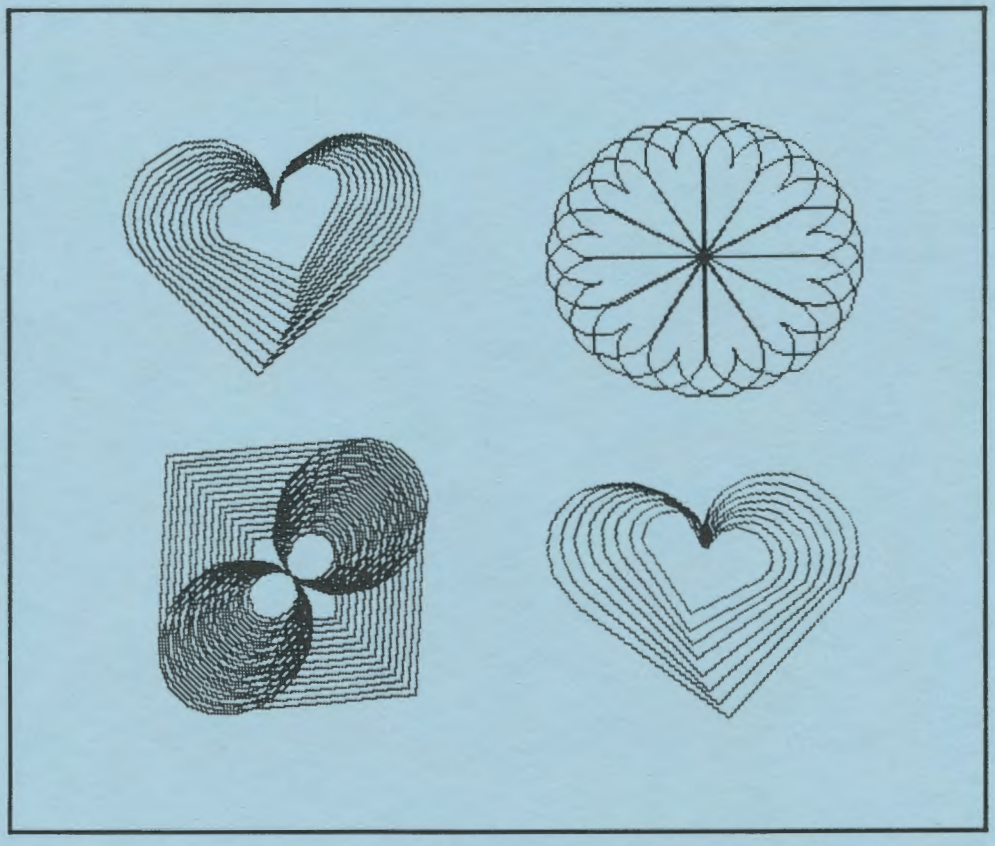
Journal of the ICCE Special Interest Group for Logo-Using Educators

LOGO EXCHANGE

FEBRUARY 1989 VOLUME 7 NUMBER 6



International Council for Computers in Education

Logo Exchange

The Journal for Logo Activities Worldwide

Now published by the International Council for Computers in Education, *Logo Exchange* brings you exciting ideas from top Logo educators throughout the world.

Each of the nine issues includes:

- Logo articles *by* classroom teachers *for* classroom teachers;
- Columns which emphasize practical ideas for the use of Logo at the primary and intermediate level;
- Articles focusing on the use of Logo in language arts and math;
- Updates on Logo research;
- A look at Logo-like activities, software and hardware;
- Articles by Logo leaders from throughout the world;
- Reports on Logo activities world-wide.

The *Logo Exchange* is the journal for ICCE's Special Interest Group for Logo-using educators. SIGLogo members are invited to participate in local, regional, and national meetings and to contribute to the flow of ideas through the *Logo Exchange*.

Logo Exchange is published monthly September through May. SIGLogo membership is $24.95 for ICCE members (include membership number on order) $29.95 for non-members. Add $5.00 for non-U.S. memberships. All billed orders are charged $2.50 for handling.

To order, contact:
ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905; ph. 503/686-4414.

Join SIGLogo before February 29th, 1989 and get the Logo Alphabet on disk FREE!

The Logo Alphabet is a set of procedures that draws all of the letters of the alphabet in the size of your choice. They were developed by Tom Lough, founding editor of the *Logo Exchange*.

You must use this order form to take advantage of the special offer to get the Logo Alphabet free when either joining SIGLogo (before February 29th, 1989) or extending your current membership for a year.

___*Yes*, I would like to join SIGLogo and receive the Logo Alphabet disk free of charge.
___*No*, I do not wish to join SIGLogo at this time, but I want the Logo Alphabet disk. I am enclosing $7.50.

Please check the version you want:

☐ Apple Logo (#LA1) ☐ Apple LogoWriter 1.1 (#LA3) ☐ Terrapin Logo (#LA5)
☐ Apple Logo II or LCSI Logo II (#LA2) ☐ Apple LogoWriter Version 2.0 (#LA4) ☐ Terrapin Logo Plus (#LA6)

SIGLogo Membership (includes the *Logo Exchange*)	U.S.	Non-U.S.
☐ ICCE member	$24.95	$29.95
☐ Non ICCE member	$29.95	$34.95
ICCE Membership (includes *The Computing Teacher*)	**U.S.**	**Non-U.S.**
☐ 1 year	$28.50	$31.00

Checks, VISA and Mastercard accepted. Add $18.00 per membership for AIRMAIL shipping of *Logo Exchange* outside the U. S. and Canada. Add $2.50 for processing if payment does not accompany your membership dues.

___Payment enclosed. Amount $_____(US Funds)

___ Charge: ___Visa ___Mastercard

Name on Card _____

Card # _____ Exp. date _____

___Bill me. Add $2.50 for handling

Check one: ___ Renewal ___ New Membership

Name _____

Address _____

_____

State _____ Zip/Postal code _____ Country_____

Phone_____ ICCE Membership #__ __ __ __ __ __

Membership # *must* be included to utilize ICCE member rates.

LOGO EXCHANGE

Volume 7 Number 4 Journal of the ICCE Special Interest Group for Logo-Using Educators February 1988

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

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

Contents

From the Editor — Desktop Publishing for Kids?
Sharon Burrowes Yoder 2

Monthly Musings — I ♥ Logo
Tom Lough 3

Logo Hearts: A Project for Young and Old
Judith Enz Clauss 5

Little Kids and Logo — Turtles in Space: An Out of this World Microworld
Leslie Thyberg 8

Logo Ideas — Search and Replace: More on Text
Eadie Adamson 10

LogoLinx — Parallel Inconsistencies
Judi Harris 12

Logo PLUS: A New Version of Logo Worth Looking At
by Eadie Adamson 14

Logo Connections — StarNet: A Fourth-Generation Computing Environment
Glen Bull and Gina Bull 16

MathWorlds — Easy as 1 1 2 2 3
by Michael Tempel, edited by Sandy Dawson 19

Assessing Logo Learning in Classrooms — V. Strategies for Learning Through Exploration
Dan Watt 23

A LogoWriter Tip
by Jandy Bird 27

Terrapin Logo on the GS
from the *Terrapin Times* 27

Search and Research -- The Nature of the Problem
Douglas H. Clements 28

Global News
Dennis Harper 30

ICCE Membership (includes *The Computing Teacher*)

	U.S.	Non-U.S.
	28.50	31.50

SIGLogo Membership (includes *The Logo Exchange*)

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

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

From the Editor

Desktop Publishing for Kids?

It seems as though I am always in front of a keyboard. For the past year, an ever increasing percentage of my time has been spent in some phase of the writing process from jotting down notes to making outlines and lists, to composing text to editing. This intimate day-to-day involvement with my Macintosh has caused me to reflect both on the writing process and the convoluted pathway that led me to this keyboard centered life.

I can remember, as early as grade school, having an intense dislike for writing. I was constantly criticized (justifiably) for my handwriting. I found using a pencil tedious at best. When I was in junior high, my mother who was for many years an executive secretary, insisted that I learn to type correctly. I'm sure that I have never properly thanked her for instilling good keyboard habits in me when I was young. While being able to type papers at home was helpful, it didn't solve my in-school writing problems. The only writing assignments that I remember enjoying in high school were part of a unit on creative writing that required only one or two short paragraphs. I recall being excited by what I had written, in fair measure because I could focus on content and was not required to think about the mechanics. That was, perhaps, the first inkling that I could really enjoy writing.

In college, I avoided courses that involved writing as much as I possibly could, a difficult task at the strong liberal arts college that I attended. Even though I could type quite well by then, the process of erasing and correcting my final papers was still tedious and unpleasant. I suspect that a significant factor in my decision to major in math was that it gave me a way to avoid a lot of writing.

Once I finished my master's degree, I found that writing was a skill that was not in much demand either as a math teacher or as a mother. My old nemesis did not haunt me for some years to come. However, when my son David entered school, I quickly discovered that he seemed to be having the same difficultieswith writing that I did. When he reached junior high, I vividly remember his struggling over the Animal Report that was required of every seventh grader. He enjoyed the research and reading for the report, but he agonized over the physical process of transferring his ideas to paper using a pencil or typewriter. As I empathized with him one winter evening, I had one of those powerful "ah ha" reactions. In the next room was our relatively new Apple II Plus. "Wow!" I said to myself, "Add a printer and a piece of software and we have a word processing system." Little did I know that this addition to our household would change not only my son's life, but mine.

Once freed from pens and typewriter correction fluid, both David and I quickly discovered the joy of writing. By the time he was a senior in high school, David was publishing in the school literary journal, had received school-wide writing awards, and was contemplating a career as a writer. And then there's me — well you know about that. I can't seem to stop writing. The words flow off the ends of my fingers in an ongoing stream. Not only has the volume of my writing increased, but the process that I use to write has also changed.

But all of this is history — albeit important for those of us who have struggled with the mechanics of putting words on paper at various stages in our lives.

More recently technology has provided me with yet another tool to enhance my writing. In the process of developing some materials for junior high students for LCSI last year, I learned how to use a Macintosh with a word processor and graphics programs, all instantly accessible with the press of a key. (See Logo Connections in this month's *LX*) As I am writing, I can describe a Logo screen, switch to a Macintosh Logo to create a turtle design, transfer that design to a graphics program to add arrows and explanation and finally copy the finished product into my word processor. As I worked within this environment, I became more and more dependent on it, finding that I wanted graphics capabilities at my fingertips at all times. In the process, I noticed that my writing was changing. I was now thinking in terms of both images and words. No longer did I have to explain everything; instead I could simply draw a picture.

Shortly after moving to Oregon last summer, I learned to use PageMaker to do the layout for LX. At first, it was just a tool that I used to put together the journal that you are reading. But as I became more proficient, I found that this powerful desktop publishing software gave me the flexibility to do almost anything I wanted. I could put together all manner of text and graphics to suit my whim — and then produce a finished Laser printed copy without glue, tape, and whiteout.

And what does all of this powerful software have to do with a "lowly" Logo program? Recall that I said my writing is changing as I increasingly make everyday use of graphics. LogoWriter, and more recently Logo PLUS, provide a similar ability to mix text and graphics in a very flexible manner. As I reflected, I thought of my high school students who hated to write as I once had. These students would enter my Logo class and immediately focus on using the graphics capabilities of Logo, producing elaborate designs or complex animations. As the term progressed, I would often notice that these same students began to spin language around their graphics images. Increasingly their programs contained text that enhanced their

graphics displays. The graphics seemed to be inspiring the writing. In fact, students in a high school English class that learned to use LogoWriter to illustrate poetry became involved in the writing process far beyond the poetry unit that we presented. (Burrowes and Bowman, 1988)

In schools today, the focus is on computer applications. Students learn word processing and spreadsheets, data bases and graphics packages. They learn to use the computer as a tool. There is less and less emphasis on programming and Logo is becoming less and less in vogue. Are those who see Logo only as a programming language missing an important facet of today's versions of Logo? Can Logo now be viewed as an extensible productivity tool? Logo's flexibility in producing a mixture of text and graphics is not at all unlike the flexibility of a package like PageMaker. In fact PageMaker feels very Logo-like to me. Should we stop thinking of Logo as a programming language at all, especially in the elementary school, and start thinking of it as a powerful and extensible tool that can be used in all subject areas?

Using Logo as a productivity tool certainly does not diminish its power to teach problem solving; the problems to be solved are different than those traditionally associated with Logo. Further, a site license for Logo is much less expensive than buying a copy of AppleWorks and Print Shop for each machine. Perhaps we should be using Logo to help improve the writing of students, providing them with a glimpse of the future when small personal computers will have the power to produce integrated sound, graphics and text.

Reference

Burrowes, Sharon and Bowman, Connie (1988). Logo-Writing: Illustrating Poetry. *The Writing Notebook., 1(2) 19-22.*

In addition to editing the *Logo Exchange*, Sharon Burrowes Yoder teaches computer education courses at the University of Oregon. In her leisure time, she reads, writes, sews, does hand work, walks on the beach and plays the guitar. She can be reached at

ICCE
1787 Agate Street
Eugene, Oregon 97403
CIS: 73007,1645
BitNet: ICCE@Oregon

Monthly Musings

I ♥ Logo

by Tom Lough

I remember a piece by Andy Rooney on CBS' 60 Minutes in which he made snide comments about bumper stickers and signs which proclaim that the owner "hearts" something. Nobody "hearts" anything. How can you "heart" New York, for example?

