month's column. The paint program serves as a "graphics editor" which makes it possible to revise the image and add additional elements as required. After any editing is completed, the final image can then be imported into Logo.

Some have expressed concern as to whether children should be allowed to use paint programs, for fear that they may no longer find turtle graphics attractive. Our sense is that each methods of generating computer graphics has its own strength, and that these strengths complement one another. Fractals can not be generated with a paint program, for example, and we anticipate that classic Logo programs such as **SpinSquare** will remain popular with many generations of children to come.

More importantly, we doubt whether technologic censorship is possible, even if it were desirable. In our area paint programs are available on computers in the children's section of the public libraries, on home computers, and in many other places. There is a well-known story about the first introduction of Logo on TI 99/4 computers in a private school in Texas. A scope and sequence chart was developed outlining the grade in which each aspect of Logo should be taught. As the story is told, the children quickly subverted this system by passing "forbidden" knowledge about Logo to children in earlier grades.

In today's world, it may prove difficult to suppress knowledge of technologies that are widely available in the world outside the schools. The technology itself evolves so rapidly that censorship proves a difficult task. Therefore it may prove more fruitful to prepare children for tomorrow's world by taking advantage of these technologies rather than ignoring them.

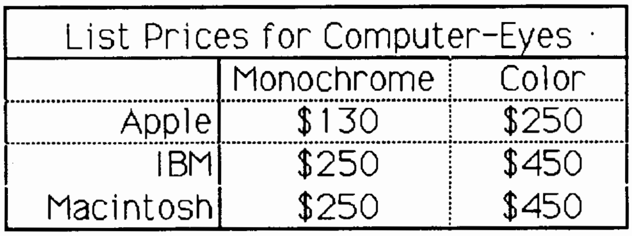
Future Images

Researchers predict that as computers and television systems mature, eventually computer images and television images will be interchangeable. A few years ago video digitizers were expensive laboratory tools beyond the reach of the public schools. Now almost any school with an Apple or IBM computer can digitize images for importation into Logo. The term "multimedia" truly is fulfilling its promise of combining more than one medium.

Note: The *ComputerEyes* digitizers for Apple, IBM, and Macintosh computers are available from Digital Vision:

Digital Vision, Inc.
270 Bridge Street
Dedham, MA 02026
(617) 329-5400

The following table lists some of the suggested retail prices for various models of *ComputerEyes* digitizers. However, some vendors offer substantial educational discounts. For example, we acquired the IBM color digitizer which has a list price of $450 for $375.



List Prices for Computer-Eyes

	Monochrome	Color
Apple	$130	$250
IBM	$250	$450
Macintosh	$250	$450

It is important to note that the color digitizer for the Apple can only be used with an Apple IIgs. However, the monochrome digitizer can be used with either an Apple IIe or an Apple IIgs.

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

BITNET addresses:
Glen: GBULL@ VIRGINIA. Gina:
GINA@VIRGINIA.

Extra For Experts

Conversations With Logo: Part II
(as overheard by Michael I. Tempel)

LogoWriter: Well, hello again. How've you been?

Person: Fine, thanks, but I have another Logo problem.

LogoWriter: What's up?

Person: I'm trying to write a "guess my number" game. The program "thinks" of a number between 0 and 100, and we see how many turns the player needs to get it.

LogoWriter: 50

Person: Huh?

LogoWriter: My first guess is 50. Is that high, low, or right?

Person: Wait! We're not playing the game yet! I just want to tell you about my problem programming it in Logo!

LogoWriter: Oh. Too bad. I like to play that game.

Person: Well if you help me get the program working we can play it all you want.

LogoWriter: Great! What's your problem?

