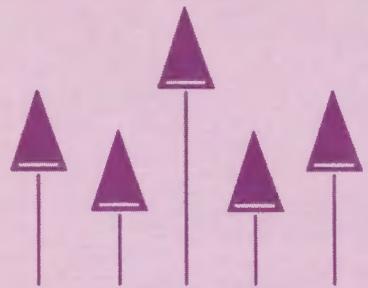
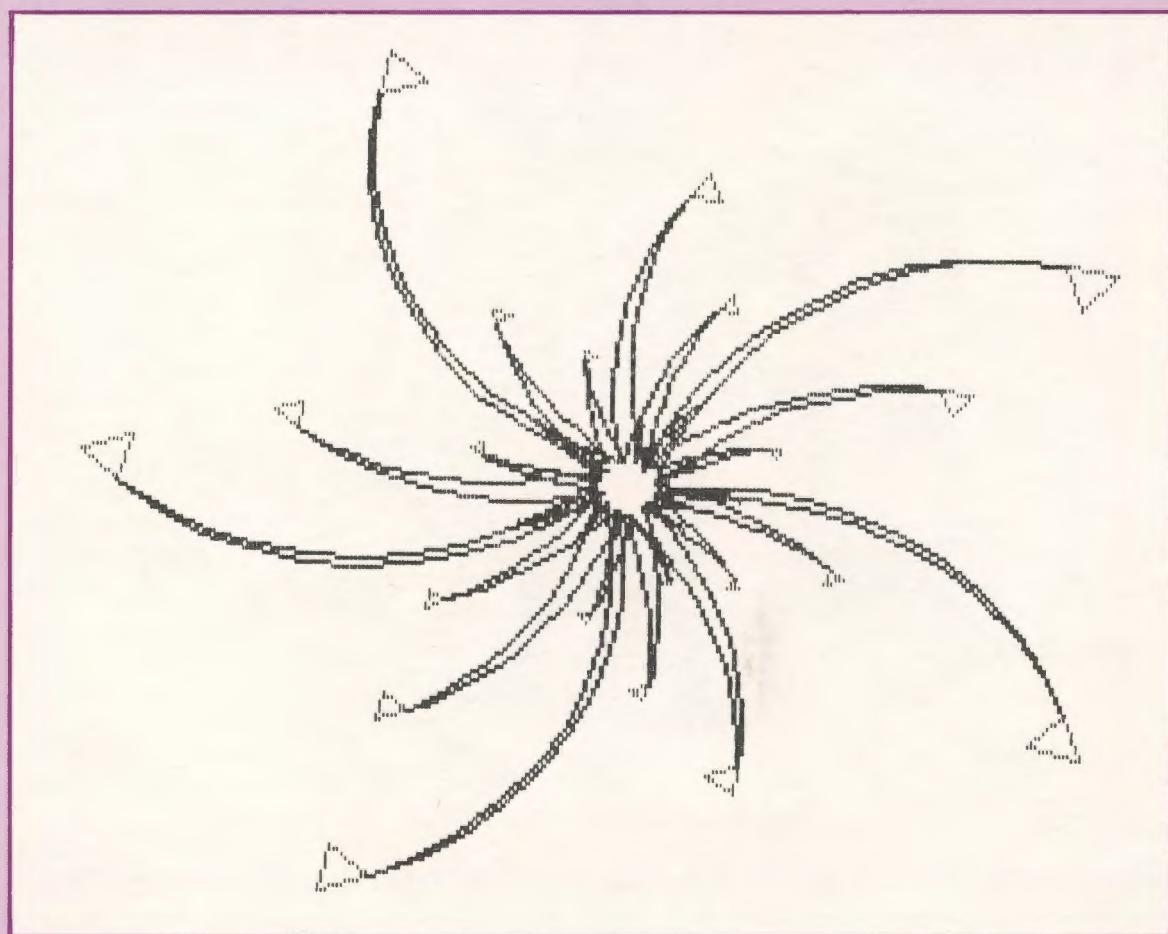