With all due respect to Mr. Rooney, I happen to like the idea that we can "heart" something. What better month than February to think about it? And what better thing to "heart" than Logo?

In the September column, I asked for readers to send in their responses to the question, "What is your favorite Logo command and why?" I'd like to thank especially Sister Mary Grace of the Holy Trinity School in Pittsburgh for sharing the responses of her students. Here, then, are some examples from all over showing what we "heart" about Logo.

Jill Miller, Grade 6, Holy Trinity School, Pittsburgh, PA: FORWARD is my favorite Logo command because that is what I want to do in life. When I go FORWARD, I learn more and more. If I go FORWARD in my daily life, I do better and better.

Gwen Varsamis, Greensboro, NC: I like FORWARD. Students of all ages understand the essence of Logo, a FORWARD moving language.

Marianna Sparks, Lubbock, TX: I love FORWARD. It is very basic [oops, I mean, FUNdamental!] and simple. I like to go FORWARD rather than BACK.

Student, Holy Trinity School, Pittsburgh, PA: FORWARD is my choice Logo command. It is one of the first I learned with which to enjoy creating geometric shapes. FORWARD makes me feel I keep on going, not only in computer work, but also in whatever I undertake to accomplish.

Sharon Yoder, Eugene, OR: My favorite command is FORWARD. I like wrapping and its implications.

Matthew Kramer, Grade 5, Holy Trinity School, Pittsburgh, PA: My favorite Logo command is BACK because I can move the turtle backwards and erase things I do not want. I do this often to improve my program.

Monthly Musings -- continued

James Barlow, Grade 5, Holy Trinity School, Pittsburgh, PA: My favorite Logo command is LEFT and its opposite RIGHT. They remind me of a set of twins — one going LEFT and the other going RIGHT. With LEFT and RIGHT, I am able to create many interesting geometric figures. LEFT and RIGHT have helped me in my understanding math problems that deal with angles.

Steve Burger, Grade 5, Holy Trinity School, Pitsburgh, PA: I like RIGHT. It is the best so far. I feel this is an important command for making angles and all kinds of geometric figures. I know many commands, but RIGHT is certainly my favorite.

Adrienne Zemanek, Grade 5, Holy Trinity School, Pittsburgh, PA: Of all my favorite commands, REPEAT is my most favorite. I like it because it is in most programs. It makes interesting geometric shapes. That's why I love REPEAT.

Anne Baker, Grade 8, Holy Trinity School, Pittsburgh, PA: My favorite Logo command is REPEAT. I wish it had an abbreviation. REPEAT is a different kind of a command because you do not have to do each line over and over again. This helps me save time and I am able to make various designs — at times it is not as I planned! I do enjoy surprises.

Neil Prakash, Grade 6, Holy Trinity School, Pittsburgh, PA: My favorite Logo command is REPEAT. I could make long programs with one Logo sentence using REPEAT. I like science, and in science you repeat everything several times before it is perfect. This is why I like this command.

Jim Catanzarite, Grade 6, Holy Trinity School, Pittsburgh, PA: My favorite Logo command is RREEPPEEAATT. I like the command REPEAT because it saves time and work. In life you repeat many things because you do not always understand things the way you should.

Lori Ludwick, Grade 8, Holy Trinity School, Pittsburgh, PA: My favorite Logo command is REPEAT because it saves time and effort from typing procedures over and over again when I am working at the computer. In Logo I like to hurry and rush to see what my result is. When I have to type things out, it takes longer. Just by using REPEAT I am able to create faster and better. Do you agree?

Jason Slepecki, Grade 6, Holy Trinity School, Pittsburgh, PA: My favorite Logo command is REPEAT with the idea of nesting. I can create many beautiful designs. REPEATs are useful in many programs that I write. It is a form of "try-outs." When I nest using REPEATs, I am amazed at my performance!

Molly Watt, Antrim, NH: My favorite command is REPEAT. I like to do the same thing over and over. REPEAT was not in the original Logo at MIT, but was added later in response to the expressed needs and desires of students and teachers. I like that developmental aspect of the command.

Sheila Cory, Chapel Hill, NC: REPEAT is such a voyage of discovery. It opens up a million doors, even if they go no further, it opens more doors than any other command.

But these are not the only favorite commands! I'll save the "hearts" of additional Logo fans for another column. Meanwhile, let's all take a moment to muse. Why do we "heart" Logo so much? For me, it is because Logo never gets boring. I always find something refreshingly new and surprising each time I work with it. And I keep coming back for more!

What's my favorite Logo command? Can you guess?

♥ FD 100!

Tom Lough
Box 394
Simsbury, CT 06070

Logo Hearts: A Project for Young and Old

by Judith Enz Clauss

Develop a Logo heart project and everyone from second graders to adults will become involved with it, even when it isn't Valentine's Day. This project involves providing students with a basic heart procedure that will allow them to draw different sized hearts. The heart idea was stumbled upon accidentally last February when a heterogeneous class of second graders was coming to the Western Carolina University campus once a week to participate in a pilot research project using Logo. Snow has caused the class to miss a couple of weeks in January and when they came again the week before Valentine's day, they informed us that next week they were going to make Valentines with Logo. I created a procedure that would allow them to use Logo to draw hearts of different sizes. (See the end of this article for both Apple and IBM listings.)

We expected students to make a Valentine and move on to other projects. This was not the case. They made heart designs and heart creatures. (See Figures 1, 2, and 3.) One girl began with a heart of size 25 and incremented by one each time until she reached 43 and the heart want off the screen. She printed her design and soon several other students wanted her to tell them how she did it. Many replicated her design, but a number of the students explored variations.

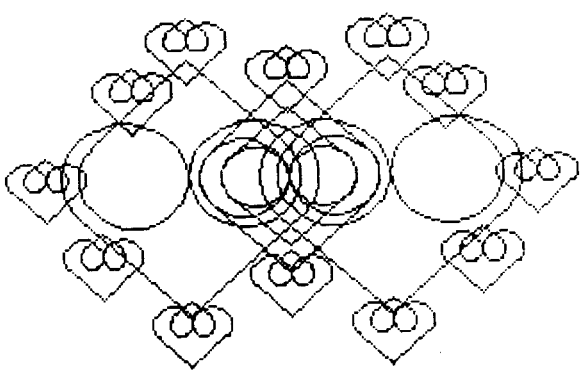


Figure 1

In the heart procedure as written, the turtle begins and ends in the top center of the heart, but its orientation is reversed. In order to make a series of hearts all going in the same direction, the turtle must be turned between runs of the heart procedure. This orientation difficulty led both to discoveries and requests for help. Numerous angle discussions were held both with and without assistance of we who were teachers. It was interesting to note that the second graders preferred using the setheading (**seth**) command rather than **left** and **right** in order to turn the turtle. They each had wheels that had left and right on one side and a compass showing every 45 degrees on the other. They simply used the wheels as a tool to decide which heading they wanted.

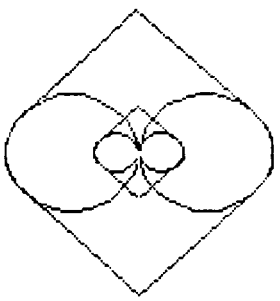


Figure 2

Figure 3

When using this same project in the summer with in-service elementary through high school teachers many similar issues were raised, accompanied by a few new ones. One teacher wanted the heart procedure to start at the base of the heart such that four hearts would fit together perfectly to make a flower-like design (see Figure 4.)

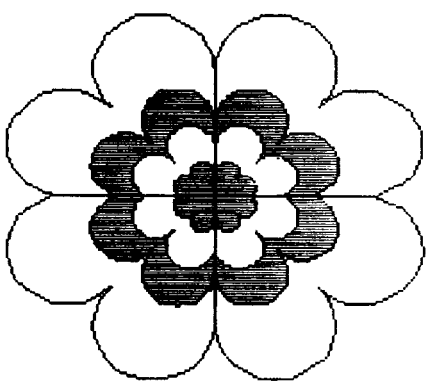


Figure 4

I suggested that she simply rewrite the heart procedure that had been provided. Her attempts and initial bugs caught the attention of the rest of the class and her problem quickly became a class project for the morning. A great deal of discussion and trial and error went on before they came up with a procedure that worked as they envisioned it. (See HEART.B at the end of the article.) Although the teachers all taught math or science, they had never dealt with angles and orientation from a turtle geometry perspective. Some of the topics discussed included:

A Logo Hearts Project -- continued

•orientation of the turtle;
•**seth** versus **left** and **right**
•symmetry: bilateral and point;
•arcs: what are they really and why doesn't 180 degrees work for the heart?

Suggestions for further explorations are included on the Copy Me page at the end of this article.

Using a heart project opens an interesting and rich area for investigation. The suggestions in this article should just provide the springboard. For example, with LogoWriter (as well as some other versions of Logo,) you can also investigate what happens when the procedures are run using multiple turtles. You and your students will undoubtedly find new directions to explore and enjoy.

Procedures

The basic heart procedure was created utilizing LogoWriter's arc procedures from the TURTOOLS page along with some unorthodox trial and error to create symmetrical hearts on a computer screen.

IBM version:

```
TO HEART :SIZE
ARCR :SIZE 225
FORWARD :SIZE * 2.432
RIGHT 90
FORWARD :SIZE * 2.432
ARCR :SIZE 225
end
```

Apple version:

```
TO HEART :SIZE
ARCR :SIZE
FORWARD :SIZE * 2.33
RIGHT 90
FORWARD :SIZE * 2.33
ARCR :SIZE 225
END
```

The arc tool from LogoWriter's TURTOOLS page is:

```
TO ARCR :RADIUS :DEGREES
MAKE "STEPS (2 * :RADIUS *
   3.1416 / 36)
MAKE "REM REMAINDER :DEGREES 10
REPEAT :DEGREES / 10 [RIGHT 5
   FORWARD :STEPS RIGHT 5]
IF :REM > 0 [FORWARD :STEPS *
   :REM / 10 RIGHT :REM]
END
```

HEART.B Procedures

IBM version

```
TO HEART.B :SIZE
FORWARD :SIZE * 2.432
ARCR :SIZE 225
RIGHT 180
ARCR :SIZE 225
FORWARD :SIZE * 2.432
END
```

Apple version

```
TO HEART.B :SIZE
FORWARD :SIZE * 2.33
ARCR :SIZE 225
RIGHT 180
ARCR :SIZE 225
FORWARD :SIZE * 2.33
END
```

Judith Enz Clauss
Western Carolina University
Department of Elementary Education and Reading
School of Education and Psychology
P. O. Box 2528
Culowhee, North Carolina 28723
704-293-3875 or 704-227-7108

About the Cover

The heart designs on this month's cover are all a part of Judith Enz Clauss' article describing a Logo Hearts Project (above.) The flower-like design was produced using HEART.B. The other three designs were produced using HEART.1, HEART.2 and HEART.3 from the Copy Me worksheet on page 7.

Heart Worksheet

1. What happens when heart is used several times without changing the turtle's orientation?

2. Create a procedure that spins the heart.

3. Create a procedure that rotates and increments or decrements the heart each time.

4. Create a design that has point symmetry with the heart procedure.

5. Create a design using more than one heart that has bilateral symmetry.

6. Create a procedure that draws a series of hearts with the same orientation, but getting smaller each time.

7. Create your own special heart creature.

8. Create an abstract design with hearts.

9. Create a heart world or a heart neighborhood.

10. Create a heart flower.

11. Predict what each of the following procedures does. Then run each procedure to see if you were correct.

```
TO HEART.1 :SIZE :HEADING
IF :SIZE > 36 [STOP]
SETH :HEADING
HEART :SIZE
SETH :HEADING + 2
HEART.1 :SIZE + 2 :HEADING - 2
END
```

```
TO HEART.2 :SIZE
IF :SIZE > 36 [STOP]
HEART :SIZE
HEART.2 :SIZE + 1
END

TO HEART.3 :SIZE :HEADING
IF :SIZE > 36 [STOP]
SETH :HEADING
HEART :SIZE
SETH :HEADING - 3
HEART.3 :SIZE + 3 :HEADING - 3
END
```

An LX Copy Me Page

Little Kids and Logo

Turtles in Space: An Out of this World Microworld

by Leslie Thyberg

I was recently invited to conduct a series of workshops for a large primary school. It was a large, stimulating school, rich in exploratory and experientially based learning experiences for the students. To my surprise, however, a large number of the faculty were quite skeptical about the benefits of Logo for their pupils. "What is the use of Logo in primary and pre-school instruction?", they challenged me.

Uses of Logo in Primary and Pre-school

Logo, can help students develop in a number of ways.