Person: Well here's my program:

```
to game
name random 101 "number
make "guesses 1
get.answers
print [Do you want to play again?]
name readlist "answer
ifelse :answer = "yes
[game]
[print "bye]
end

to get.answers
print [What's your guess?]
name readlist "answer
if :answer = :number
[(print [Right! in] :guesses
"guesses)  stop]
if :answer > :number
[print [Too high]
make "guesses :guesses + 1
get.answers stop]
if :answer < :number
[print [Too low]
make "guesses :guesses + 1
get.answers stop]
end
```

Person: It randomly picks a number between 0 and 100. Then it asks for a guess. If you get it, you see a "that's right" message and the game is over. If not, then the program checks to see if your guess is high or low and tells you. That's where the problem is.

LogoWriter: Let's try it.

Person: OK.

game

LogoWriter: What's your guess?

Person: 23

LogoWriter: < doesn't like [23] as input in get.answers

Person: Why not?

LogoWriter: Because [23] isn't a number. < can only compare numbers with each other.

Person: If it's not a number, then what is it?

LogoWriter: It's a list.

Person: But I just typed in 23. That's a number.

LogoWriter: Yes, but readlist takes what you type and reports it as a list. In this case, you typed single number, but readlist will read any combination of things and report it as a list. It's very flexible.

Person: Yeah, but I need to read 23 as a number.

LogoWriter: Well, try readnumber.

Person: Ok. I'll edit my procedure and change readlist to readnumber. There. Now . . .

game

Extra for Experts - continued

LogoWriter: What's your guess?

Person: 23

LogoWriter: I don't know how to readnumber in get.answers

Person: But you told me to try readnumber!

LogoWriter: Well sure, but you'll have to write it first.

Person: Thanks a lot! Where do I start?

LogoWriter: What do you want readnumber to do?

Person: I want it to read what I type at the keyboard . . .

LogoWriter: Readlist does that!

Person: I know, but I want it to read a number, not a list.

LogoWriter: Readlist just reads what you type. It's not how it reads it that counts, it's how it reports it. You want a procedure that reads what you type and reports a number.

Person: Well I'm not quite sure what to do, but I'll get started. I'll use readlist . . .

```
to readnumber
output do.something.with readlist
end
```

LogoWriter: Good start. Now do.some.thing.with needs to turn the list into a number.

Person: Wait. The thing inside the list is a number. Can't I extract it?

LogoWriter: Sure. You could use . . .

Person: First!

LogoWriter: . . or last. If there's only one thing in the list it doesn't matter.

Person: Ok. So . . .

```
to readnumber
output first readlist
end
```

LogoWriter: Now try it

Person: game

LogoWriter: What's your guess?

Person: 50

LogoWriter: Too high
What's your guess?

Person: 25

LogoWriter: Too high
What's your guess?

Person: Ok let's stop the program. I see that it's working.

LogoWriter: But I want to play more.

Person: Oh all right!
game

LogoWriter: What's your guess?

Person: 50

LogoWriter: Too low
What's your guess?

Person: 75

LogoWriter: Too high
What's your guess?

Person: 67

LogoWriter: Too high
What's your guess?

Person: 56

LogoWriter: Too low
What's your guess?

Person: 59

LogoWriter: Too low
What's your guess?

Person: 61

LogoWriter: Too high
What's your guess?

Person: 60

LogoWriter: Right! in 7 guesses
Do you want to play again?

Person: yes

LogoWriter: bye

Person: Wait a minute! I said yes, I did want to play again. You said bye. What's wrong?

LogoWriter: It's the same problem as before.

Person: Huh?

LogoWriter: "yes doesn't equal [yes]

Person: Oh, I see, just like 23 doesn't equal [23]

LogoWriter: That's right. You could fix it by . . .

Person: I know! I could change the end of game to

```
ifelse :answer = [yes]
       [game]
       [print "bye]
```

LogoWriter: Sure. That'll work. Or instead of that change, you could leave the ifelse command alone and change the previous line to use your readnumber procedure instead of readlist:

```
name readnumber "answer
ifelse :answer = "yes
       [game]
       [print "bye]
```

Person: But "yes isn't a number.