Journal of the ICCE Special Interest Group for Logo-Using Educators



LOGO EXCHANGE

NOVEMBER 1987

VOLUME 6 NUMBER 3



International Council for Computers in Education

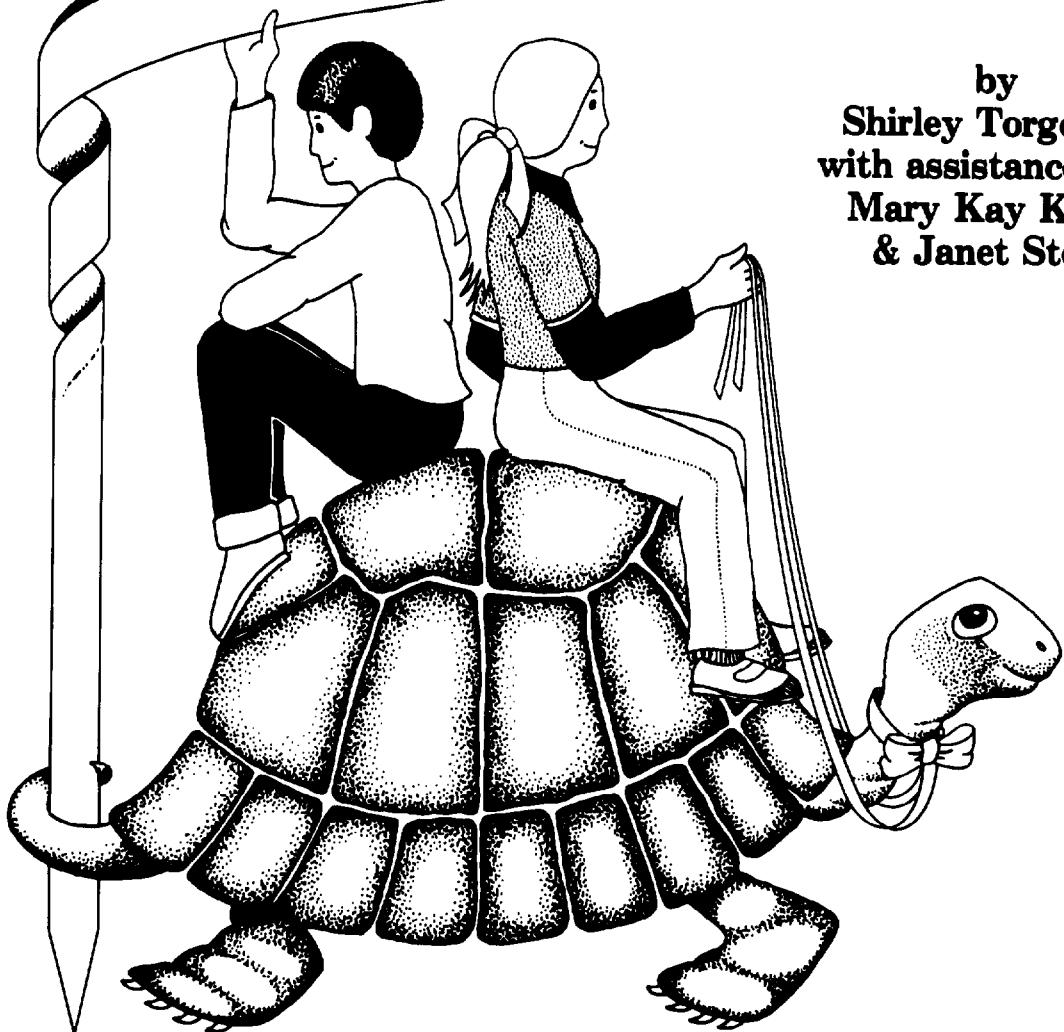


Publications

Logo

in the Classroom

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



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

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

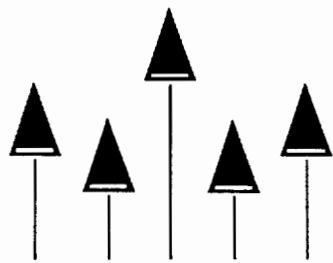
\$13.00 (US) plus \$2.50 shipping.



University of Oregon

1787 Agate St.

Eugene, OR 97403



LOGO EXCHANGE

VOLUME 6 NUMBER 3

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

NOVEMBER 1987

Founding Editor

Tom Lough

Editor-In-Chief

Sharon Burrowes

International Editor

Dennis Harper

Field Editors Eduardo Cavallo

Patricia Dowling

Anne McDougall

Richard Noss

Fatimata Seye Sylla

Hillel Weintraub

Contributing Editors

ASTROLUG

Eadie Adamson

Gina Bull

Glen Bull

Doug Clements

Bill Craig

Sandy Dawson

Judi Harris

Rob Muir

Barbara Randolph

Linda Sherman

Gary Stager

Managing Editor

Anita Best

Special Interest Group Coordinator

Keith Wetzel

Advertising Director

Kathleen Geygan

Publisher

International Council for
Computers in Education

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

POSTMASTER: Send address changes to Logo Exchange, UofO, 1787 Agate St., Eugene, OR 97403.

CONTENTS

From the Editor	Sharon Burrowes	2
Monthly Musings	Tom Lough	3
Stager's Stuff	Gary Stager	4
MathWorlds	A.J.(Sandy) Dawson, editor	7
Logo LinX	Judi Harris	12
Logo Ideas	Eadie Adamson	15
Technical Tip	Eadie Adamson	17
Teaching Tools - The Ins and Outs of ASCII	Glen L. Bull and Gina L. Bull	18
LXionary	Bill Craig	23
The Adventures of Jacques and Elsie	Linda Sherman	24
Book Review	Marie Balet and Lou DeLucia	25
IntLXual Challenges	Rob Muir	26
Logo Pals	Barbara Randolph	27
LX International Logo News	Dennis O. Harper	28

SIGLogo Membership (includes the Logo Exchange)

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

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

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

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

From the Editor

Surprise! You are now reading the third issue of the *Logo Exchange* for this academic year. If you are a regular subscriber, you no doubt thought this would never happen. Sometime in September, you probably began to wonder if those rumors of the demise of the *Logo Exchange* were true. Then, on a lovely fall day in October, you began to think about the money you spent on your subscription that you could spend on something else. Should you complain...but to whom? And when you were just about to become quite annoyed, an issue arrived....and then another...and then another!

Now that things are closer to being on track, I want to offer an apology to all of you. All of us appreciate your patience and support and we hope it was worth the wait. Switching publisher and editor turned out to be more time consuming than anticipated.

Again this month we welcome a new column, Logo Ideas. The columnist, Eadie Adamson, is the computer coordinator at Allen Stevenson

School in New York City where she works with students in grades 4 through 9. Eadie first learned Logo in 1983 when she was asked to join the computing department at her previous school. She received an Impact II grant which was used to buy a floor turtle (which still lives!) to use with kindergarten children. Now in her third year at Allen Stevenson, Eadie is working on a Masters degree in Technology and Media at Teacher's College, Columbia University.

I know you will enjoy Eadie's contributions to LX. She is an enthusiastic teacher and delightful writer. You will no doubt find many ideas to try in your own environment in Eadie's column in the coming months.

Now that you should be getting regular issues of the *Logo Exchange*, let us hear from you. Many of the columnists would welcome your input. I certainly would appreciate any suggestions you have for LX as a whole. And, of course, I'm always looking for articles or pictures for the cover! ▲

Cover: Clark Walmer is in the eighth grade at East Avenue Middle School in Livermore, California. He has been studying Logo with Steve Sesko since he was in fourth grade. This pattern was part of his presentation at the West Coast Logo Conference in February, 1987. The design is based on the Fibonacci sequence: the sides of the triangles are Fibonacci numbers, the increase in the angle of curve is accomplished by successive Fibonacci numbers, and the number of arms of each size is a Fibonacci number.

Monthly Musings

by Tom Lough

Thinking About Thanking

November has a special holiday for school children in the USA. Thanksgiving is popular not only because it means a couple days out of school. It also stands for an important idea.

It might be worthwhile for us to take a moment to think about thanks. Saying "Thank you" is near the top of the list of common courtesies. Countless parents and teachers have drilled the idea into the heads of their young charges, while trying to model the idea as well.

Few activities are more powerful than those in which the teacher or parent act as models. If our students see us saying "Thank you" often, it will have a positive influence on them. Maybe we can even step up our rate of thanks during November.

If we each look about us, I'm sure we can find countless opportunities to say "Thank you" to those around us. I wonder how long computer coordinators, parent volunteers, custodians, and principals have to wait between "thank you's"? How long has it been since we said "thank you" to our students, just for being there? What about a "thank you" to teaching colleagues for their help and support?