Using Logo...
- Stimulates young learners:
  - is highly motivational
  - requires interaction
  - engages and involves attentive behaviors
  - provides a sense of enjoyment
- Accommodates young learners:
  - adaptable to different skill levels
  - adaptable to different attention spans
  - adaptable to different learning rates
- Facilitates learning:
  - enables self-expression without demanding sophisticated motor skills
  - provides a variety of experiences with graphics and language
  - offers opportunities for learning cooperation, collaboration, and other social skills
- Informs young learners:
  - gives individual and immediate feedback
  - uses images, text, and sound
- Gives the learner control:
  - offers vast array of potential microworlds for exploration
  - provides a means of self-expression
  - allows the manipulation of symbols
- Enhances appropriate developmental learning skills:
  - visual - e.g. skills that involve seeing, remembering, and discriminating
  - expressive language (spoken) - e.g. skills that involve speech, language and vocabulary (such as naming pictures, discussing or giving information, giving directions or descriptions)
  - expressive language (written) - e.g. skills that involve writing to convey meaning, proofreading and debugging, using print and programming conventions)
  - receptive language (listening) - e.g. skills that involve hearing and remembering (e.g. following oral directions)
  - cognitive - e.g. math skills that involve numbers, uses of numeral, their meaning, rules, and relationships
  - social - e.g. skills that involve interacting positively with others (taking turns, help seeking, etc.) cultural literacy - skills that involve learning to interact within a learning environment

Keeping those principles in mind, I asked them to think about how using Logo with primary students could enhance content area learning. I happened to be visiting the school shortly after the recent successful shuttle space mission. I noticed they had some Muppet keyboard software, which included a game called Pigs in Space. I asked, "What if we replaced Miss Piggy and her astronaut friend with Turtles in Space? Using the IDEAL Problem Solving Model (see my October, 1988 *LX* article) I suggested that I would lead students through a whole range of activities revolving around the theme of Turtles in Space. The following ideas are just a capsule of the universe of activities available.

For an initial activity, read some children's literature motivators to the class such as *Stowaway to the Mushroom Planet* or Ezra Jack Keats' *Give my regards to the man in the moon*. Then have the students do some brainstorming. In IDEAL terminology, this is the phase for IDENTIFYING the problem. For example, you might have your students list all the objects they think they would need for a year long trip in space. Then you could have them prioritize what they have listed by putting a star next to the ten most important items. To further focus the problem (DEFINE), I would then have students select five of the ten starred items for the following activity.

Have the students do one or more of the following: write a paragraph to describe each object; draw each object on a separate notecard, drawing paper or graph paper; or have them use such magazines as Smithsonian or National Geographic to cut out pictures to make a collage. Having engaged in these pre-computer activities, I would then have them use their designated computer time to generate Logo graphics to go along with what they had been doing as a creative thinking stimulus. Having the students plan their drawings beforehand and labelling each of the components provides a good opportunity for teaching about outlining and planning. You may, however, have some students with a different thinking style. If they tend to be a little "spacey" and want to make it up as they go, let them! An alternative or additional activity would be to have the students list all the words they can think of to

describe an alien. If you are fortunate enough to have LogoWriter, merging the writing and the graphics gives this activity an extra boost. Have your students draw a comic strip featuring Alf or an E.T. like character; or encourage them to write a space adventure story with illustrations; and/or create their own episode for Star Trek.

To take them further along the creative thinking path, I would propose to them some alternative scenarios to write, draw, and react to. For example ask students to respond to this: "Suppose you have just crashed onto a strange planet. List all the words to describe the setting (emphasizing descriptors for all five sense - sights, smells, sounds, and tactile sensations.) Do at least one of the following activities about your Crash Landing - write a news report; create a skit and act it out; make a clay model of the planet; generate a Logo graphic to go with your writing.

Or, suggest to the students that they could be like the Star Trek crew. How would they begin to explore the planet on which they have crashed? Ask them to list all the possible situations in which they might find themselves. Again, generating a story or play with a computerized illustration is a great activity. Scenery and props can be plotted out with Logo or as an auxiliary activity.

To synthesize some of the learning that has been taking place, ask the students to create their own original products for the theme Turtle in Space. Begin by listing all the possible tie-ins for the theme. Have the students choose their best ideas and draw, make, or describe what their products will be.

An alternative would be to suggest that they have each been chosen to design a rocket for use in the Turtle Space program. Have them make a sketch and label each component. This is an excellent way of guiding your students through generating a top-down plan for a Logo project. Concurrent with this I also had my students build a model (out of Legos, or any other building material available) and make a scale drawing (with a legitimate graph paper scale or one of their own devising) .

To take them still further in creative thinking they can Elaborate on being selected to live aboard a Space Station for one year. List all the things they would need beyond daily necessities. For a writing activity have them record in individual or class diaries (in the style of Captain Kirk!) about their week aboard the Space Station. Use Logo to draw a label a model for the space station. Your more advanced students can utilize Turtle Coordinates (SETXY or SETPOS).

One final idea. Create a utility program, similar to Dan Watt's game of SHOOT in his book, *Learning with Logo*. I have an excellent one, written by a nine-year old and use it for imaginary galactic adventures. Its young author called it Turtle Target. I have used Turtle Target for everything from an introductory means for gaining control over the turtle (estimating angles and distance) to part of a unit on the solar system where the target is an asteroid or the Death Star. If you would like a copy, please feel free to send me a self-addressed, stamped envelope, and I will enclose a handout of the procedures for you and your students to enjoy.

Bibliography:

Watt, Daniel. (1983). *Learning with Logo*. New York: Byte Books/McGraw-Hill.

Dr. Leslie F. Thyberg
Chatham College
Woodland Road
Pittsburgh, PA 15206
Applelink: ALS 038

Leslie is a former classroom teacher. Ten of her thirteen years of experience were with children in grades three and under. She is currently an assistant professor of education at Chatham College and does frequent workshops and presentations on Logo and writing. She is engaged in first-hand learning through the starry eyes of her three-year old.

Logo Ideas

Search and Replace: More on Text
by Eadie Adamson

Last month's Logo Ideas looked at writing adventure stories, getting the text on the screen and programming changes of pages. Although the focus was on text, as you apply those ideas in your classroom, don't neglect the possibilities of adding graphics to the adventures in the form of drawings, stamped shapes and even animation. One of my students did a nice example moving a car across a page and onto the next page. Try it!

This month we'll look at some of the other word processing primitives in LogoWriter: REPLACE, SEARCH, CUT, INSERT, SOL, CU.

Fixing a spelling error: Using REPLACE

Young children love to write stories but often need to correct spelling. When my students work on writing with LogoWriter, I give them small index cards. When they need the spelling of a word, they give me their card and I print the word they need on the card.

Later, when their writing is done, I look over their spelling. I give them another card with *correct* spellings of words and let them look for the incorrect word. Then I have them use a procedure named FIX which corrects the words one at a time.

FIX uses the word processing primitive, REPLACE, which takes two words as inputs. The first input is the word to be replaced and the second is the the word to replace it with. From the command center one can type:

```
REPLACE "si "is
```

and a misspelled "si" is replaced with "is," the correct spelling. Since quotation marks are sometimes difficult for little fingers to manage, I wrote a FIX procedure which replaces the misspelling with the correction. FIX first asks for the word to be replaced, collects that word as the value for OUT. Then it asks for the word to be inserted, collecting the second word as the value for IN. Once both inputs are collected, REPLACE goes to work. The students merely type FIX and answer the questions. FIX looks like this:

```
TO FIX
TYPE [FIX WHAT WORD?]
TYPE CHAR 13
NAME FIRST READLISTCC "OUT
TYPE [WITH WHAT WORD?]
TYPE CHAR 13
NAME FIRST READLISTCC "IN
TYPE CHAR 13
REPLACE :OUT :IN
END
```

The students love to correct spelling this way. It accomplishes the task faster and easier for most than moving up to the text, deleting the word accurately and typing the correction.

You might also write tool procedures to replace frequently misspelled words. For example, if you habitually reverse letters in some words, write yourself a correcting procedure or two for those words.

REPLACE actually does three things: it SEARCHes for a word, CUTs it and INSERTs another word in its place. You might even try writing your own REPLACE using SEARCH, CUT and INSERT. SEARCH finds and highlights a word, CUT deletes selected text, INSERT adds a word or list of words without a carriage return.

SEARCH and a Logo Sing Along

If you have been playing with music, here's how to make a dancing cursor highlight the words for the song, making a Logo Sing Along. You will need to use INSERT to put text on the screen without a carriage return, SOL (which stands for "start of line") to move to the beginning of the inserted line of text, and SEARCH to find the words. You'll need the music notes as well, of course.

Here are the steps to follow:

1. INSERT a line of text (INSERT leaves the cursor at the end of the line; PRINT moves the cursor down to the next line).
2. Move the cursor back to the beginning of the line: SOL
3. SEARCH for the first word.
4. Play the first note.

   For example, to add a new dimension to the "Row, Row, Row Your Boat" music video project:

   ```
   SEARCH "ROW
   C 20
   ```

5. Repeat steps 3 and 4: SEARCH for the next word, play the second note, and so on. Repeat this process for the next line of the song.

6. Break each line into a procedure of its own to make debugging easier.

7. Write a superprocedure that calls each line of the song in turn.

As a last task, you might add animation where it's appropriate. The PULL and GLIDE procedures below make the man in the boat move. (I changed the shape so that the man and the boat are one shape rather than adding another shape for the boat.)

Now, conduct a sing-along!

Here's the beginning of a sing along for "Row, Row, Row Your Boat". Use it as a guide for your own version:

```
TO ROW
CT
INSERT [ROW, ROW, ROW YOUR BOAT]
SOL
REPEAT 2 [SEARCH "ROW C 20]
SEARCH "ROW
C 15
SEARCH "YOUR
D 5
SEARCH "BOAT
E 30
INSERT CHAR 13
GLIDE
END

TO GENTLY
INSERT [GENTLY DOWN THE STREAM...]
SOL
SEARCH "GEN
E 10
SEARCH "TLY
D 5
SEARCH "DOWN
E 10
SEARCH "THE
F 5
SEARCH "STREAM
G 20
PULL
REPEAT 3 [CF]
INSERT CHAR 13
GLIDE
END
```

(REPEAT 3 [CF] moves the cursor over the three dots following the word STREAM. INSERT CHAR 13 moves down to the next line.)

```
TO ROWBOAT
SETUP
ROW
GENTLY
END

TO SETUP
PU
SETH 90
SETSH 18
END

TO PULL
SETSH 18
END

TO GLIDE
SETSH 19
REPEAT 25 [FORWARD 2]
END
```

Notice the INSERT CHAR 13 at the end of ROW and GENTLY procedures. This command moves the cursor off the highlighted word while the animation is in progress. Now, finish the song!

Some songs may not be suitable for animation, but will work well with simply highlighting the words. Try this with "Frere Jacques", which is ideal for moving by syllables. A group of students could each work on a different tune and then put them together into a Logo Sing Along for another class to enjoy. If your students have already been programming music, this finishes a music project very nicely.

There are a number of different ways to work with text in a project such as this. Instead of using INSERT for the text, we could have used PRINT and then CU, for "cursor up". Yet another approach is to PRINT all the text on the page, move to the top of the page with the command TOP, and then SEARCH for each word in turn.

The same technique can be used very effectively with poetry, too. PRINT the text on the screen. Move to the beginning of the poem with the command TOP. SEARCH for the first word. Use WAIT and a number to vary the speed. This is a good way to study the rhythm of the spoken word.

More on text next month.....

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

Logo LinX

Parallel Inconsistencies
by Judi Harris

Does the notion of **Logo worksheets** seem slightly incongruous to you? Does **simple recursion** escape you? Somewhat self-contradictory, perhaps, like a **lasting fad** or the **only choice**? If so, you have discovered *oxymorons*. These "contradictory expressions that make absolute sense" (Blumenfeld, 1986) can stimulate eager word study and **serious humor** in the Logo-infused classroom environment.

Consider, for example, the **old news** that was **taped live**, **clean dirt**, **dry ice**, and **freezer burn**. **"Same difference,"** you say? **"Almost exactly""** I'd reply. Dr. Warren S. Blumfeld, Ph.D., recently described these phascinating phrases as "two concepts (usually two words) that do *not* go together, but are used together...the bringing together of contradictory expressions." (Blumfeld, 1986, p. 11) Students' work with oxymorons directs attention to synonyms, antonyms, new vocabulary; even geography.

Planned Serendipity

Logo can be used to **temporarily cement** oxymoronic phrases. Using the familiar PICK tool,

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

lists of related nouns and adjectives that output **randomly chosen** words to concatenate into oxymorons. For example,

```
TO LARGE.ADJECTIVE
OUTPUT PICK [TREMENDOUS GARGANTUAN
   GIGANTIC JUMBO HERCULEAN ENORMOUS
   IMMENSE GIANT COLOSSAL MIGHTY]
END

TO SMALL.NOUN
OUTPUT PICK [SHRIMP FLEA LILIPUTIAN
   DWARF PIGMY MIDGET GNAT MICROBE
   RUNT MOUSE ATOM]
END
```

can be concatenated with:

```
PRINT SENTENCE
   LARGE.ADJECTIVE SMALL.NOUN
```

and that's how **jumbo shrimp** are born.

Constant variation can be achieved with the addition of an iterative command,

```
REPEAT 3 [PRINT SENTENCE
   LARGE.ADJECTIVE SMALL.NOUN]
```

to which the computer may respond:

```
GIANT MICROBE
TREMENDOUS PIGMY
COLOSSAL GNAT
```

As you can now see, "Mighty Mouse" also owes a great debt to oxymoronic combinations.