LogoWriter: Well, your readnumber procedure is really a readword procedure. It reads any words, not just numbers. If readlist reports [50] then readnumber reports 50. If readlist reports [yes], then readnumber reports "yes.

Person: That makes sense. I should probably call the procedure readword instead of readnumber.

LogoWriter: Ok. Let's play the game some more!

Person: OK.
game

LogoWriter: What's your guess?

Person: 50

LogoWriter: Too low
What's your guess?

Person: 75

LogoWriter: Too low
What's your guess?

Person: 87

LogoWriter: Too high
What's your guess?

Person: 81

LogoWriter: Too high
What's your guess?

Person: 78

LogoWriter: Right! in 5 guesses
Do you want to play again?

Person: Sure! why not?

LogoWriter: bye

Person: oops!

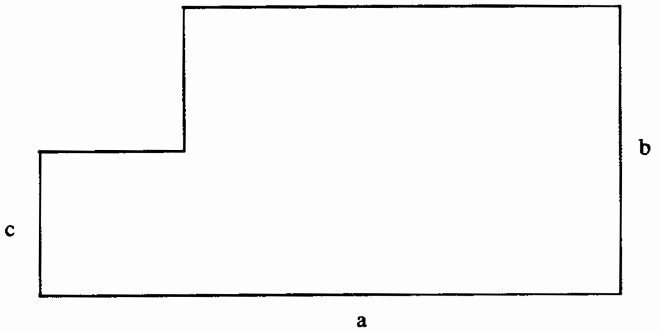
Michael Tempel
LCSI
330 West 58th Street
New York, New York 10019
212-765-4780

Logo: Search and Research

Strategies for Solving Turtle Geometry Problems

by Douglas H. Clements

Jolene and Estaban are using the turtle to draw this shape.



Estaban chooses input to FORWARD of 80, 60, and 30 for sides a, b, and c. Jolene says, "The bottom (a) was 80, so the top two must be 65 and 15."

Two Strategies

Recall last month's column on top-down and bottom-up styles of programming. These differences in *style* are reflected in different strategies students use to solve turtle geometry problems.

Researchers often describe two strategies. One team called the first the "visual approximation strategy" (Dytman & Wang, 1984). This involves many small steps, few overt planning behaviors, and little use of guides such as horizontal or vertical lines (e.g., the sides of the screen). It reflects a bottom-up approach. In contrast, the "analytic strategy" involves a small number of larger size steps, more planning, and more use of guiding lines. It is similar to a top-down approach. More first- through fifth-grade students used the visual approximation strategy. Those who did use the analytic strategy, however, were more accurate in solving problems. Use of the analytic strategy seemed to raise performance in older students who were low in Logo proficiency. Unfortunately, no low ability young students used this strategy. So, the researchers could not determine if this would have helped them.

Working with sixth grade students, Kieran, Hillel, and Erlwanger (1986) identified two similar strategies.

1. Visual strategy: Students choose inputs on the basis of visual cues, independently of what has gone on before. Students often say, "It looks like...." (This is what Estaban did.) The visual strategy is not related to the ability to visualize. Rather, it refers to reliance on a figure's visual aspects.

2. Analytical strategy: Students choose inputs on the basis of mathematical relationships inherent in the figure's geometry. Students often say something like, "Because this angle is 60°, then...." (This was Jolene's strategy.) Such analysis often involves the discovery of general principles underlying concrete figures and Logo code (Hillel & Kieran, 1988).

This is not a matter of being right or wrong. Sometimes a visual strategy may be more "successful," in the sense of getting a program to run (Hillel & Kieran, 1988; Kieran et al., 1986). As an example, consider the task of drawing an equilateral triangle with a horizontal base. A student might choose RIGHT 30 for the initial turn, explaining that, "It looks a little less than 45 so it is 30." This is correct, but not based on analysis of angle relationships. On the other hand, some students choose 60 because "it is 120 minus 60." Here, there was an attempt to relate angle measures, although it was in error. This is an analytic strategy.

Why do students use visual strategies?