Let's think about another level of "thank you." I'd like to suggest that we express our appreciation to the individuals and companies who have made it possible for use to have Logo. During November, why not send a thank you note to Seymour Papert, to Logo Computer Systems Inc., and to Terrapin Inc.? After all, they have put a lot of time and effort into making sure you have the best Logo versions possible.

But why stop with just yourself? How about

giving your students the opportunity to express their appreciation for Logo also? With a little encouragement, I'd bet that they could design some very creative correspondence to send.

I'd like to close with a special "thank you" of my own. Thanks to Anita Best of the International Council for Computers in Education and to Sharon Burrowes for helping to set up the Logo Special Interest Group, SIGLogo, and for the excellent effort in continuing the Logo Exchange.

FD 100!

Tom Lough
PO Box 5341
Charlottesville, VA 22905

Send your "thank you's" to:

Prof. Seymour Papert
MIT E15-309
20 Ames Street
Cambridge, MA 02139

Michael Quinn, President
Logo Computer Systems Inc.
1000 Boulevard Roche
Vaudreuil, Quebec J7V 2B3

J. Sheridan McClees, President
Terrapin Inc.
376 Washington Street
Malden, MA 02148



Stager's Stuff

by Gary Stager

Let's Make a Rebus or "Uncle Rebus"

Making a Logo rebus is a fun activity for students regardless of their age or Logo ability. (A rebus is a representation of a word or phrase by pictures, letters, numbers, or a combination of these to suggest the syllables or words — like in the game Concentration.) It is an especially appropriate project for students who haven't yet tackled the concept of creating procedures. Rebuses are best created in LogoWriter because of the ability to easily combine text and graphics and to stamp user-defined shapes.

Creating a LogoWriter rebus is an excellent activity for developing language arts skills; vocabulary, sentence structure, phonics, symbolic representation, as well as for providing practice in a number of LogoWriter techniques: word processing, defining shapes, stamping shapes, moving the turtle.

The first part of this article will suggest a set of steps for creating a simple LogoWriter rebus and the second part of the article will explore a way in which you might "teach" LogoWriter to create its own rebus.

How to Make a Rebus

- 1) Start with a NEWPAGE.
- 2) Name the LogoWriter page with NAMEPAGE "pagename"
- 3) Be sure the graphics and text on the page are cleared with CG and CT.
- 4) Press APPLE-U* to wake up the cursor on the page so that you may type on the page.
- 5) Type your desired text until you wish to insert a shape (a picture). Press the space bar.
- 6) Press APPLE-D* to return to the command center.
- 7) Be sure the turtle is showing with ST.
- 8) Select the the turtle shape you want to insert and type SETSH # to change the turtle's costume to that shape.

(If the shape does not exist, type SHAPES and design the desired shape on the flip-side (APPLE-F**) of the SHAPES page. You can also draw your

picture with turtle graphics if you don't want to be limited by the size of a turtle shape.

- 9) Use the Turtle-Move keys APPLE-9* and move your shape (turtle) to the desired location on the page (where you left a space in the text).
- 10) Press ESCAPE
- 11) Type PD STAMP to leave the turtle's impression on the screen. If you made a mistake, type PE STAMP PD and select a new shape (using SETSH #) or move the turtle using Turtle-Move.
- 12) Press APPLE-U* to continue typing your rebus text on the page.
- 13) Repeat steps 4-12 until your rebus is complete. Be sure to save your page by pressing ESCAPE!

Notes:

* On the IBM PC or PC Jr. replace the APPLE key with the function key. On the Commodore, replace the APPLE key with the COMMODORE key.

** On the IBM PC and PC Jr., replace APPLE-F with CTRL-F to flip the page to the flip-side and back again.

Project Extensions

In a paper I wrote entitled, "The Literate Turtle" (1986 - Gary S. Stager, LH Publications, Wayne, NJ), I proposed the possibility of a literate turtle that could not only draw and measure, but read and write. Such a turtle would be smart enough to read words and react with them. When the turtle "reads" certain vocabulary words, it might provide a response; draw a picture; create an animation; look for data on another page or perform any number of other tasks. In this article our goal is to create a turtle smart enough to read a short story and create a rebus out of it automatically.

The following are a few tool procedures which can be provided to students so that they can design their own automatic rebuses. Studying the short procedures and their accompanying explanations provide some insight into some of LogoWriter's text processing primitives.

Let's state our problem before we begin to solve it. We want to be able to have the turtle look for a word on our page and replace the word with a particular shape.

The first thing we need to do is to write a procedure with two inputs, :WORD and :SHAPE.NUMBER.

```
TO PUT :WORD :SHAPE.NUMBER
TOP
SEARCH :WORD
IFFALSE FOUND?
  [STICK.IN :SHAPE.NUMBER]
  [TOP STOP]
PUT :WORD :SHAPE.NUMBER
END
```

TOP places the cursor at the TOP of the page.

LogoWriter only searches for words below the cursor.

SEARCH :WORD instructs LogoWriter to find a particular word specified in the variable, :WORD.

IFFALSE is a conditional that requires three inputs; a condition, a TRUE consequence, and a FALSE consequence. In this case we are checking to see if the word we were searching for has been found.

FOUND? outputs TRUE if a word we've searched for has been located in the text and FALSE if it hasn't been found.

If the word we were searching for was found, the procedure STICK.IN, shown below, is executed. The number in the input name :SHAPE.NUMBER is passed to STICK.IN so that shape can be stamped in place of the word in the STICK.IN procedure. If the word is not found by SEARCH, the cursor is reset to the top of the page and the procedure stops. After the word was found and the STICK.IN procedure was run, the PUT procedure calls itself and looks for another occurrence of the particular word it just replaced.

```
TO STICK.IN :SHAPE.NUMBER
CUT
REPEAT 5 [INSERT CHAR 32]
```

```
REPEAT 3 [CB]
PU
SETSH :SHAPE.NUMBER
SETPOS CURSORPOS
PD
STAMP
END
```

STICK.IN is a subprocedure to PUT which requires one input, :SHAPE.NUMBER, the number of the shape you wish to stamp.

CUT removes the word found by SEARCH and places a copy of it in the LogoWriter clipboard.

REPEAT 5 [INSERT CHAR 32] inserts 5 spaces in place of the word we cut to make room for the shape to be stamped. INSERT places a word or list in the page without a carriage return.

REPEAT 3 [CB] tells the cursor to move backwards three spaces so that it is near the center of the space we created in the line above. CB stands for cursor back and has the same effect as pressing the left cursor arrow.

PU lifts the turtle's pen up.