Ideological Manifestations

Randomly planned oxymorons can be quite philosophical. Descriptors such as these:

```
TO DESCRIPTOR
OUTPUT PICK [ALMOST MOSTLY
   TENTATIVELY [JUST ABOUT] [SORT OF]
   SOMEWHAT PARTIALLY TEMPORARILY]
END
```

when concatenated with certain thoughts:

```
TO THOUGHT
OUTPUT PICK [PERFECT CERTAIN EXACT
   COMPLETE ANONYMOUS COMMITTED
   ORGANIZED PERMANENT EVERYBODY]
END
```

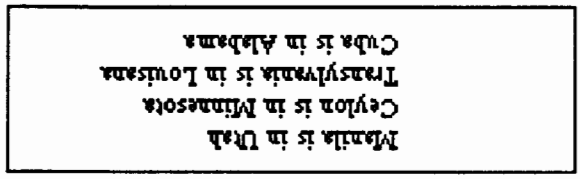
can yield some **serious comedy**, all too often spoken in earnest.

```
REPEAT 6 [PRINT
   SENTENCE DESCRIPTOR THOUGHT]

TEMPORARILY PERMANENT
JUST ABOUT COMPLETE
MOSTLY ANONYMOUS
TENTATIVELY COMMITTED
ALMOST CERTAIN
ALMOST PERFECT
```

Geographic Oxymorons

Believe it or not, I lived in **Holland, Pennsylvania** for seven years. I suppose that must be near **Norway, South Carolina**. Where can each of these **country cities** can be found on a United States map?

```
TO COUNTRY.CITY
OUTPUT PICK [TRANSYLVANIA, HOLLAND,
   CUBA, MANILA, CEYLON, NORWAY,]
END

TO STATE
OUTPUT PICK [PENNSYLVANIA UTAH
   [SOUTH CAROLINA] MINNESOTA
   LOUISIANA ALABAMA]
END
```



There are many seemingly misplaced cities that students can locate and record.

Philadelphia, Mississippi
Miami, Missouri
Atlanta, Texas
Atlanta, Indiana
Paris, Texas
Orlando, Oklahoma
Rome, Georgia
Rome, New York
East St. Louis, Illinois
Missouri City, Texas
Denver, Pennsylvania
St. Paul, North Carolina
Cincinnati, Iowa
Minneapolis, Kansas

Your pupils may enjoy adding to this unusual list.

Contrary to popular opinion, **city-states** were not confined to the Roman Empire. America has quite a few.

Wyoming, Ohio
Oregon, Missouri
Oregon, Wisconsin
Florida, New York
East Texas, Pennsylvania
Kansas, Alabama
Indiana, Pennsylvania
Carolina, Rhode Island
New England, North Dakota

Initial Conclusion

Schools are already full of oxymorons. The **Teacher Lounge** is usually anything but. Has your administrator recently asked you for an **original copy**? And, just what is a **student teacher**? Why not continue a **new tradition** and suggest Logo-fueled oxymoron study to help brighten these winter months?

I would be remiss, though, if I did not caution you. "In this world of **constant change, near misses,** and **routine emergencies,** there is comfort in the **subtly obvious** art of the oxymoron. But be warned...the...oxymoron is addictive."(Blumenfeld, 1986, back cover)

Reference

Blumenfeld, W. S. (1986). *Jumbo shrimp & other almost perfect oxymorons.* New York: Perigee Books.

Judi Harris taught students in Philadelphia-area elementary through graduate schools to use computer in teaching and learning for six years. She now does similar work at the University of Virginia, where she is completing her doctoral work in Instructional Technology.
She can be reached at

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

Logo PLUS: A Review

Logo PLUS: A New Version of Logo Worth Looking At

by Eadie Adamson

Terrapin's new Logo PLUS for 128K Apple computers, which became available last fall, is full of new features which any Logo user will appreciate. Logo PLUS is definitely not just another version of Logo, although some of its innovations may provoke arguments about what Logo really is or should be.

First and foremost, Logo PLUS is a ProDOS based version of Logo. Earlier versions of Terrapin were based on Apple's older DOS 3.3 operating system. This switch to ProDOS makes Logo PLUS files compatible with most current word processors and makes it possible to send procedure files via telecommunications. For those who are reluctant to deal with ProDOS, Logo PLUS can easily be used as if those features were not there.

Logo PLUS is a very thoughtful extension of the original Terrapin Logo. For instance, an accessories file on the language disk contains a disk formatter and a font editor as well as a file conversion option. There is no need to load another program to perform these tasks. Since many users will want to convert their old Terrapin DOS 3.3 files to ProDOS, placing the file conversion utility on the language disk just makes sense. Using the font editor, one can edit, modify, create and save fonts if the fonts contained on the language disk or the utility disk do not fill your needs. An option on the accessories menu allows you to return directly to Logo PLUS or to another application.

Several interesting additions make Logo PLUS a Logo worth considering: a variety of text styles available for use on the graphics screen, a shape editor which allows creation of new shapes for the turtle, a PRINTSCREEN command allowing full-size or small size pictures, color printing in two sizes, and many new primitives including color detection, a FILL command, a NOTE primitive for making music, DOT and SDOT, and much more.

The Terrapin turtle is still the traditional triangle shape, but a shape editor now makes it possible to redesign the turtle so that it has a real turtle shape or any other shape you might want. Each created shape will rotate just as the turtle does, but it is possible to keep a shape oriented in a particular direction with the LOCKHEADING primitive. UNLOCKHEADING or DRAW will allow the turtle, in whatever shape, to rotate again.

The shape editor itself is likely to feel familiar to those who have used the shape editor in programs such as Print Shop or Pixit. Again, this has been developed with the user in mind. For example, it is easy to copy a shape and then, with a simple command, flip it horizontally or vertically. This means that the shapes for a complex series of animations involving reversing directions can be made quickly and easily. A special ZOOM command allows full-screen editing as well. ZOOM can even be used to create a shape out of an image on the screen. The size of the shapes will be limited by the amount of space in the area of memory which is set aside just for shapes. There's even a command, SHAPEROOM, which can tell you how much shape space is left, since the space for shapes is independent of the Logo workspace.

There's even more that you can do with shapes. SETSIZE lets you alter turtle size; STAMP will leave a copy of the turtle shape on the screen. The turtle, in whatever shape, can be set to one color while it draws in another. TCOLOR will report your current turtle color. Files of shapes that you create can be stored on your data disk. The utilities disk contains some sample shape files to get you started.

As I tried out Logo PLUS I found the accompanying 136-page booklet to be amazingly thorough. Clear instructions for using each new feature are often accompanied with neat project ideas. A sample project for animation takes the user step-by-step through the process of creating a series of shapes and then gives ideas and procedures for using them, providing a few challenges as well.

Text can now be entered on the graphics screen and even used interactively with new primitives such as GREQUEST (GRQ) and GREADCHARACTER (GRC). I particularly like the timer version of these two primitives which lets you determine how long to wait for a response from a user. All of the various text styles can be accessed from procedures as well as with the special keystrokes used in the immediate mode.

In white on black, text on the graphics screen is dazzling. The text style can be either plain, bold or italic, each with or without underlining. White text on a black background is extremely successful as is text display in inverse mode. Since one can use graphics text from procedures with the special graphics primitives, this is a wonderful enhancement. It is now possible write an interactive computer game that has text and graphics on the screen at the same time.

I greatly appreciated the explanation in the manual of what goes on with Apple's color system when I tried working with colored text. Colored text, one of the features I had eagerly anticipated using, turns out to be very problematic. In

fact, I had so many problems getting clear colored text on the screen that I began to wonder why this feature was added at all. I also wished that this part of the manual had included a chart of what color text will display well on what color background. Even when using bold-face type style, which the manual suggests, most of the color combinations I tried were not satisfactory. Using the ZOOM feature to edit colored text is not helpful here, since the editor operates only on black and white. Filling in missing parts of blue letters gave me blue and white letters. Inverse text, which appears as a black strip with letters in color, appears to work best. Although I was disappointed with the colored text, I was delighted with the black-and-white display.

Text on the graphics screen has not turned Logo PLUS into a word processor, however. The available text space is only the size of the screen. When the cursor reaches the bottom of the screen, it reappears at the top. Text space is thus limited in the same way as picture space and is saved as if it were graphics when you use the SAVEPICT command.

With the new Logo PLUS, it is possible to create data files. The *Getting Acquainted* booklet includes a project which takes you through the steps of creating and storing such files.

What this all adds up to is a new version of Logo well worth a hard look. Logo PLUS has been produced with many of the user's needs in mind, both in the special features which have been included and in the remarkably thorough booklet which accompanies the language and utilities disks. Logo PLUS is definitely an exciting new development.

Logo PLUS comes with a language disk, a utilities disk and the *Getting Acquainted with Logo PLUS* booklet. Site licenses, lab packs and single packs are available. There is also a trade-in or, more properly, a trade-up option for owners of the old Dos 3.3 Terrapin Logo.

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

Terrapin Announces Logo PLUS

Terrapin, Inc., producer of the original version of Logo, introduces Logo PLUS, a new version of Logo for 128K Apple II computers. Logo PLUS contains over 75 additional commands and features, giving users more power and flexibility. Logo PLUS is easy to learn since the new features are incorporated into the "classic" Logo language — all of the familiar commands function in the same manner.

With Logo PLUS it is easy to type text onto the graphics screen in a variety of colors, fonts and styles, such as bold, underline, italics or inverse. The font editor allows users to modify predesigned character sets or create their own designs for new alphabets, mathematical symbols, foreign language characters or even secret codes.

The easy-to-use shape editor lets users design a variety of shapes for the Logo "turtle" and stamp them on the screen. Animations can then be created by moving, rotating and changing the new turtle shapes.

Unique to Logo PLUS is a special ZOOM feature for detailed screen editing and flipping the entire design to its mirror image. In addition, a color detection command, which gives the turtle a "sense of touch," and a color fill command offer users the ability to create more sophisticated pictures, maps, charts and graphs, and interactive programs. Add music with the built-in NOTE command and print the finished graphics screen to an Imagewriter printer with a single command.

The Logo PLUS Utilities Disk contains over 40 programs, including sample projects and procedures, helpful tools, demonstration programs and more. Logo PLUS is also compatible with Terrapin's wide range of curriculum materials, which provide additional activity suggestions and projects.

Logo PLUS is based in ProDOS rather than DOS 3.3, yet it has been designed so that most people will never notice the difference in operating systems. Logo PLUS runs on any Apple IIe, IIc or GS computer with 128K of memory. For more information, contact

Terrapin, Inc.
376 Washington Street
Malden, MA 02148
617-322-4800

Logo Connections

StarNet: A Fourth-Generation Computing Environment

by Glen and Gina Bull

The first versions of Logo did not have turtle graphics. The terminals used at the simply time did not have graphics. These terminals consisted of Teletype-like devices that printed on paper rather than on a monitor screen. As graphics terminals became available, it became possible to add many of the features that now characterize Logo. There are two implications. The first is that Logo is not a static language, but one which evolves as the computing environment changes. The second is that the nature of this environment has a great deal to do with instructional uses.

Thus far there have been three generations of computing environments. The first generation consisted of mainframe computers fed in a batch processing mode. A program consisting of a stack of punched cards would be turned in at the computing center. The next day the results could be picked up. There was not much scope for the interactive style of programming used in Logo. The second generation of computing environment was made possible by time-share systems. These systems permitted individuals to communicate with the computer in an interactive style. This made it possible to type a question and receive an answer in seconds rather than hours or days. The third computing environment became possible with the advent of personal computing. Each person could have her own computer. A fourth computing environment consisting of networked systems is now evolving.

The Evolution of Personal Computing Tools

Before we take a look at the implications of networked environments, we would like to take a highly personal tour through the history of personal computing tools.

1. Microcomputers without Applications

Our first personal computer is still stored in the basement in working order. It has four kilobytes of memory, and was constructed by the electrical engineering department. Its programs were assembled on the mainframe and then downloaded onto paper tape. The paper tape was fed into the computer. This computer did not come with any tools. Instead, it was necessary to create and download separate programs for each task.

2. The First Tools

Our second computer is stored beside the first one. This computer initially had 16 kilobytes of memory, and used the CP/M operating system. A computing language (Tiny BASIC) stored on paper tape could be loaded into the computer, making it possible to program directly on the computer. Later we expanded the memory to 64 kilobytes and obtained a "glass Teletype" terminal with a screen for text but no graphics. An early word processor written in BASIC provided the first general purpose tool for this computer.

3. The Advent of Logo

Our third computer was the first one which could run Logo. It was a TI 99/4 computer donated by Texas Instruments. Logo was supplied on an audio cassette and loaded into the computer using an ordinary tape recorder interfaced to the computer. This system usually worked — except on rainy days. Sometimes there were other problems. One teacher was distressed to find that his daughter had replaced a semester of programs with Christmas carols. However, the TI computer had a color monitor and graphics, and in some respects was years ahead of its time. Although this computer did not have a word processor, Logo provided a general purpose tool for instructional applications.

4. Multiple Tools

We later acquired an Apple II+ computer. It had a word processor, a spreadsheet (Visicalc), and Logo. For the first time we had a computer with more than one general purpose tool. However, information could not be readily shared between applications. For example, at first the information in the spreadsheet could not be easily moved to the word processor. In order to include spreadsheet figures in a word processed document, we would first print the spreadsheet, and then retype the figures in the word processor. Similarly, programs generated with Logo were first printed, and then retyped in the word processor for our Logo Exchange columns.

5. Information Interchange

Later versions of Logo made it possible to generate text files which could be shared with other programs. These text files are sometimes known as ASCII files, for American Standard Code for Information Interchange. At this point it was no longer necessary to print a Logo program and retype it into a word processor. Instead, the Logo procedures could be read directly into a word processor. This was much more convenient, and resulted in fewer typographical errors as well. It also became possible to save data from a spreadsheet in text form, and read the data into Logo for further manipulation and generation of graphics.