1. They may see programming too much as "drawing with the turtle." People do not usually see drawing as an analytic task. Overuse of "playing turtle" might also de-emphasize analysis.

2. Students may believe that drawings that are "close" are "good enough." From this perspective, visual strategies often "work."

3. The mere presence of geometric figures may invite visual strategies. Figures should provide cues, of course. But students often treat cues as facts. They do not verify them or try to understand how they relate to one another.

For these reasons, students prefer to use visual strategies. This is fine, as far as it goes. Visual strategies help students gain control over the environment and enjoy success. They can even serve as catalysts for analytical approaches (e.g., inverse relationships), especially if the problem is not very complex (Hillel & Kieran, 1988).

The continued use of visual strategies, however, often inhibits students from arriving at mathematical generalizations (Hillel & Kieran, 1988). The problem is that students using only visual strategies are walking down a dead-end street. They choose each command independent of their previous choices. They rarely build up rich, connected knowledge. Knowledge about geometry arises only if students use an analytical strategy. That is, they must look for explicit angle and length relationships. A visual strategy may produce Logo code that draws the correct figure. It will not, however, lead to generalizations because it doesn't capture the *properties* of the figures it draws (Hillel & Kieran, 1988).

Encouraging Analytic Strategies

Concerned about this, Hillel and Kieran designed a restricted Logo environment to encourage analysis. Sixth-grade students used the following simple primitives:

MOVE, a slow FORWARD movement that left no trace;

TRT, a slow right rotation;

TLT, a slow left rotation;

TEE :N, a state transparent "T" of height :N; and

VEE :N for a state transparent "V" with arms of length :N and an interior angle of 60°.

Hillel and Kieran omitted FORWARD and BACK deliberately to sabotage students' "drawing" perception of turtle geometry. They gave students tasks such as shown in those in the figure below. They worked in pairs with little or no direct instruction.

On the Task 1 (see below), all students attempted a solution using visual strategies. For example, one pair's first attempt was:

```
TRT 45
TEE 50
TLT 135
TEE 50
TLT 135
TEE 50
```

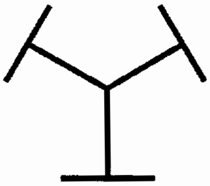
They adjusted this based on the feedback from the screen to:

```
TRT 45
TEE 50
TRT 135
TEE 50
TRT 135
TEE 50
```

These students seem to be focusing on vertical symmetry of the figure and the orientation of the bottom "T."

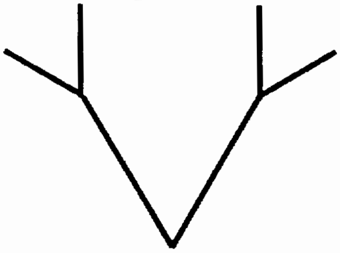
On Task 2 (see below), some students attempted analytic solutions. They focused on non-overlapping figures, equal horizontal and vertical distances between the crosses, and symmetry relative to the vertical axis.

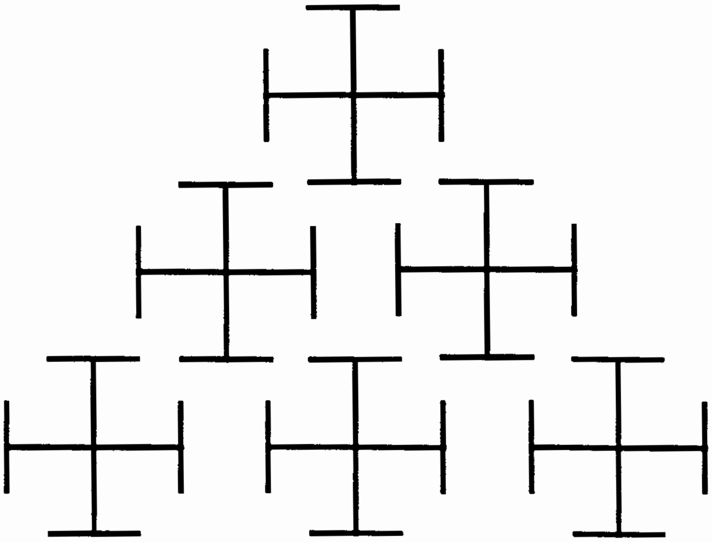
On task 3 (see below), Stephie tried to move from one small "V" to the other by using a turn of 90, then 45. Neither worked. Only then did she look closely at the figure and try