SETSH :SHAPE.NUMBER sets the turtle's shape to the number in the variable :SHAPE.NUMBER.

SETPOS CURSORPOS tells the turtle to move to where the cursor is. SETPOS instructs the turtle to set its position to a particular point on the screen and requires a list of the X and Y coordinates as input, for example [25 50]. CURSORPOS reports a list of coordinates representing where the cursor is located.

PD puts the turtle's pen down.

STAMP leaves the turtle's impression on the screen.

The following is an example of how the automatic rebus maker works:

```
TO REBUS
TEXT
DOIT
END
```

```
TO TEXT
CT           ← cleartext
RG           ← reset graphics
PRINT []     ← print a blank line
PRINT [This is a story about a boy
      named]
PRINT [Bill.]
PRINT []
PRINT [Bill has a rabbit and a cat.
      He lives]
PRINT []
PRINT [in a house by an oak tree.]
END
```

```
TO DOIT
PUT "boy 15
PUT "rabbit 22
PUT "cat 21
PUT "tree 23
PUT "house 20
END
```

REBUS is the superprocedure that makes everything work. TEXT prints a short story on the page. For purely cosmetic reasons I printed blank lines so that the text is double spaced. DOIT uses the PUT procedure we created above and tells it which words to replace with which shapes (the shape numbers may not be the same on your disk if you have changed the shapes page - if your story comes out really silly, this is probably the cause). You do not have to use the PUT procedure in any particular order. It will replace a word with a shape as many times as it appears in any order.

I would love to see copies of rebuses created by you or your students. Send them to the address listed below.

—Gary S. Stager is the Director of Training for the Network for Action in Microcomputer Education in New Jersey and the Vice President of SIGLogo. He may be reached at 12 Locust Place Wayne, NJ 07470 or CIS #: 73306,2446 APPLELINK #:K0331.

Terrapin Sponsors a Logo Contest for Innovative Teachers

Terrapin, Inc. is sponsoring a contest for educators who have created innovative ways of using Logo in the classroom. Entry topics may include creative ways of integrating Logo into the curriculum, methods of demonstrating abstract concepts in a concrete manner, Logo games that make learning fun, and ways of creating an awareness of Logo among colleagues. Three Grand Prize winners will each be awarded their choice of \$500 worth of Terrapin Logo software curriculum materials. Five runners-up will receive Terrapin's new Logo Data Toolkit. In all, \$1750 worth of Terrapin products will be awarded.

Terrapin has extended the deadline for *Logo Exchange* readers. Please send your entries in immediately (by the end of February). The spring issue of Terrapin's Logo newsletter, *The Terrapin Times*, will identify the contest winners and feature their innovative ideas so that others may benefit.

Educators may enter by submitting a one to two page description with a minimum of the following information: a brief description of the project, the goal or objective, the appropriate age group or skill level, the subject area, the length of time required to implement it in a classroom, any special materials or additional hardware required, and the way in which it is tied to the overall curriculum or particular subject area. If lengthy Logo procedures are an integral part of the project, a disk of these procedures should be included.

Entries will be judged on originality, ease of implementation, simplicity of use, and opportunity for individual and/or group exploration and discovery.

Entries will be judged by the educational staff of Terrapin, Inc. To enter or obtain more information, contact Ann Ciara, Terrapin, Inc., c/o Logo Innovations, 376 Washington St., Malden, MA 02148, ph:617/322-4800.

MathWorlds

edited by
A. J. (Sandy) Dawson

When I attended the National Council of Teachers of Mathematics Annual General meeting in Anaheim last April, I had the pleasure of attending a number of Logo and mathematics sessions. Two come to mind immediately: one by Johnny Lott, and the other by his colleague and co-author, Rick Billstein. These two gentlemen have, along with a third colleague, Shlomo Libeskind, collaborated on a number of books, one of which, *MIT Logo for the Apple*, gives readers a start on the topic of this column: namely, tessellations. I was very pleased when

Johnny Lott agreed to write a guest column for MathWorlds. I was delighted when I received his material and found it to be so engaging. Like many others I have long been a fan of Escher drawings, and I continue to marvel at the inventiveness and creativity of the man in producing the drawing he did. Now Johnny Lott has brought those ideas to the computer, and provides us all with ideas as to Logo procedures we can use and modify to create our very own Escher-type drawings. Make that Turtle tessellate, folks!

Escher-like Logo-type Tessellations

by
Johnny W. Lott

Maurits C. Escher (1898-1972), a master of tessellations, was strongly influenced by the work of Moorish artists. With little or no formal training in mathematics, his work has often been depicted as being as close to mathematics as to art. "What is a tessellation?" you ask. A tessellation is a complete covering of a plane (or monitor screen) by one or more figures in a repeating pattern with no overlapping of the figures. About tessellations, Escher said (Macgillavry, 1965):

In the course of the years I designed about a hundred and fifty tessellations. In the beginning I puzzled quite instinctively, driven by an irresistible pleasure in repeating the same forms without gaps, on a piece of paper.

Logo, with recursion and turtle graphics, provides a natural environment to try to create Escher-like tessellations. Given the set of procedures below, a monitor screen can be tessellated with an equilateral triangle, a square, and a regular hexagon. Specifically the set of procedures given will tessellate the screen with squares using -120 and 120 as boundaries for x-coordinates of the squares and -100 and 100 as boundaries for the y-coordinates of the squares. The main procedure is WALL which requires inputs for the coordinates of the point at which the first square is to have a lower left vertex and an input for the length of a side of that square. (All procedures in this paper are written in Terrapin Logo).

```

TO WALL :XPT :YPT :S
SETUP :XPT :YPT
WALLPAPER :YPT :S
END

TO SETUP :XPT :YPT
PENUP
SETXY :XPT :YPT
PENDOWN
END

TO SQUARE :S
REPEAT 4 [FORWARD :S RIGHT 90]
END

```

```

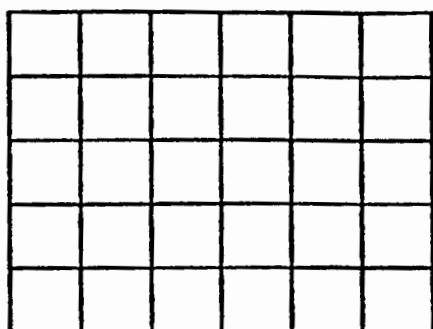
TO WALLPAPER :YPT :S
SQUARESTRIP :S
PENUP
SETUP (XCOR + :S) :YPT
PENDOWN
WALLPAPER :YPT :S
END

TO SQUARESTRIP :S
IF XCOR + :S > 120 TOPLEVEL
IF (ANYOF (XCOR - :S < -120)
    (XCOR + :S > 120) (YCOR - :S < -100)
    (YCOR + :S > 100)) STOP
SQUARE :S
FORWARD :S
SQUARESTRIP :S
END

```

(Note: The "IF (ANYOF...)" line in the above procedure must be typed without pressing Return.)