6. The Current Status

Despite the convenience of interchanging information in this way, the method differs slightly for each word processor

and spreadsheet. The practice of interchanging information with other programs never really became popular with Logo users, partly because it is not possible to specify a universal method for exchanging information with all word processors. Therefore articles about this use are only of interest to Logo users who own the specific word processor discussed.

7. The Next Generation

When the Macintosh was designed, a built-in tool called the *clipboard* was included for exchange of information. This makes it possible to develop a graphic with MacPaint, copy it to the clipboard, and then paste in into MacWrite. Since a standard method for exchanging information among applications is provided, Macintosh users tend to use several tools in combination with one another (MacWrite, MacPaint, Hypercard) to a much greater extent than otherwise would be the case. Logo Connections was conceived of as a column which would be about using Logo in combination with other tools. There is not a similar column called Clipboard for the Macintosh, as far as we know, since this practice is intuitive for Macintosh users.

Despite development of a standard method for exchanging information via the clipboard, it was still necessary to exit one program before going to another. It took half a minute or so to leave MacPaint and start MacWrite. A program called Switcher solved this problem. Switcher made it possible to keep two programs in the memory at the same time. It became possible to switch from MacPaint to MacWrite with the flick of a key. Later Apple incorporated this capability into its new operating system, Multifinder. IBM is developing a similar capability in OS/2 (Operating System 2). Both of these operating systems will provide the capacity to keep two or more programs in memory at the same time, and easily move between them.

Networking: The Fourth Generation

The first three generations of computing environments consisted of batch processing, time share systems, and personal computing. The next generation of computing environment now evolving consists of networked systems. An electronic computer network called the Internet now connects universities, government agencies, and businesses across the United States. This column is being typed using Word Perfect on an IBM Model 30 computer. When it is completed, it will be sent over the network from GBull@Virginia to Yoder@Oregon. Sharon Yoder, the editor of *Logo Exchange*, will then capture the document, edit it using Microsoft Word, and then import it into Pagemaker on a Macintosh. In columns which involve Logo code, LogoWriter procedures developed on an Apple IIe will be transferred via modem to the IBM. Thus a typical column may involve different applications on different computers on separate sides of the country. The network makes it possible to share information among applications.

This may seem fairly remote from a school environment, but the implications are not so distant as you might think. A project called StarNet is under development at the Technical Education Research Centers (TERC) in Cambridge, Massachusetts. This pilot project will provide teachers with a comprehensive set of computing tools, and link them together in an instructional network. Bob Tinker, president of TERC, describes the intent of the project in this way,

> An approach we have termed microcomputer-based laboratories, or MBL, uses the microcomputer, equipped with a small array of transducers and general-purpose software, as a universal instrument performing the functions of storage-tube oscilloscope, counter-timer, frequency analyzer, and other devices. Easy-to use software makes these functions readily available to the student. The computer is also a universal tool for analyzing, comparing, storing and graphing data, as well as generating models and exploring theories. Telecommunications can bring students together to cooperate in conducting projects and setting up networks that are much more complex than any one student could handle alone. Thus, MBL increases student capabilities and productivity while reducing costs, and telecommunications technology fosters collaboration and reduces the demand on each teacher.

The tools which classes will receive will include a programming language consisting of LogoWriter with MBL extensions, a word processor, a spreadsheet, and a database (Appleworks), a drawing program, a telecommunications program, and other tools for analysis, modeling, and simulation. A macro facility will serve as the glue which will make it easy to shift from one application to another. This environment will encourage Logo connection using Logo in combination with other computing tools.

A group of teachers with a common set of tools linked by a network adds a new dimension to instructional computing. Because each teacher will have the same set of tools, it will be possible to specify methods for exchanging information among applications that will work for everyone. The macro facility will make it easier to develop these information interchange techniques. The network will provide a way for classes to exchange programs, data, and ideas.

Logo Connections -- continued

The Internet, as its name implies, is not a single network. Rather it is a group of interconnected networks linked by electronic gateways. It is possible that at some point gateways will be developed which will link StarNet to the Internet. If this occurs, teachers will not only be able to talk with one another but with consultants in universities, businesses, and government agencies throughout the country. The network will not spread throughout school systems overnight. It took about a decade for the third generation computing environment (personal computing) to permeate school systems. A decade ago few school systems had microcomputers. Now there is hardly a school in the country which does not. The speed with which the fourth generation computing environment of networked teachers travels will depend in part on the success of pilot projects such as StarNet. However, when networks link teachers across the country it will change forever the way we think of computing.

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

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

MathWorlds

edited by
A. J. (Sandy) Dawson

When doing my doctoral work, I happened to take a course in number theory and was almost instantly intrigued by the simple, elegant but oftimes very difficult problems which could be studied. Indeed, number theory became my favorite mathematical topic, and I have maintained that interest over the years. Perhaps it was the fact that I knew of one person whose doctoral dissertation in mathematics consisted of one page, because that page contained a simple and elegant solution to one of those very difficult number theory problems.

So when I received a manuscript from Michael Tempel which dealt with a number theory problem, I was immediately smitten. I hope you enjoy his material as much as I did.

Easy As 1 1 2 2 3

by
Michael I. Tempel

On the front page of the science section of the New York Times this morning there was an article headlined "Intellectual Duel: Brash Challenge, Swift Response". John Conway, a mathematician best known for inventing the Game of Life (an early version of the Phantom Fish Tank, a recent LCSI product) made a presentation to a symposium at the Bell Labs in New Jersey. He presented a number series that he found interesting. It starts like this:

```
[1 1 2 2 3 4 4 4 5 6 7 7]
```

Can you guess Conway's rule?

Here's a hint. It's similar to the Fibonacci series in that the next number in the series is calculated by adding together two other numbers in the series. The Fibonacci series looks like this:

```
[1 1 2 3 5 8 13 21 34 55 89] etc.
```

By adding together the last two numbers in the Fibonacci series we get the next number. The series starts as

```
[1 1]
```

1 + 1 = 2 so we have

```
[1 1 2]
```

1 + 2 = 3 so it becomes

```
[1 1 2 3]
```

The next addition, 2 + 3 produces the series

```
[1 1 2 3 5]
```

and so it grows.

In Conway's series, the selection of the two numbers which are added together is not so simple. Do you give up?

Here's how it works:

Take the last number in the series and use it as a counter. Count backwards that many places and see what number you find. If the series is up to:

```
[1 1 2 2 3 4 4 4]
```

then the counting number is 4. In the fourth position from the end we find a 3. Remember that. Now use the same counting number to count forward through the list. The number in the fourth position from the front of the list is 2. Now add 3 and 2 to get the next number. The series becomes

```
[1 1 2 2 3 4 4 5]
```

Now the counter is 5. In the fifth position from the end is 3. In the fifth position from the beginning is 3. (They happen to be the same number). The next number is 6 so the series becomes

```
[1 1 2 2 3 4 4 4 5 6]
```

So what? Why does this rate front page treatment in the New York Times? Well, Conway noticed something interesting about the series. The last number in the series divided by the length of the series is always a number close to one half. When the series is

```
[1 1 2 2 3 4 4 4]
```

the last number is 4. The length of the series is 8. The ratio is exactly 0.5. Here are the next four steps of the series followed by the ratios of the last number to the length of the series:

MathWorlds -- continued

```
[1 1 2 2 3 4 4 4 5]          0.5556
[1 1 2 2 3 4 4 4 5 6]        0.6
[1 1 2 2 3 4 4 4 5 6 7]      0.6364
[1 1 2 2 3 4 4 4 5 6 7 7]    0.5833
```

Conway and his wife proved that as the series gets longer, this ratio converges on 0.5. Is this worthy of a New York Times headline? Maybe not, but Conway offered $1,000 to anyone in the audience who could find the point in the series beyond which the ratio never deviated from .5 by more than 10%. Collin Mallows took up the challenge. After a few days of "...messing around on the backs of envelopes" he pulled out his Cray super computer, determined that it was the 3,173,375,556th position in the series and collected his prize. That's news. Actually, due to a slip of the tongue Conway offered $10,000, not $1,000. He claimed otherwise, but the people at the Bell Labs had the whole show on video tape. Mallows agreed that he probably meant $1,000 and accepted the lesser amount.

Well, I don't have a Cray, but an hour after reading the article I found myself confined to an airplane seat on the way to Montreal with a Toshiba 1000 running LogoWriter. I started to play.

You may have already noticed that I've been representing Conway's series as a bracketed list. The Times used the text book convention

```
1,1,2,2,3,4,4,4,5,6,...
```

Since I know Logo, representing the series as a list seemed natural. Also, it put it in a form that may be manipulated by Logo procedures.

To start with, I figured I'd write a Logo program to generate Conway's series. My approach was to write a procedure that would take any instance of the series as input and report the series with the next number stuck on the end.

```
conway [1 1]
```

should report

```
[1 1 2]
```

```
conway [1 1 2]
```

should report

```
[1 1 2 2]
```

...and so on.

The new number is obtained by adding two numbers so my Logo procedure will need to use +. The last number in the series is used as a counter. The Logo primitive **last** can be used to report that number. **Item** can be used to report the number found at a particular position in the list. **Lput** could append the new number to the end of the current series. Here's what I came up with:

```
to conway :series
output lput (item last :series
   :series) + (item last :series
    reverse:series):series
end
```

But wait. Reverse isn't a primitive. No problem:

```
to reverse :list
if (count :list) = 1 [output :list]
output lput first :list
reverse butfirst :list
end
```

I tried my procedure.

```
show conway [1]
[1 2]

show conway [1 2]
[1 2 3]

show conway [1 2 3]
[1 2 3 4]
```

Something was wrong. These were just the counting numbers. The procedure looked right. After puzzling over this for a while I realized that the procedure was fine. I had started with the wrong initial series. The minimum series, as with Fibonacci, is [1 1].

```
show conway [1 1]
[1 1 2]

show conway [1 1 2]
[1 1 2 2]

show conway [1 1 2 2]
[1 1 2 2 3]
```

It worked!

I wrote a procedure to automatically generate ever longer conway series:

```
to many.conway :series
print :series
many.conway conway :series
end
```

I started generating successive instances of the series along with the ratio of the last number to the length of the list. In Logo this ratio is

```
(last :series) / (count :series)
```

I graphed the changes in this ratio as Mallows did with his Cray, only I used the turtle. Here's are the procedures I used:

```
to setup
rg
pu
setpos list minus 159 50
pd
end

to graph :series
if xcor > 155 [stop]
setpos list xcor + .5 100 *
   (last :series) / (count :series)
graph conway :series
end
```

Setup puts the turtle near the left edge of the screen and at a ycor of 50. The first line of graph checks to see if the turtle is getting close to the right edge of the screen and stops the procedure if it is. Then the turtle's xcor is moved half a turtle step to the right and the ycor is set to be 100 times the ratio of the last number in the Conway series to the length of the series. Multiplying the ratio by 100 expands the fluctuations so they are visible.

Then **graph** is called again with an input that is the next instance of the series. Start the graph with

```
graph [1 1]
```

Here's what it looks like after the series reaches a length of about 300 numbers.

The low points on the curve are where the ratio of **last :series** to **count :series** is .5. As the series grows, two things happen. The interval between the points where the ratio exactly equals .5 stretches out. Second, the high points become lower. From this Logo graph it certainly does look like the ratio will converge on .5. The graph is noticeably flattening even before it reaches a length of 300.

I added a line to graph to see exactly how these low points were spaced

```
to graph :series
if xcor > 159 [stop]
if ((last :series) / (count :series))
    = .5 [print count :series]
setpos list xcor + .5 100 *
   (last :series) / (count :series)
graph conway :series
end
```

This prints the length of the series whenever the ratio of the last number to the length of the series is exactly .5. Here's what I got:

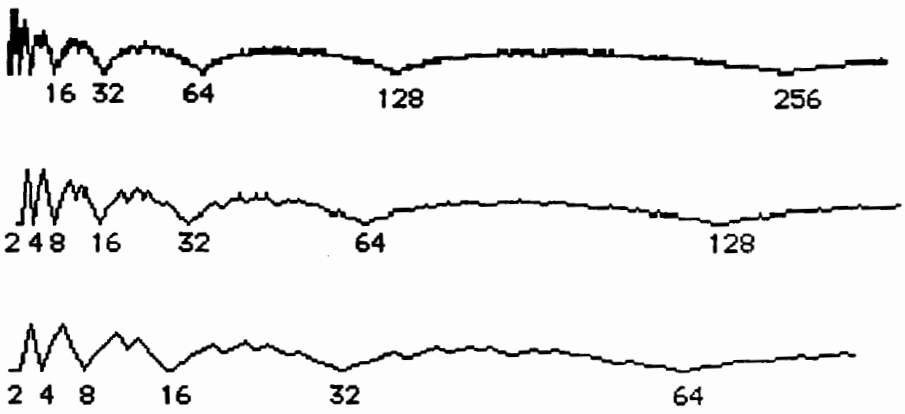
```
2
4
8
16
32
64
128
256
```

Wow! That looks familiar.

The left side of my graph is unclear, kind of squashed together. I rewrote the graph procedure to take larger steps in the x direction, 2 instead of .5. This spreads out the graph but it doesn't get as far into the series. I then increased the x step to 4 to spread things out even more.

Here are all three graphs:

MathWorlds -- continued



The numbers show the length of the series at each low point, that is, when the crucial ratio is .5.