Task 1

Task 3





Task 2

to analyze it. She eventually moved down the one side of the large "V" and turned TRT 120, explaining "180 - 60 = 120." So, she showed a movement from visual to analytic strategies.

This shift usually occurred when students could not solve the task using only visual strategies. On the next task, however, students would often go back to visual strategies. This tendency persisted throughout the entire year.

This is similar to research findings on the van Hiele theory. Elementary students can think about a figure's properties, but typically only do so as a result of probing (Hillel & Kieran, 1988). Left to their own devices, they prefer the safety of holistic, visual thinking. There is one important difference between the van Hiele and the Logo research. In Logo, the shift to analytical strategies was fostered by the feedback provided by the turtle. Typically, it did not require the probing of the interviewer. So, Logo can provide the stimulus to shift to another mode of reasoning when necessary. This feature is lacking in traditional geometry environments. However, students did not always accept that the structured tasks required analytical thinking. They seemed to believe that "more or less" precision was adequate. Also, their screens' low resolution allowed them to believe, for example, that 10 35° rotations returned the turtle to its initial orientation. These results suggest several recommendations for teaching.

Implications for Teaching Analytic Strategies

1. Choose tasks carefully. There may be little reason for students to abandon visual approaches unless you present tasks that require analytical strategies. They are more likely to solve open–ended or unstructured tasks with visual strategies.

2. Model and discuss both visual and analytic strategies. How students "see" a task affects how they will go about solving it.

3. Take care to help students build and think about path-command correspondence—the connection between geometric paths drawn by the turtle and the Logo commands that produce these paths (Battista & Clements, in press; Clements & Battista, 1989).

4. Monitor students' problem-solving progress. Students may begin by analyzing a figure, only to "patch up" their code later using visual strategies and screen feedback. Encourage re-*analysis* instead.

5. Have students justify their solutions not on the basis of drawn figures, but on the fit between their geometric analysis of the figure and their Logo code.

6. Consider alternate grouping arrangements. Students showed a dramatic increase in commitment to solve problems when the researchers let them work on their own computers (Hillel, Kieran, & Gurtner, 1989). If you use pairs, encourage them to disagree and then resolve their disagreements (Nastasi, Clements, & Battista, 1990).

7. Clarify goals through class discussions, assignments, and "conferences" at the computer. The goal of—and even the motivation for—turtle geometry work should not just be to "draw a figure," but to explore and learn about mathematics.

References

Battista, M. T., & Clements, D. H. (in press). *Logo geometry*. Morristown, NJ: Silver Burdett & Ginn.

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

Dytman, J. A., & Wang, M. C. (1984, April). *Elementary school children's accuracy and strategy use in problem solving*. Paper presented at the meeting of the American Educational Research Association, New Orleans, LA.

Hillel, J., & Kieran, C. (1988). Schemas used by 12-year-olds in solving selected turtle geometry tasks. *Recherches en Didactique des Mathématiques, 8/1.2*, 61-103.

Hillel, J., Kieran, C., & Gurtner, J.-L. (1989). Solving structured geometric tasks on the computer: The role of feedback in generating strategies. *Educational Studies in Mathematics, 20*, 1-39.

Kieran, C., Hillel, J., & Erlwanger, S. (1986). Perceptual and analytical schemas in solving structured turtle–geometry tasks. In C. Hoyles, R. Noss, & R. Sutherland (Ed.), *Proceedings of the Second Logo and Mathematics Educators Conference* (pp. 154-161). London, England: University of London.