A sample printout which shows a tessellation with squares drawn using the above procedures is given in Figure 1.



WALL -60 (-60) 30

Figure 1

To create Escher-like tessellations, the necessary clues are given by Ranucci and Teeters in *Creating Escher-type Drawings*, 1977. One possibility is to create a tessellation using deformations of sides of a regular polygon and a translation to complete the drawing. The procedures given above draw a square (SQUARE :S) and use a translation (FORWARD :S), and then the strip of squares is repeatedly drawn (WALLPAPER :YPT :S).

To create a Logo-type tessellation utilizing the process in WALL, all that is necessary is to determine how to deform the sides of the squares in such a way that they will fit when translated. Consider Figure 2(a) where a square is drawn, Figure 2(b) where one side of the square is deformed, and finally Figure 2(c) where the deformation is translated to the side of the square parallel to the one where the original deformation took place and attached to that side as shown.

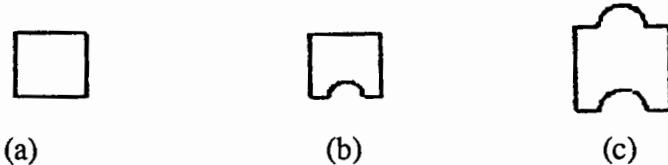


Figure 2

In order to write a set of procedures for a tessellation utilizing the drawing in Figure 2(c), we use the SETUP procedure above, edit other above procedures, and use arc procedures listed below from the ARCS file of the MIT Terrapin Logo Utilities disk as follows:

```

TO WALL1 :XPT :YPT :S
SETUP :XPT :YPT
WALLPAPER1 :YPT :S
END

TO WALLPAPER1 :YPT :S
SQUAREARCSTRIP :S
PENUP
SETUP (XCOR + :S) :YPT
PENDOWN
WALLPAPER1 :YPT :S
END

TO SQUARARCSTRIP :S
IF XCOR + :S > 120 TOPLEVEL

```

```

IF (ANYOF (XCOR - :S < - 120) (XCOR
    +:S > 120) (YCOR - :S < -100)
    (YCOR + :S.>100)) STOP
SQARC :S
FORWARD :S
SQUARARCSTRIP :S
END

TO SQARC :S
FORWARD :S
RIGHT 90
FORWARD :S / 4
LEFT 90
RARC :S / 4 180
LEFT 90
FORWARD :S / 4
RIGHT 90
FORWARD :S
RIGHT 90
FORWARD :S / 4
RIGHT 90
LARC :S / 4 180
RIGHT 90
FORWARD :S / 4
RIGHT 90
END

TO RARC :RADIUS :DEGREES
RIGHT 2.5
RARC1 :RADIUS * 0.017453 :DEGREES
LEFT 2.5
END

TO RARC1 :SIZE :DEGREES
REPEAT QUOTIENT :DEGREES 5
    [FORWARD :SIZE * 5 RIGHT 5]
CORRECTARCR :SIZE (REMAINDER :DEGREES 10)
END

TO CORRECTARCR :SIZE :AMOUNT
FORWARD :SIZE * :AMOUNT
RIGHT :AMOUNT
END

TO LARC :RADIUS :DEGREES
LEFT 2.5
LARC1 :RADIUS * 0.017453 :DEGREES
RIGHT 2.5
END

TO LARC1 :SIZE :DEGREES
REPEAT QUOTIENT :DEGREES 5
    [FORWARD :SIZE * 5 LEFT 5]
CORRECTARCL :SIZE (REMAINDER :DEGREES 10)
END

TO CORRECTARCL :SIZE :AMOUNT
FORWARD :SIZE * :AMOUNT

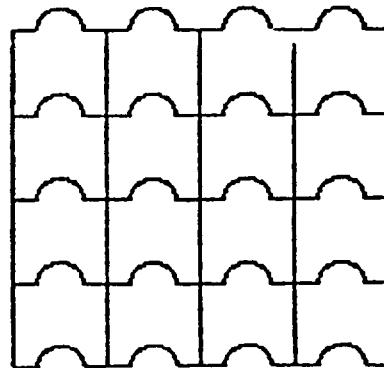
```

```

LEFT :AMOUNT
END

```

Using these procedures, a tessellation similar to the one in Figure 3 may be obtained.



WALL1 -60 (-60) 30

Figure 3

A method of using the equilateral triangle to create a different tessellation is to change the shape by altering a side and rotating the changed side about a vertex to form the third side and a new figure with which to tessellate. Figure 4 shows how this might be done.



Figure 4

The set of procedures for drawing a tessellation with this shape are more like a tessellation drawn with a regular hexagon. Depending upon where the starting and stopping points are, there will appear to be gaps in the tessellation. The interested reader should try these procedures to see that this is not the case. The entire set of procedures for creating the tessellation are given below. SETUP is used from above.

```

TO WALL3 :XPT :YPT :S
SETUP :XPT :YPT
WALLPAPER3 :XPT :YPT :S
END

```

```

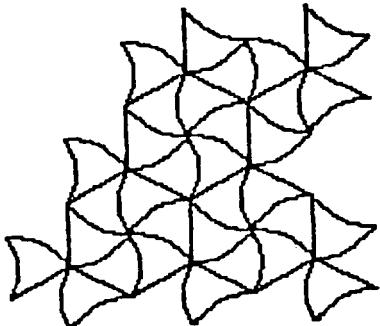
TO WALLPAPER3 :XPT :YPT :S
LOCAL "X"
MAKE "X XCOR
RTRISTRIP :S
PENUP
SETUP (:XPT + :S * (SQRT 3)) :YPT
PENDOWN
WALLPAPER3 :XPT :YPT :S
END

TO RTRISTRIP :S
IF XCOR - :S > 120 TOPLEVEL
IF (ANYOF (XCOR < -120) (XCOR + :S *
(SQRT 3) > 120) (YCOR < -120)
(YCOR + :S * 3 > 120)) STOP
REPEAT 3 [RTRIANGLE :S RIGHT 120]
PENUP
FORWARD :S
RIGHT 60
FORWARD :S
LEFT 60
PENDOWN
RTRISTRIP :S
END

TO RTRIANGLE :S
FORWARD :S
RIGHT 150
LARC :S 60
RIGHT 120
RARC :S
RIGHT 90
END

```

A tessellation drawn using the above procedures is given in Figure 5.