Now what about the high points? As you can see from the graphs, the rise and fall between each pair of low points is not smooth. However, there is a single high point in each interval. Here are the positions where those high points occur:

Position in the series	Number in that position	Ratio
3	2	.6667
6	4	.666
11	7	.6364
23	14	.6087
44	26	.5909
92	53	.5761
178	101	.5674

Is there any pattern here? Except for the first two ratios, which are equal, each one is smaller than the one before. Also, the difference between one ratio and the next gets smaller as we move along. According to Mallows, by the time the series reaches the 3,173,375,556th place that ratio will be below .55 and remain that way as the series continues to grow.

Well, we're no where near that point and Logo is running out of space in memory. I suppose we can get back to this once Logo is running on a Cray. I think that's enough for now. I intend to keep playing with these ideas. Maybe you will, too. Let me know if you come up with something interesting.

Michael Tempel is director of Educational Services for LCSI in New York. He can be reached at 330 W. 58th, Suite 5M, New York, New York 10019 and on Compuserve at 73007, 532.

The Network for Action in Microcomputer Education, NJ Department of Education,
Association of Math Teachers of New Jersey, and NJ School Boards Association present:

The 1989 New Jersey Educational Computing Conference

March 10-11, 1989
Raritan Valley College
Somerville, New Jersey

The keynote speaker will be Tom Snyder of Tom Snyder Productions. The conference features guest speakers, dozens of educational presentations, vendors, and exhibits. There will be numerous Logo presentations, including full-day LEGO TC logo and LogoWriter workshops on March 9, 1989.

For more information, contact:
NJECC '89
N.A.M.E. - Fallon Education Center
51 Clifford Drive
Wayne, NJ 07470
(201) 633-3121

Assessing Logo Learning in Classrooms

V. Strategies for Learning Through Exploration

by Dan Watt

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

This month I will write about each of the subheadings included in the fifth cluster of critical aspects, Strategies for Learning Through Exploration, illustrated with examples of student work. For a fuller sense of what we mean by critical aspects of Logo learning, and our rationale for this approach to assessing Logo learning, please read the September '88 column in this series.

Is Brad wasting his time?

Brad is a fourth grader who engages in lots of Logo explorations. According to his teacher,

> He works at an average fourth grade level in all areas ... although he needs a lot of teacher reminding, both behaviorally and academically. ... However, when it comes to working with Logo, Brad has remarkable perseverance. He is anxious to complete tasks he sets for himself and appreciates teacher help when stumped.

Nevertheless, she was concerned about what he was learning, and often unclear about how to help him. For one thing, he had so many apparently unrelated procedures in his workspace. For another, he had lots of different files on his disk, all of which had many of the same procedures in them, although with slight differences.

Some of Brad's procedures were part of a large city project that he had worked on for some time. Others seemed to be unrelated to anything in particular. Many of them used REPEAT, often with large numbers and changes of pen color and background color. When he was asked what the different procedures did, or why he made them, Brad had difficulty remembering.

What was Brad learning from his explorations? Was he wasting his time? Or did his this work have some purpose that he could not articulate and his teacher couldn't easily detect? And how could she support him to use exploratory learning more effectively? These are the types of questions I'll be addressing this month.

Process Knowledge Vs. Content Knowledge

For this column and the next two, I will focus on some of the *process* components of Logo learning rather than on the *content*. In the last four columns, I shared examples of how students have used some of the specific content embedded in Logo by its developers: programming ideas involving procedures and variables, and mathematical ideas embedded in turtle geometry. Our source of knowledge was clear and specific: we looked at actual printed examples of children's work, either completed or in progress.

In looking at the *processes* by which students actually construct knowledge while working on their Logo projects, I must confess that I'm on much shakier ground than when looking at the *content* of their learning. Examples of students' printed work can tell us a great deal about which ideas they are using, and how their ideas have changed over time, but they give very few hints as to how they selected the particular ideas they decided to use, how they moved from idea to idea, how they collaborated with other students, and how they handled problems they encountered along the way.

In our collaborative research project (Watt and Watt, 1988), each teacher selected two students to observe as often as possible, regularly collected their work in progress (including their written plans, notes and hand drawings), and engaged these students in conversations about what they were doing and thinking. [One teacher in the project used IBM Logo dribble files — exact records of every command her students actually typed (Olive, 1986).] Most of the teachers who worked with us reported that insights gained by carefully following two students helped them support the learning of many of the other children in their classes as well. (See Watt, M., 1989, for a case study.)

Classroom teachers who want to understand more about how their own children work through projects and gain knowledge might also benefit by reading reports of careful research by other observers. Sylvia Weir's outstanding book, *Cultivating Minds*, offers a number of mini-case-studies of Logo learning in process (Weir, 1987). The *Final Report of the Brookline Logo Project* also offers a number of detailed case studies of Logo students at work (Papert, et. al, Watt D., 1979a, 1979b). Readers interested in more formal research can refer to Douglas Clements' series on Logo and problem-solving, which have been published in his *Logo Exchange* column, Logo: Search and Research, beginning in October, 1988 (Clements 1988-89).

Assessing Logo Learniong In Classrooms -- continued

A. Keeping Records of Exploratory Work

Logo students who regularly record what they have done, in the form of handwritten notes, sketches and printouts of Logo pictures and procedures, are much better able to recall what they have done, why they did it, and what order they did things in. They can look back to see how something was done before, and therefore don't have to endlessly repeat previous work. But experience has shown that Logo students rarely keep thorough records on their own. Teachers must support this in many ways: teach students to print their own work; have students keep regular journals; have students to keep records of exploratory projects on forms designed specifically for that project.

If dribble files are available, students can use them to trace their own work. Andy (grade 5) created the design shown in Figure 1 by typing TRIANGLE, LEFT 5, over and over again, using REPEAT and individually typed commands. When his work session was over, he asked his teacher for a copy of his dribble file. He then added up all the times that the sequence, TRIANGLE LEFT 5 was repeated, and wrote the procedure, TO LOTS, shown below.

```
TO TRIANGLE
 REPEAT 3 [FD 50 RT 120]
 REPEAT 3 [FD 75 RT 120]
 REPEAT 3 [FD 100 RT 120]
END

TO LOTS
 REPEAT 70 [TRIANGLE LT 5]
END
```

Figure 1

B. Taking Advantage of Accidents

Explorations can lead to useful results that can be used for purposes other than what a student might have had in mind originally. For example, one day Brad experimented with rotating figures. In attempting to complete the rotated design R.T, shown in Figure 2, he used many different values for the number of repeats and the angle of rotation. One of them created a filled-in circle to which he gave a new name, R.T. Brad, who was extremely interested in filling shapes, used this idea in a later project, BALLOON. (As you can see, he was also exploring the effects of different background and pen colors.)

C. Using and Looking For Patterns

Andy was interested in complex geometric shapes. He experimented with nested shapes, and with rotated shapes (Figure 1, above, and Figure 3). Notice how he used a pattern: a constant increase in size, to draw both his TRIANGLE and SPIRALS procedures. He attempted to use a similar idea to go from a pentagon to a hexagon (Figure 4). Although Andy was looking for symmetries in his work, he had not yet discovered a key pattern involving 360 degrees. (See this column for December 1988.)

```
TO R.T
 REPEAT 12 [SQ RT 23 SQ]
END

TO R.T.
 HT REPEAT 180 [SQ RT 2 SQ]
END

TO BALLOON
 FILLED.CIRCLE SUN.B.B
END

TO FILLED.CIRCLE
 HT REPEAT 180 [SQ RT 2 SQ]
END

TO SUN.B.B
 RT 45 PU FD 100 PD HT REPEAT 76 [SQ RT 5]
END
```

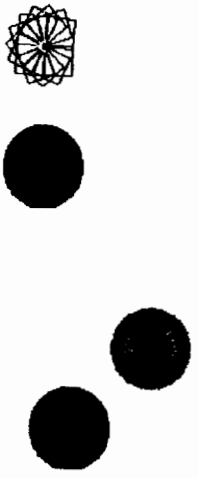


Figure 2

```
TO TRIANGLE
 REPEAT 3 [FD 50 RT 120]
 REPEAT 3 [FD 75 RT 120]
 REPEAT 3 [FD 100 RT 120]
END

TO SPIRALS
 REPEAT 36 [FD 1 RT 10]
 REPEAT 36 [FD 2 RT 10]
 REPEAT 36 [FD 3 RT 10]
 REPEAT 36 [FD 4 RT 10]
 REPEAT 36 [FD 5 RT 10]
 REPEAT 36 [FD 6 RT 10]
END
```

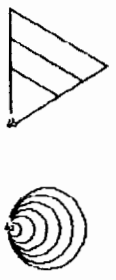


Figure 3

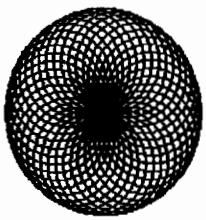
```
TO PENTAGON
 REPEAT 5 [FD 50 RT 72]
END

TO HEXAGON
 REPEAT 6 [FD 60 RT 82]
END
```

Figure 4

D. Testing the limits

Understanding extreme cases, using large and small numbers, is often a very useful exploration strategy. Here are some more of Brad's explorations (Figure 5). Notice that although he used 9999 as a standard "large number," the other numbers he used were carefully controlled to produce specific effects. D.5.4, for example, fills the entire screen.

```
TO DESSY.1
 RT 45
 FD 9999
 REPEAT 45 [BG 5 BG 3 RT 45 FD 9999]
 REPEAT 180 [FD 1 RT 1 FD 77 RT 77]
END
```



```
TO CHEEZE
 REPEAT 50 [LCIRCLE 40 LT 9999]
END
```

Figure 5

```
TO CITY.2
 SQUARE HT
 REPEAT 45 [FD 1 RT 90 FD 90 LT 90 FD 1 LT 90 FD 90 RT 90] PC 1 PU HOME PD
 LT 90
 FD 908 RT 90
 RECTANGLE
 LT 90 FD 20
 RT 90 RECTANGLE
 FD 20
 RT 90
 FD 40 LT 90
 FD 20 LT 90
 FD 40 RT 90
 FD 20 RT 90
 FD 40 LT 90
 FD 20 LT 90
 FD 40 RT 90
 FD 20 BK 40
 BK 60 RT 90
 FD 80 FD 15
 LT 90 STREET
 PU
 BK 60
 LT 90
 FD 35
 PD
 PC 0
 FD 60
 LT 90
 FD 45
 LT 90
 FD 60
END
```

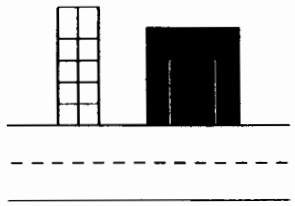


Figure 6

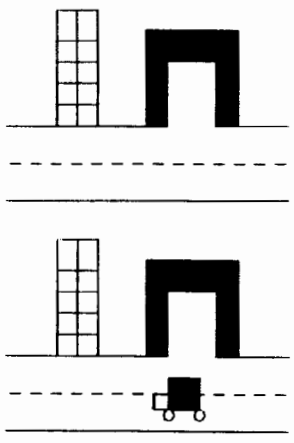
E. Making and Testing Hunches and Hypotheses

Students should be encouraged to guess, predict, to formulate rules, as part of the process of exploration. And, they should be encouraged to write them down, tell how they tested them, and what the results were. Andy and Davis, kept records of their work, and could tell their teacher exactly which ideas they were testing. But many students move back and forth between random exploration and careful testing of hunches or looking for patterns. If they do not keep records, they may forget about the thoughtful problem solving they were doing, and explain that they were "just messing around." Brad was a frequent victim of his own amnesia in this way. His teacher was able to conclude that his work was valuable, only by carefully analyzing it after the fact!

F. Generalizing from one or more examples

Brad's work at filling in squares provides an excellent example of how an exploration can be adapted, modified and eventually generalized. Brad explained that his first fill-in procedure was R.T. (Figure 1, above). As he observed how a circular shape was filled in, by taking small rotations, he got an idea for filling in a square shape using small turtle steps. First, he figured out how to fill in a square that was part of his CITY project (figure 6). Later he used the same approach to fill a smaller square to draw a truck, which was to be part of the city (Figure 7). His ultimate objective was to have a garage door open, have the truck come out, and then have the door close. To open the door, he created the OPEN procedure, which (when used with the turtle's pen set to the background color) drew an open door. Then a similar procedure, CLOSE, was used to refill the empty doorway so that it appeared to close.

```
TO OPEN
 LT 90 REPEAT 30 [FD 45 LT 90 FD 1 LT 90 FD 45 RT 90 FD 1 RT 90]
END
```



```
TO TRUCK
 SQ1 30
 LT 90
 SQ1 15
 LARC 5
 BK 30
 LARC 5
 REPEAT 15 [FD 30 RT 90 FD 1 RT 90 FD 30 LT 90 FD 1 LT 90]
 PU FD 30 LT 90 FD 22.5 RT 90 FD 15
END

TO CLOSE
 REPEAT 30 [FD 45 RT 90 FD 1 RT 90 FD 45 LT 90 FD 1 LT 90]
END
```

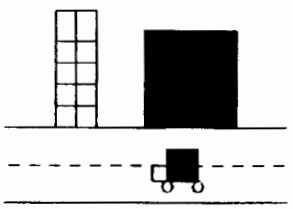


Figure 7

Asessing Logo Learning in Classrooms -- continued

Finally, Brad generalized the idea by writing a procedure, which used a variable to fill in any square!

```
TO FILL.SQ1 :SIDE
SQ1 :SIDE
REPEAT :SIDE/2 [FD :SIDE RT 90
   FD 1 RT 90 FD :SIDE LT 90 FD 1
   LT 90]
END
```

G. Sharing Ideas With Others

Finally I want to suggest the importance of learning from others as a key strategy for learning through exploration. Once Brad figured out how to fill in a square, several of his classmates began to experiment with filled-in shapes themselves. Brad's persistence in filling in shapes, and his willingness to share his work with others, led to many fruitful explorations on the part of his classmates. A classroom climate that encourages sharing — of work in process, and of the process itself, as well as of finished work — is a critical factor in successful exploratory learning.