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

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

Global Logo Comments

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

Logo Exchange Continental Editors

Africa
Fatimata Seye Sylla
UNESCO/BREDA
BP 3311, Dakar
Senegal, West Africa

Asia
Marie Tada
St. Mary's Int. School
6-19, Seta 1-chome
Setagaya-ku
Tokyo 158, Japan

Australia
Jeff Richardson
School of Education
GIAE
Switchback Road
Churchill 3842
Australia

Europe
Harry Pinxteren
Logo Centrum Nederland
P.O. Box 1408
BK Nijmegen 6501
Netherlands

Latin America
Jose Valente
NIED
UNICAMP
13082 Campinas
Sao Paulo, Brazil

As 1991 begins, it is time to think about upcoming Logo conferences. The fifth Logo and Mathematics Education conference will be held from March 30 to April 4 in Cairns, Australia. The LME5 is a conference for those interested in mathematics education in which computers play a significant part in providing the medium for expression of the mathematics. See details in the announcement elsewhere in *LX*.

A major Logo conference is the EuroLogo 91. This year's conference will be held from August 27 to 30 in Parma, Italy. (For details, see the announcement in the November issue of *LX*.) The EuroLogo conference is a biannual conference that focuses on educational applications of Logo and Logo-related programming languages. The Parma conference will carry on the activity that first started in Dublin in 1987 and was then followed by Ghent in 1989. Just as in previous conferences, the main characteristic of Euro Logo 91 will be the presentation of works by European and non-European researches. Classroom experiences with Logo, Logo prospects and research, programming environments, and teacher training in relation to Logo will be the four major themes of the conference.

International Informatics Olympiad

European editor Harry Pinxteren reports on the results of the International Informatics Olympiad held in Minsk, USSR. In a contest where most participants used Pascal and BASIC, Logo programmers took two of the top three awards. Harry's report follows.

At the International Informatics Olympiad in Minsk, USSR (July 16th-20th), two Logo students from the Netherlands have been awarded in the following categories:

Fleur Kelpin (16): best female participant
Fieke Dekkers (15): best youngest participant

Fieke Dekkers also received a special prize for the "most persistent" participant, because she succeeded in coping with severe problems with her computer. All other participants made a good effort, especially those from the western countries who had to work with a Cyrillic keyboard with some of the Latin characters missing.

There was a general agreement that Fleur Kelpin wrote the most elegant solution for the main problem of the contest—a sorting game. She found the absolute minimal solution, where all others wrote a heuristic one. So Fleur should have won the prize for best student. However, the computer could not solve her solution in real time. Therefore a heuristic solution, using PASCAL, finally was awarded after a long discussion by the jury!

This success is even more amazing because the Dutch Logo students had only a four-day training session in LCN Logo as a preparation and did not have any prior Logo experience at all.

All in all they had a preparation period of less than one month, where most participants had a special training of years—in one case, nearly six years!

The power of LCN Logo as a functional language (as based on the Lisp dialect SCHEME) received a lot of positive media coverage. As Fleur said,

> I never realized I could reach such results—expressive power and high level abstraction—in such a short period of time, using Logo as a programming language. I always thought it was meant for children!.

In an international setting predominated by procedural languages (BASIC, Pascal) these outstanding results turned

Global Logo Comments - continued

out to be an excellent promotion for the use of Logo in education. For that matter, it became a massive refutation of the kid-image of this language, which still prevails in most countries in the world—until this Olympiad that is.

Next year's contest will be in Athens. LCN Logo will be upgraded by then and our students will have one year of experience. There will be one new problem: more participating countries will send proficient Logo programmers. There were very enthusiastic reactions from representatives from about 20 countries. So next year's contest will indeed be a challenging one.