WALL3 -60 (-60) 30

Figure 5

The tessellation in Figure 5 makes use of the altered triangle to draw many figures but also uses what might be called the negative space to complete the picture. What is meant here is that not every

altered triangle in the final picture is traced around. This technique is sometimes helpful especially if the figure to be used to tessellate is complicated and takes some time to draw. The final example to be given here is of that type. It is a takeoff on M.C. Escher's sketch entitled *Study of Regular Division of the Plane with Birds* done in India ink and watercolor in 1955. It uses an altered rectangle as the basic unit of the tessellation. The rectangle is altered on all sides and negative space is used in the final drawing. A grid with the altered rectangle is shown in Figure 6 followed by the procedures used to draw the tessellation and a sample of the tessellation in Figure 7. The arc procedures and SETUP procedure given earlier are also used here but not listed again.

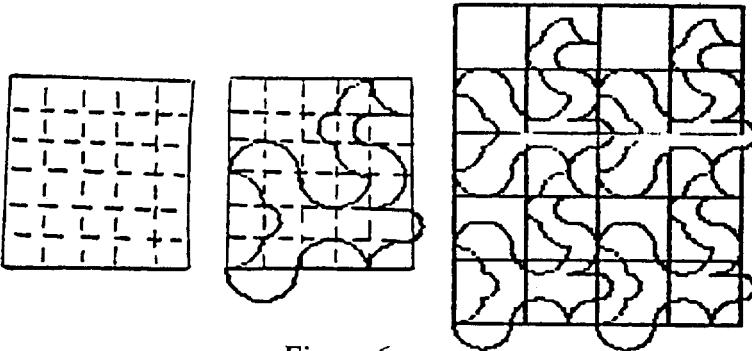


Figure 6

```

TO WALL4 :XPT :YPT :S
GOOSEPAPER :YPT :S
END

```

```

TO SETUP :XPT
PENUP
SETXY :XPT :YPT
PENDOWN
END

```

```

TO GOOSEPAPER :YPT :S
GOOSESTRIP :S
SETUP (XCOR + 5 * :S) :YPT
GOOSEPAPER :YPT :S
END

```

```

TO GOOSESTRIP :S
IF XCOR + 5 * :S > 120 TOPLEVEL
IF (ANYOF (XCOR < -120) (XCOR + 5 *
:S > 120) (YCOR - :S < -120)
(YCOR :6 * :S > 120)) STOP
GOOSE :S
PENUP
FORWARD 6 * :S

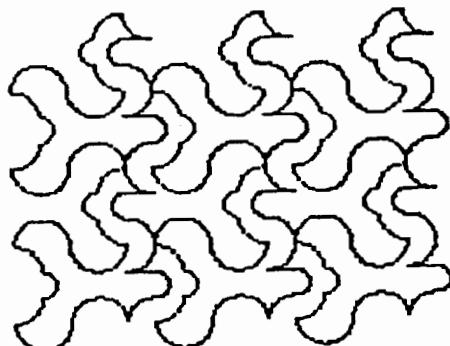
```

```

FORWARD 6 * :S
PENDOWN
GOOSESTRIP :S
END

TO GOOSE :S
RIGHT 90
LARC :S 90
RIGHT 90
LARC :S/2 180
RIGHT 90
LARC :S 90
RIGHT 90
RARC :S 180
LARC :S 180
LEFT 90
RARC :S 90
LEFT 90
RARC :S 90
LEFT 90
RARC :S/2 180
LEFT 90
RARC :S 90
RIGHT 90
LARC :S 90
RIGHT 180
FORWARD :S
LARC :S/2 180
RARC :S 180
RIGHT 180
FORWARD :S
RARC :S/2 180
LARC :S 90
RIGHT 180
LARC :S 180
RARC :S 180
END

```



WALL4 -60 (-60) 12
FIGURE 7

The tessellations given here are but a start and should be used only as ideas for creating your own. Other samples that can be drawn are suggested in *Creating Escher-type Drawings* by Ranucci and Teeters. It is strongly suggested that the interested reader use graph paper, both regular and isometric, to

help in the creation of designs.

Interested readers may contact Johnny Lott at: Department of Mathematics, University of Montana, Missoula, Montana 59812.

Selected Bibliography

Billstein,R., S. Libeskind, and J. Lott. *MIT Logo for the Apple*. Menlo Park: Benjamin/Cummins Publishing Company, 1985.

Billstein,R. S. Libeskind, and J. Lott. *Apple Logo: Programming and Problem Solving*. Menlo Park: Benjamin/Cummings Publishing Company, 1986.

Haak, S. "Transformation Geometry and the Artwork of M.C. Escher," *Mathematics Teacher*, 69 (December 1976) :647-652.

Kenney, M., and S. Bezuszka. *Tessellations Using Logo*. Palo Alto: Dale Seymour Publications, 1987.

Macgillavry, C. *Symmetry Aspects of M.C. Escher's Periodic Drawings*. Utrecht: A. Oosthoek's Uitgeversmaatschappij NV, 1985.

Ranucci, E. "A Tiny Treasury of Tessellations," *Mathematics Teacher*, (February 1968) :114-117.

Ranucci, E. "Master of Tessellations: M.C. Escher, 1898-1972," *Mathematics Teacher* 67 (April 1974) :299-306.

Ranucci,E. and J. Teeters. *Creating Escher-type Drawings*. Palo Alto: Creative Publications, 1977.

Teeters, J. "How to Draw Tessellations of the Escher Type," *Mathematics Teacher* 67 (April, 1974) :307-310.

A. J. (Sandy) Dawson is a member of the Faculty of Education at Simon Fraser University in Vancouver, British Columbia. His CompuServe number is 76475,1315.

He can also be reached electronically as Sandy_Dawson@SFU.BITNET or Sandy_Dawson%SFU@um.cc.umnich.edu.



Logo LinX

by Judi Harris

Historical Lessons in Recursion

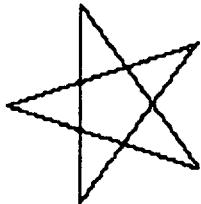
History, with all her volumes vast,
Hath but one page."