References

Clements, D. (1988-89). "Logo: Search and Research," *Logo Exchange*, October 1988 - spring 1989: a series of reports about research on Logo and problem-solving.

Olive, J. (1986). "The Collection and Analysis of Qualitative Data in a Logo Learning Environment Using Dribble Files." *Proceedings of the 8th Annual Meeting of PME-NA*, Michigan State University, East Lansing MI.

Papert, S., et. al. (1979). "Final Report of the Brookline Logo Project, Part II: Project Summary and Data Analysis," *Logo Memo #53*, MIT Logo Group, Cambridge MA.

Papert, S. (1980). *Mindstorms: Children,Computers and Powerful Ideas*. Basic Books, New York NY.

Watt, D. (1979a). "Final Report of the Brookline Logo Project: Part III, Profiles of Individual Students' Work." Logo Memo #54, MIT Logo Group, Cambridge MA.

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

Watt, M. (1989). "A Reflective Teacher's Interventions with a Tentative Learner of Logo," *The Computing Teacher* (in press).

Watt, M. and Watt, D. (1988). "Teachers as Collaborative Researchers: Professional Development Through Assessing Logo Learning," *Proceedings, National Educational Computing Conference*, Dallas TX.

Weir, S. (1987). *Cultivating Minds: A Logo Casebook*. Harper and Row Publishers, New York NY.

The work described here was conducted at Education Development Center (EDC), 55 Chapel Street, Newton Massachusetts, and supported in part by the National Science Foundation under grant # MDR 8651600, Exploratory Research on Critical Aspects of Logo Learning. The ideas and opinions expressed are those of the author and do not necessarily reflect the views of EDC or the National Science Foundation.

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

A LogoWriter Tip

When working with LogoWriter in a laboratory setting, it is often useful to create sample pages for the students to use. One of the most useful features of LogoWriter 2.0 in this situation is that you can LOCK a page, and the students can use it without destroying the original form of the page. (An example might be a demonstration FILL page, where there are shapes on the page. The students use turtle move keys to learn how to move the turtle into the shape, press ESC, and then fill the shape with color).

Once you have created the sample page and LOCKed it, and you may need to save the page on each of the student disks (I have 15 lab disks in my lab.) Whatever technique that you use for copying onto the new disk (ESC, savepage, copyfile) will transfer the page unlocked. When faced with transferring several sample pages at the same time to all of my lab disks, it seemed like a good idea to use LogoWriter's power to help. Thus, the following procedure:

```
To correct
getpage "pagename.to.be.locked
lock
wait 20
getpage "pagename.to.be.locked
lock
end
```

You can put as many pages as you need into this procedure. The WAIT 20 was inserted to have enough time to see that the procedure was running as we wished.

Put this procedure onto a page called CORRECT. Then, at each computer, put the disk with the CORRECT page on it in the disk drive, and type

```
GETTOOLS "CORRECT.
```

Once that is done, you can put the lab disks in each machine, go to a NEW PAGE, and type

```
CORRECT
```

and the procedure will correct (lock) all the pages for you.

Warning note: If the page to be locked has a STARTUP procedure, that procedure will run first. Depending on the STARTUP, you may have to modify the CORRECT procedure. For example you might use CG or CT before LOCKing the page.

Jandy Bird
Conover Road School
080 Conover Road
Colts Neck, N.J. 07722
CIS 73517,3270

Terrapin Logo on the GS

(Here are some hints for using Terrapin Logo on the Apple II GS. Many of these ideas apply to other versions of Logo as well, so if you have a GS, read on. — editor)

The Control Panel

The Control Panel offers many different options for using the Apple IIGS computer. The settings for most of these options do not affect the way Logo runs and should generally remain at their default values (indicated by a check mark on the screen.) Refer to your Apple IIGS manual for complete information about the Control Panel options.

To temporarily leave Logo to modify Control Panel options, you must press Control-Reset. (You cannot access the Control Panel from within Logo using the standard Control-Open apple-Escape command.)

When you have finished making changes, return to Logo from the Control Panel by selecting QUIT from the main menu. Type Control-Y followed by Return and the Logo question mark prompt will appear. Then type Control-G. All your procedures and variables will be intact, but any graphics already on the screen will be lost, so you may want to save your design before entering the Control Panel.

Printing Tips

BAUD — The BAUD setting in the Printer Port menu must be set to 9600 in order to print Logo graphics or text.

SLOTS — If you are using a printer interface card, place it in slot 1. Set the SLOT 1 option (in the SLOTS menu) to YOUR CARD. If you are connecting an Imagewriter printer directly to your Apple IIGS with a cable, set the SLOT 1 option to PRINTER PORT. You can then use any of the graphics screen dump programs on the Logo Utilities Disk and in the Utilities II Package.

Removing the graphics screen border

DISPLAY — It is possible to use the entire graphics screen with no border. If you set the border color to black (this is found in the DISPLAY menu of the Control Panel), the Logo screen will act as if there were no border at all. The total area will now be black and lines will wrap around the entire screen. If you change the background color using Logo's BG command, the whole screen will change color. Note that the x- and y-coordinates are the same for both screens.

Music

SYSTEM SPEED — In order to run the music tools on the Logo Utilities Disk, you must set the System Speed to NORMAL.

From the *Terrapin Times*, Winter 1987-88, p.2

Logo: Search and Research

The Nature of the Problem
by Douglas H. Clements

Gina and Robbie were discussing their Logo project, which was still in its infancy.

Gina: What do we make here?
Robbie: That's where the sun is gonna go. We gotta go over here and do a circle with curvy lines around it like our drawing.
Gina: So making curvy lines will be the hard part to figure out....

Their thinking centered on one mental metacomponent (process): Deciding on nature of the problem. They were determining just what the problem was all about and what would be required to solve it. We know that getting students to understand what is being asked of them is often half the battle. Why might we believe that Logo programming could help foster this process? Is there any evidence that it actually does? What can we as teachers do in our role of facilitators? We'll answer each of these questions in turn.

Why Believe that Logo Can Help?

Logo's turtle graphics allows students to pose numerous significant problems of varying levels of complexity for themselves. They must generate ideas for their own projects, represent these as goals, and identify the specific problems involved in reaching these goals. So they must determine the nature of each problem and subproblem.

In other words, Logo provides an environment not only for problem solving, but also for problem posing. Certain educational environments will encourage such mental activities. Students are most likely to develop this metacomponent if they create a large proportion of their own projects and if these projects are important and interesting to them.

Is There Evidence that Logo Truly Helps?

One observational study measured the frequency of occurrence of specific metacomponents (Clements & Nastasi, 1988). Students within the Logo environment displayed these problem-solving processes to a greater degree than those in CAI environments. In particular, they were more likely to clearly show that they were deciding on the nature of the problem. Gina and Robbie's discussion was taken from this study.

But, do students use these new abilities on tasks outside of Logo? Assessing transfer effects to non–computer tasks, Clements (1986) found that a Logo group significantly outperformed both CAI and non-computer groups on several metacomponents, including deciding on the problem. On one task, students were given items that were posed as simple detective stories to be solved. For example, they were told:

> "I'm trying to write a detective story, but I got stuck. Can you help me think of the clues the detective should use to prove who committed the crime? The detective had to find out which sneaky kid hid the cookie jar. The cookie jar was taken from the high cupboard that only the biggest kid could ever reach. Missy liked cookies. Her bigger brother was Billy. Billy's bigger brother was Timmy. They loved cookies. How could the detective prove who took the cookie jar for sure?"

Students were given credit for deciding on the nature of the problem if they identified the relevant aspect of the story for problem solution—the characters' heights. Logo students were significantly more likely to decide on the nature of the problem than students of either of the other two groups.

In a third study in this sequence, students were again asked to solve problems (Clements & Merriman, 1988). These were specially designed to tap one metacomponent. Students were first allowed to solve each problem without help. If they were unsuccessful, they were given a series of prompts. The fewer prompts they needed, the stronger was that metacomponent.

For example, one problem that required deciding on the nature of the problem was an analogy (which was presented to students in the form of pictures).

boy pulling wagon : girl pushing child on swing

is analogous to

car pulling trailer : (ski lift; bulldozer pushing dirt;
horse pulling cart;
dogs pulling sled)

This type of problem was chosen because young students often use associative instead of mapping relations in analogy tasks. That is, they mis-define the problem. For example, they might associate the car with the bulldozer. If they did make such an error, they were given prompts such as "What kind of problem is this?" or "We give you these two pictures. You need to find what goes here so theat these two (indicate bottom two, globally) go together in the same way as these two (indicate top two)."

It seems that Logo programming has the potential to enhance this ability. A question might occur to you. Why has research on this particular ability been so positive, given the mixed results for "general problem-solving abilities" as reported in the previous column? That is, "What works?" It may be critical to note that each of these successful efforts specifically tailored the Logo environment to encourage students to think about the nature of the problems they were posing and solving. To emulate these successful efforts, teachers can:

•Give students the responsibility for designing many of their own Logo projects. Ensure that they accept this responsibility and follow through on their ideas.

•Challenge students to analyze and compare a wide variety of problem types, in and out of Logo. In Logo, for example, help students recognize and use larger units of programming code, such as several procedures or lines of code that work together to achieve a purpose. Often called programming templates, they can be thought of as re–useable patterns of code (see Harvey, 1985). One familiar template is

```
TO procedure  :input
IF EMPTYP :INPUT [STOP]
do.something.to FIRST :INPUT
procedure  BUTFIRST :INPUT
END
```

Help students analyze and compare the problems such templates can help solve. Do similar procedures ever apply outside of Logo?

•Help students approach problems confidently, in varied, appropriate ways. For example, sometimes an exploratory approach is useful to get a feel for a Logo problem. At other times, students need to develop a clearer picture of what their goal is (at these times especially, "deplanning"—changing one's goal and thus the nature of the problem—should be discouraged).

•Discuss problem-solving processes such as deciding on the nature of the problem explicitly. In one of the studies we discussed, students were introduced to "homunculi" (Clements & Merriman, 1988). This set of cartoon characters represented the metacomponential processes. Specifically, "The Problem Decider" was a person thinking about what a problem means (via a "think cloud"). The Problem Decider often asked questions such as, "What am I trying to do?" and "How do the parts of the problem fit together?" These homunculi were introduced as a part of the Logo–programming/problem–solving process. Teachers used the homunculi to describe processes one had to engage in to solve many types of problems.

•Welcome and encourage the curiosity of students—the questions they generate and the problems they pose for themselves.

•Have students work in small groups. Urge them to try to take the point of view of the other members of the group.

•Use such questions and prompts as:

1. In your own words, what is the problem?
2. Explain what you are trying to do to a friend who knows Logo.
3. Describe the Logo procedure you're trying to write so that a friend could solve it themselves.
4. Do you remember what the IF command was for?
5. What kind of procedure will you need to use?
6. Do you need more information to solve your problem?
7. Is this problem similar to any other problems you have solved in Logo before?
8. Remember your _____ Logo project? How did you decide what the problem was then?
9. What made you think to look for a pattern in all the REPEAT statements?

These suggestion are usually just elaborations of those described as the "first phase of teaching problem solving" in a previous column. We'll move into the second phase next month.

References

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

Clements, D. H., & Battista, M. T. (1988). *Progress report: The Development of a Logo–based elementary school geometry curriculum.* Project funded by the National Science Foundation. SUNY at Buffalo/Kent State University. (NSF Grant No.: MDR-8651668)

Clements, D. H., & Merriman, S. L. (1988). Componential developments in Logo programming environments. In R. Mayer (Ed.), *Teaching and Learning Computer Programming: Multiple Research Perspectives* (pp. 13-54). Hillsdale, NJ: Erlbaum.

Clements, D. H., & Nastasi, B. K. (1988). Social and cognitive interactions in educational computer environments. *American Educational Research Journal*, 25, 87-106.

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

Harvey, B. (1985). *Computer science Logo style: Intermediate programming.* Cambridge, MA: MIT Press.

Horton, J., & Ryba, K. (1986). Assessing learning with Logo: A pilot study. *The Computing Teacher*, 14(1), 24-28.

Global News

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

Logo Exchange Continental Editors

Africa	Asia	Australia	Europe	Latin America
Fatimata Seye Sylla	Jun-ichi Yamanishi	Jeff Richardson	Harry Pinxteren	Jose Valente
Lab Informatique et Ed	Faculty of Education	School of Education	Logo Centrum Nederland	NIED
BP 5036 Dakar	Toyama University	GIAE	P.O. Box 1408	UNICAMP
Senegal, West Africa	3190 Gofuku	Switchback Road	BK Nijmegen 6501	13082 Campinas
	Toyama 930 Japan	Churchill 3842	Netherlands	Sao Paulo, Brazil
		Australia		

This month finds many short Logo notices reaching my Virgin Islands address which I will pass on to *LX* readers. In addition, the column will include a report on an extensive Logo project being conducted in Recife, Brazil.