11th Annual
International Contest

What is the International Computer Problem Solving Contest (ICPSC)?
To some schools it is a chance to challenge their top computer and mathematics students in an annual computer problem solving event that compares their solutions to the best in the world. To others it is a challenging set of problems to be used as enrichment material for computer programming classes. However you choose to use the ICPSC, it can be a valuable resource for any computer programming teacher.

What is unique about the ICPSC?
The ICPSC combines the art of problem solving with the skill of computer programming. Our contest challenges teams (from one to three students each) to create short, original solutions to a set of five problems within a two-hour period. All problems can be solved with short computer programs written in any language on any computer system. The problems fall into five categories: 1. Computation, 2. Simulation, 3. Graphic Patterns, 4. Words, and 5. Mind Benders.

Can an individual teacher in a school become a contest director for his/her school?
Yes! We want to challenge and enrich your students; you don't have to make it a major district-wide event if that would stop you from doing it. Of course, the more students in your local area who can get involved, the better. But if you only have one student in your whole school who wants to compete, that is all you need.

Can Logo be used?
Our original contest problems were not appropriate for Logo students. So we added a Logo Division in all three age groups.

What are all the age groups?
Elementary Division (Grades 4-6)
Junior Division (Grades 7-9)
Senior Division (Grades 10-12)
These are now offered in either the Open contest (all languages) or the Logo contest.

When is the contest held?
The 11th Annual ICPSC is scheduled for Saturday, April 27, 1991. Friday, April 26 and Monday, April 29, are alternate dates that can be used if Saturday is impossible. This year's set of contest problems will be mailed to the contest director on April 1, 1991. Sets of previous problems and solutions for students to practice on are always available.

How can I get more information about the ICPSC?
The Computing Teacher annually publishes the results of the ICPSC in the Dec/Jan issue. Also the *Logo Exchange* is publishing stories on the Logo contest. You may receive a free copy of ***Compute It!***, the official newsletter of the ICPSC, which is published in November each year, by writing to: Dr. Donald T. Piele, ICPSC, P.O. Box 085664, Racine, WI 53408, Ph 414-634-0868.

Computers.

Is your school seaworthy, or is it ready for dry-dock ?

As the 90s pull us further into the technology age, it's becoming increasingly difficult to assess your school's ongoing computer needs. ISTE's *Evaluating Computer Integration in the Elementary School: A Step by Step Guide* equips you to understand the present and plan for the future.

Five main chapters pilot you through your evaluation:

1. An overview of computer education and program evaluation,
2. Planning the Evaluation,
3. Collecting the Information,
4. Interpreting the Information, and
5. Reporting the results.

Evaluating Computer Integration in the Elementary School: A Step by Step Guide is designed for evaluation of computer use integrated into standard curricular areas rather than individual computer courses. **$15.00** plus $3.25 To order, contact ISTE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905; ph. 503/346-4414.

Anchor your schools computer integration program, with ***Evaluating Computer Integration in the Elementary School: A Step by Step Guide*** and sail smoothly through the 90's.

Make computer education a win/win situation in your classroom.

Research indicates inequities exist in nearly all classrooms. *Yes I Can: Action Projects to Resolve Equity Issues in Educational Computing* clearly explains these issues and details actions to achieve positive results. The projects target the problems of females, minorities, and kids with special needs, whose access to computer labs is often limited.

Drawn from the results of several action projects by the Educational Computer Consortium of Ohio (ECCO), *Yes, I Can* shows the way to:

- improve student's self-esteem
- use a word processor as a motivational aid
- emulate role models' successes
- improve administrators' understanding of the role of computers in a educational setting

ISTE'S *Yes I Can* also details how you can get other teachers involved in computer equity. References for projects are included with the book.

Yes I Can: Action Projects to Resolve Equity Issues in Educational Computing $15.00 plus $3.25 shipping. Contact ISTE 1787 Agate St., Eugene, OR 97403-9905; ph. 503/346-4414.

The world is full of inequity, but when it comes to changing it in your class—yes you can!

Logo Exchange
ISTE/University of Oregon
1787 Agate Street
Eugene, OR 97403-1923