—Lord Byron (1812-1818)

You probably have heard the saying, "History repeats itself." Perhaps it would be more accurate to say, "History recurs." What is the difference?

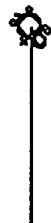
Consider this Logo procedure:

```
TO STAR :SIDE
REPEAT 5 [FD :SIDE RT 144]
END
```



STAR 75

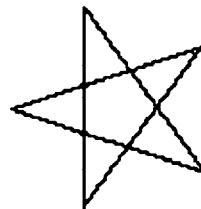
These commands tell the turtle to repeat FD :SIDE RT 144 five times. Each time the line is repeated,



The five parts of the resulting drawing are exactly the same.

Here is another way to tell the turtle to make a five-pointed star. What changes as each recursive copy of RSTAR is executed?

```
TO RSTAR :SIDE :HEADING
FD :SIDE
SETH :HEADING
IF :HEADING = 720 [STOP]
RSTAR :SIDE (:HEADING + 144)
END
```



RSTAR 75 144

STAR contains an iterative structure (REPEAT). RSTAR accomplishes the same result with a recursive structure.

Consider another interpretation of the STAR procedure:

```
TO STARS :SIDE :NUMBER
IF :NUMBER < 1 [STOP]
STAR :SIDE
PU
SETX (XCOR + 20)
PD
STARS (:SIDE + 10) (:NUMBER - 1)
END
```



STARS 45 5

These five stars are similar, but not identical, in appearance. The basic shape is the same, but the size and position are different. The stars above are *recursive copies*; they share a common pattern, but vary in dimension and placement.

Traditionally, the single star drawn with the procedure STAR :SIDE is considered *iterative*. This implies that the commands inside the brackets for the REPEAT statement (FD :SIDE RT 144) are repeated identically each time they are run.

Things Just Aren't the Same

"When two do the same thing, it is not the same thing after all." -

—Publilius Syrus, 1st century B.C.

Outside the realms of mathematics and computer science, there is no such thing as a perfect copy. Sameness is really similarity. Copies are really patterns. (The implications of this idea for Xerox, Inc. are mind-boggling!) History does not repeat itself; it recurs.

Although the academic study of history does not focus upon recurring patterns of events, the human mind, in an effort to comprehend and apply history's lessons, often perceives and codes event sequences into intuitive structures that recur, with modifications, over the centuries.

The evolution of the sonata form, as musical historians have reconstructed it, provides an excellent microcosmic example of an historical pattern. Simply stated, it is A B A.

The first section, or *exposition*, of most late 18th century symphonies, chamber music, and sona-

tas states a musical theme in the tonic key, a second theme in the dominant or relative major key, and a closing theme, also in the dominant or relative major. The second section, or *development*, uses themes or portions of themes first presented in the exposition to modulate to different melodies and keys. The final section of music written in the Classical sonata form is called the *recapitulation*. Here, the material comprising the exposition is restated, but in the tonic key. The exposition and recapitulation were commonly faster in tempo and more major in pitch than the slower, more minor development.

The sonata form itself is doubly recurrent; the exposition's three-part structure is contained within the larger three-part structure of the sonata form. Often, melodic phrases within the three sections of the sonata were also recurrent in pitch orders, modulating to different keys.

This repetitive structure recurred through history. The roots of the structure lay in the Italian opera overture, or sinfonia, which was most popular in the early 1700s. The three-part organization of these pieces can be described simply as fast-slow-fast.

In about 1730, Italian composers (such as Sammartini) began to write separate concert symphonies according to further developments of the A B A format. The sonata form itself was an amalgamation of Baroque and Italian opera structures. Each major composer expressed successive copies of the sonata structure, based upon the copies of the major composers before him. The level of recurrence in just this small example seem infinite.

As Ever, Recursion

Structures which recur can be detected on macrocosmic levels, also. Consider, for example, the changes happening now in our Western society as the use of computers pervades more and more aspects of our lives. This has been called the "Information Age," where access to and processing of information provides economic, political, and personal power. These times have been compared on many levels to

The Idea Lab

The Idea Lab is documentation and set of 6 disks of interactive games written by sixth graders at Ladue Junior High in St. Louis, Missouri. They used LogoWriter to create their games. The games are interesting and can serve as models for you and your students to develop your own interactive software.

Some of the topics included are: Riddles; Explorers; Homonyms; Symmetry; and Organs of the Body. The disks only work in conjunction with LogoWriter for the Apple.

For more information write to:

The Idea Lab
Ladue Junior High
9701 Conway Road
St. Louis, MO 63124

The New Take-Home Pak Gives Students and Teachers a Logo to Go!

Terrapin's new Take-Home Paks are designed for groups of students and teachers who use Logo as part of their curriculum and want an affordable Logo disk to use at home. Each Take-Home Pak is just \$30 per person, with a minimum of ten participants. There is no additional charge for shipping or handling.

The Take-Home Pak comes complete with a Logo Language disk, a disk of Utility programs, a Quick Reference Card and a Warranty Card. By returning the warranty card, the user becomes a registered owner - eligible for upgrades, replacements disks and backup disks. Take-Home Paks are available for either Terrapin Logo for the Apple or Commodore Logo.

The Terrapin Take-Home Pak makes it convenient to use Logo at home to extend and enrich the classroom experience. No longer does Logo exploration have to be confined to a few hours in an over-worked computer lab. The Terrapin Take-Home Pak makes Logo exploration practically limitless! For more information and an order form, contact Terrapin, Inc., 376 Washington Street, Malden MA 02148; ph. 617/322-4800.

the years at the turn of the 20th century, the "Industrial Age," when machines were more concerned with assisting production of goods than assisting the production of knowledge. No, conditions are not nearly the same as they were in 1900, but there are striking similarities.

Is it any wonder, then, that calls for societal and educational reform during the Industrial Revolution sound familiar? Then and now, there were and are concerned action to uphold the rights of minorities and women, determined efforts to standardize the quality of public education at the state level, bold attempts individualize and modernize instruction, and, of course, massive efforts to bring society's new machines into the school environment.

Viewed from a different perspective, recurrent structures that reflect the shape of our collective experience manifest on many levels and in numerous experiential arenas. History can be conceived as a three-dimensional, expanding spiral, rather than as a linear timeline. Events on later arms of the spiral are rooted in, and similar in structure to, earlier events. And, theoretically, it all started with one point somewhere in space and time. All occurrence has its origin in that one point. Each event is but a variation on the first; a unique, but similar recurrent pattern.