•The State University of Gent, Belgium is organizing the second European Logo Conference, EUROLOGO '89 to be held on August 30 and September 1 of this year. Major topics will include: classroom experiences, Logo projects, research, teacher training and new developments. Presentations, workshops, poster sessions and demonstrations will be held.

If last year's EUROLOGO in Ireland was an indication of this conference's quality, EUROLOGO II will be very worthwhile indeed. If you are traveling to Europe this summer, you may want to be in Gent on August 30! If you wish to participate or attend please write to:

State University of Gent
Department of Education - EDIF
EUROLOGO '89
H. Dunantlaan 1
B-9000 GENT, BELGIUM

•There is an excellent video tape featuring children learning with robot floor turtles that has recently been produced in Australia. The title of the video is *What the Turtle Taught Us*. *LX* readers interested in obtaining a copy should write to:

Brian Doig
Philip Institute of Technology
Alva Grove, Coburg Victoria 3058 AUSTRALIA

•Children's Computer World of Shanghai, China continues to shine in spreading Logo throughout the world's most populous country. Ling Qi-Yu informs the *Logo Exchange* that many Chinese are now using a new version of Terrapin Logo in which one can use Chinese characters as names for procedures, variables, strings, and error messages (and almost everything else). Mr. Ling sent along a sample Logo program that helps students learn new verbs and nouns and then use them in sentences. *LX* readers interested in Logo or computer education in China should contact:

Mr. Ling Qi-Yu
Children's Computer World
China Welfare Institute
157 Changshu Road
Shanghai, China

•Michael Moynihan, the chairman of the Computer Education Society of Ireland, informs the *Logo Exchange* that Ireland is gradually evolving from the beginner's level. There is an increasing number of elementary schools using Logo as well as a small group of secondary teachers; the language is being emphasized in teacher training colleges and in ten Logo training centers located throughout Ireland. The Irish are very proud of the fact that their country produced the winner in the Elementary Logo Division of the International Computer Problem Solving Contest in both 1987 (John Farragher of Limerick) and 1988 (Anne Chazarreta of Cork). For further information about Irish Logo activity please write to:

Mr. Michael Moynihan
Coláiste an Spiroaid Naoimh
Bishopstown, Cork IRELAND

•Another international conference of interest to *LX* readers is entitled "*Children in the Information Age.*" This UNESCO and IFIP sponsored conference will be held in Sophia, Bulgaria from May 20 to 23. Logo and problem solving will be a featured topic. For further information please contact the conference chairman and avid Logo enthusiast :

Acad. Blagovest Sendov, Chairman
29 Aksakov Street
Sofia 1040 BULGARIA

•The Annual Chilean Logo Conference will take place from March 27-29 in Santiago. Contact:

IV Congreso Internacional Logo
Av. Vicuna Mackenna 79
2º Piso, Depto. 201
Santiago, CHILE

The Logo Community in the Northwest Part of Brazil

Submitted by
Paulo Gileno Cysneiros
Project EDUCOM
Centro de Educaçao da Universidade Federal de Pernambuco
Cidade Universitária
50.739 - Recife - PE, BRAZIL

As was mentioned in last September's *LX*, one of the centers of the EDUCOM project is located in the northeast city of Recife (population 2 million). The city presents complex social problems, where luxurious buildings by the seashore co-exist with slums in flooded areas. The educational system is formed by private schools for the well-to-do and public schools for the lower income classes.

The EDUCOM project is part of the Center of Education of the Federal University of Pernambuco. EDUCOM is made up of an interdisciplinary group of researchers, graduate and undergraduate students of several departments, and secondary school teachers of the public school system. The work with Logo started in 1985 and was developed by a group of people studying Papert's *Mindstorms* and using Apple Logo. The idea was to study Papert, to do Logo, and to develop a Logo methodology adapted to the local culture. Today the Logo work consists of specific research projects, teacher training programs, and development of Logo material.

The first research project was the study of a group of students using Logo and was conducted by Meira (1987). He studied the influence of Logo programming on the development of the concept of angle by comparing a Logo-using group with a control group of children. He found that: (1) the Logo programming styles were heavily marked by the use of routines constructed with low sophistication; (2) there were three levels of development in utilization of rotational commands — labeling, intermediate, and additive coordination of angles; (3) constructions of the "rotation theorem" during debugging did not stimulate subjects to use more sophisticated strategies; and (4) the concept of angle experienced in Logo was generalized to other situations.

This investigation was continued by Magina and Falcão (1988). They studied the use of basic Logo primitives (movement and rotation of the turtle) by 113 students from 5th to 8th grades of a public school. The authors were interested in the comprehension of the concept of 90 degree angle, usually assumed as the angle between a horizontal and a vertical line. The results showed that the students' Logo activities were not sufficient to eliminate their orthogonal shape misconception. However, the results of this study were important to the design of other studies about angles. Today Magina is working on her master's thesis on this subject. She is using a computer game with 8 and 9 year old children to study their capacity to estimate angles.

Another research project was conducted by Falcão (1987) who studied the relevance of computer usage on the acquisition of the concept of force. He used diSessa's Dynaturtle with high school students. The results showed that the study group had a better performance with respect to selection of strategies although there was almost no change in the comprehension of the concepts involved in the activities.

Logo has been used to train public school teachers and in computer literacy courses for 5th to 8th grade students from public schools (Cysneiros and Magina, 1988). There is a great demand for this type of activitiy and more teachers and students want to know about computers.

Another important activity of EDUCOM of Pernambuco is the development of Logo materials. They are producing and disseminating materials about basic and advanced subjects in Logo for the MSX computer. The objective of this material is to try to change the widespread view that Logo is a computer language for children's playing and making drawings only. One member of the Logo group is working on a book that starts with the concept of a list, instead of turtle graphics. This book is being used in a course for school teachers. They intend to publish it sometime this year.

References

Cyseiros, P. and Magina, S. (1988) O Ensino de Informática na Escola de Primeiro Grau de Pernambuco. Non published research report.

Falcão, J.T. (1987) Da Física Intuitiva á Dinâmica Newtoniana: Relevância de Simulações em Computador como Auxiliares Instrucionais. Non published master's thesis.

Magina, S. and Falcão, J.T. (1988) Ensino "Curricular de Logo na Escola Pública: Concepção Ortogonal de Angulo Reto. Submitted to publication in the Revista Brasileira de Estudos Pedagógicos, INEP MEC, Brasília, DF.

Maira, L.L. (1987) Geometris em Ação na Programação em Logo. Non published master's theses.

Statement of Ownership Management and Circulation (Required by 39 U. S. C. 3685.) 1A. Title of Publication: Logo Exchange. 1B. Publication no: 02789175. 2. Date of filing: 12/16/88. 3. Frequency of issu: once a month, 3A. No of issues published annually: 9 . 3B. Annual Subscription Price: $24.95. 4. Complete mailing address of known office of publication (not printers): 1787 Agate St., Eugene, OR 97403, 5. Complete mailing address of the headquarters of general business offices of the publisher (not printer): 1787 Agate St., Eugene, OR 97403. 6. Full names and complete mailing address of publisher, editor, and managing editor. Publisher: International Council for Computers in Education, 1787 Agate St., Eugene, OR 97403, Editor: Sharon Yoder, 1787 Agate St., Eugene, OR 97403, Managing Editor: Anita Best, 1787 Agate St., Eugene, OR 97403. 7. Owner: International Council for Computers in Education, 1787 Agate St., Eugene, OR 97403. 8. Know bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities: none. 9. For completion by nonprofit organizations authorized to mail at special rates: Has not changed during preceding 12 months. 10. Extent and nature of circulation. A. Total no of copies: Average no. of copies each issue during preceding 12 months: 3033; actual no. copies of single issue published nearest to filing date: 3000; B. Paid Circulation: 1. Sales through dealers and carriers, street vendors and counter sales: none; 2. Mail subscription: Average 2060; Actual 1,426. C. Total Paid Circulation (Sum of 10B1 and 10B2): Average: 2060; Actual: 1426. D. Free distribution by mail, carrier or other means samples, complimentary and other free copies: Average: 500; Actual: 1,500. E. Total distribution (Sum of C and D): Average 2,560; Actual 2,926. F. Copies not distributed. 1. Office use, left over, unaccounted, spoiled after printing: Average: 473; Actual: 74. 2. Returns from News Agents: none. G. Total (Sum of E, F1 and 2 — should equal net press run shown in A) Average: 3,033; Actual: 3,000. 11. I certify that the statements made by me above are correct and complete: Tasnee Fry, Bookkeeper, ICCE.

Work and Play in the Pacific Northwest this Summer:

Increase Your Leadership and Logo Skills and Enjoy a Week in Scenic Oregon!

Attend these ICCE workshops:

Leadership Development

Topics:

- Long-range planning for computers in schools
- Effective inservice and staff development
- Computer/brain/problem solving
- The computer coordinator
- Frontiers and the future
- Grant writing and other writing for publication

Staff: Dave Moursund and Sharon Burrowes Yoder
Dates: July 9 - July15, 1989
Location: Eugene, Oregon
Cost: $300 before May 1st; $350 after May 1st; covers 2 quarter-hours graduate credit, materials, & instruction.

Receive a 10% discount if you attend both!

Logo for Leaders: Beyond Turtle Graphics

Topics:

- Planning the implementation of Logo on a building or district level
- Integrating Logo into the curriculum
- Logo and problem solving
- Comparison of versions of Logo
- Beyond the beginning: Logo syntax & grammar
- Hands-on time for development of Logo projects
- Assessing Logo learning
- Writing for publication

Staff: Sharon Burrowes Yoder and Dave Moursund
Dates: July 16 - July 22, 1989
Location: Eugene, Oregon
Cost: $300 before May 1st; $350 after May 1st; covers 2 quarter-hours graduate credit, materials, & instruction.

For more information, contact Dave Moursund or Sharon Burrowes Yoder at the International Council for Computers in Education , 1787 Agate Street, Eugene, Oregon 97403 (503) 686-4414.

NEW FROM ICCE

Teaching Thinking Skills With Databases

Teach with databases in any subject!

Teaching Thinking Skills With Databases, a new step-by-step guide for Grades 3-8, provides detailed instructions and ideas for teaching with databases in *any* subject area.

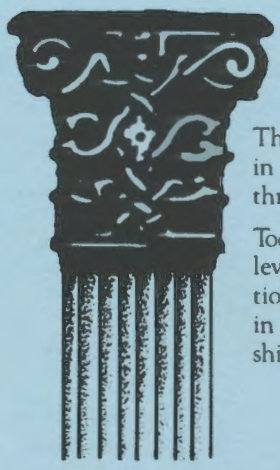
Fifteen steps progress from lower order to higher order thinking skills, and each step is illustrated with scripted lesson plans on the 50 United States. (The states unit serves as a model for any subject area.) **Teaching Thinking Skills With Databases** contains 14 data files on disk and 46 worksheet and transparency masters, all packaged in a three-ring binder.

Available for *AppleWorks*® and *FrEdBase.** Purchase price includes school site license. By Jim Watson. $30 plus $3.50 U.S. shipping prepaid; $5 non-U.S., Alaska, Hawaii, or P.O. Box shipping. ICCE members receive a 10% discount by giving their membership number on the order. Quantity discounts are available. All unpaid orders are charged $2.50 for handling.

For more information, see Jim Watson's article in *The Computing Teacher,* August/September 1988. To order, contact ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905. Ph: 503/686-4414.

*Also available from ICCE for $25 plus $3.50 for shipping.

ICCE



About ICCE

The International Council for Computers in Education was founded by Dr. David Moursund in 1979 as an organization that would foster appropriate instructional use of computers throughout the world.

Today ICCE is the largest professional organization for computer educators at the precollege level. It is nonprofit, supported by 12,000 individual members and more than 50 organizations of computer-using educators worldwide. These organizations are statewide or regionwide in scope, averaging 500 members each. Approximately 84% of ICCE's individual membership is in the United States, 12% is in Canada, and the remainder is scattered around the globe.

About *The Computing Teacher*

ICCE publishes *The Computing Teacher* journal. *The Computing Teacher* provides accurate, responsible, and innovative information for educators, administrators, computer coordinators, and teacher educators. The journal addresses both beginning and advanced computer educators through feature articles, columns, software reviews, and new product and conference listings. Contributors to *The Computing Teacher* are leaders in their fields, consistently providing the latest information in computer education and applications.

Publications, Special Interest Groups

In addition to *The Computing Teacher*, ICCE provides a number of publications to computer-using educators. ICCE's Special Interest Groups provide in-depth information for computer coordinators, teacher educators, computer science educators, and Logo-using educators. *C.A.L.L. Digest* is published nine times per year for ESL teachers. ICCE committees address a variety of ethical and practical issues important to the computer-educating community.

Independent Study Courses

ICCE offers graduate-level independent study courses, designed to provide staff development and leadership. These courses have been approved by the College of Education at the University of Oregon and carry graduate credit from the Oregon State System of Higher Education. Participants correspond with instructors by mail.

Write for information and a free catalog today!

ICCE • University of Oregon • 1787 Agate St. • Eugene, OR 97403 • Ph: 503/686-4414