What recurrent patterns can you and your students identify in local history? state history? national history? world history? ancient history? art history? philosophical history? political history? family histories? technological history? Logo history?

"The disadvantage of men not knowing the past is that they do not know the present. History is a hill or high point of vantage, from which alone men see the town in which they live or the age in which they are living."

—G.K. Chesterton (1933) ▲

LOGO IDEAS

by Eadie Adamson

Discovering Random

Often experiencing or discovering something is a more effective path to understanding. The intellectual leaps involved are more personal and have a more lasting impact. Teaching Logo offers us a chance to test this theory. The challenge is to apply it even when the lesson might be approached in a more didactic fashion.

That being said, how can one help students discover what RANDOM does in Logo, particularly when used with numbers? I decided to approach RANDOM by asking my fifth grade students to participate in an experiment. I made no attempt to explain RANDOM, although I did try to elicit from them in the beginning what they thought the word meant in order to develop some very general expectations about the experiment.

The Experiment — Stage One

Everyone needed a pencil and a piece of paper by their computer for this experiment. On the board I put a sample of the chart they were to use, leaving space on the left side of the board for later use :

0	1	2	3	4	5	

I asked the students to speculate on what would happen if the computer was given the command PRINT RANDOM 5. Using LogoWriter, we wrote a procedure on the flip side:

TO R
CT

```
REPEAT 10 [PR RANDOM 5]
END
```

(CT, for CLEARTEXT, gets rid of the last set of numbers before generating the next set. You might try R without CT to see why we used it.) Flipping to the front of the page, each student typed R once, counted the number of 0's, 1's, etc., and entered the totals on their record sheet. Each student was asked to generate the ten numbers ten times. By about the fourth time some observed that the number 5 was not appearing. (RANDOM and a number will output a number in the range from 0 to one less than the number itself.) I asked them why they thought this was happening and encouraged them to formulate their own hypothesis and to check it further while they finished the experiment.

By the time this process was completed, each student had a good working definition, based upon his own experience, of random number generation. The experiment also provided the opportunity to talk about accuracy in experiments. I pointed out that for each set of numbers, the total number of one's, two's and so on should add up to ten. If it did not, their information was inaccurate. In a real scientific experiment, they would have to begin again!

The next challenge which I threw out was, "Well, if you can't get a 5 from PRINT RANDOM 5, how can you get a 5?" Everyone's first answer was, "Try PRINT RANDOM 6." That worked, but then I asked, "How would you get 5's but no 0's?" That stumped them, but it was a good chance to discuss how Logo evaluates inputs and to show how to accomplish this task. (PRINT 1 + RANDOM 5 is the answer.) We talked about how one might use this in a game to be sure that a certain minimum distance

was moved or that a minimum number of points could be generated by adding the minimum number to the RANDOM number.

The Experiment — Stage Two

(This part I've skipped recently.)

When this part of the experiment was complete, I asked each student to add up the total number of 0s, 1s, etc. on their record sheet. As each finished this task, we put their totals on the board, placing their initials to the left of their list of numbers. This was a good time to suggest using the computer to get their totals, either using + or SUM. Multiple inputs to SUM require parentheses.

```
PRINT 8 + 6 + 9 + 3      or
PRINT (SUM 8 6 9 3).
```

At this stage, as totals were entered on the board, it began to look like a contest to see whose computer generated the most instances of a given number. I asked them to hold that thought until we had taken the final step. We then shared the task of finding the grand total of appearances of a given number. Again, using SUM was the more effective method of handling this. Amazingly enough, for the students, these totals were very nearly the same for each number! This led us to a discussion of examples of random numbers in use around us; particularly in the New York State Lottery.

The Follow up — Random Ideas

This exercise elicited a number of ideas for future projects. Two interesting projects suggested by my class were a dice game and a school lottery in which the computer would select the numbers! (The last is an challenging problem. How can you be sure to get a series of six numbers with no repeated numbers?)

If you try this experiment, a simple illustration of RANDOM should follow. You might ask

everyone to type FD RANDOM 100 and then look at the different results on everyone's computers, or write a SQUARE procedure using FD RANDOM 100 and compare results.

For some fun, if you're using Logo II, I suggest trying this (found in Brian Harvey's *Computer Science Logo Style*) to see how else RANDOM works:

```
REPEAT 50 [SETCURSOR LIST RANDOM
            36 RANDOM 22 TYPE "HI"]
```

This will prompt all sorts of experimentation with words and sentences. Try placing small filled squares randomly all over the screen (an effective use of Apple Logo II's and LogoWriter's FILL command).

Another illustration of RANDOM, once the concept is understood, is the following:

Write a procedure for a small square and a small rectangle (neither with dimensions larger than 50 or 60). Then write PAINT:

```
TO PAINT
  SQ
  FORWARD RANDOM 80
  RIGHT 90
  REC
  FORWARD RANDOM 50
  RIGHT 90
  END
```

After trying things such as REPEAT 20 [PAINT], make PAINT recursive (add PAINT as the last line of the procedure). Later your students might, as did mine, want to add random colors. Let your students try to figure this one out. How can they eliminate black?

If you are using LogoWriter, the RANDOM experiment connects nicely with the Dice 4 game in the LogoWriter project booklet, *You and Your Friends*, pp. 12 -19. Perhaps your students can make their own version of a dice game once they've completed Dice 4. Also check out the LogoWriter blue

TO GREAT LAKES/EAST COAST LOGO CONFERENCE

PRE-CONFERENCE WORKSHOPS

[Thursday May 5, 1988]

CONFERENCE

[Friday May 6, 1988] [Saturday May 7, 1988]

END



TO CREDITS

Presented by :ECCO [Educational Computer Consortium of Ohio]

Co-sponsors

[Cleveland State University] [University of Virginia Curry School of Education]

[John Carroll University] [Kent State University]

END

MAKE "GUEST SPEAKERS

[Seymour Papert, Dan and Molly Watt, Michael Tempel, Tom Lough,
Brian Silverman, Alison Birch, Michael Battista, Douglas Clements,
Steve Ocko, Sharon Burrowes, Brian Harvey and many more]



EDUCATIONAL COMPUTER CONSORTIUM OF OHIO

1123 S.O.M. Center Road
Cleveland, Ohio 44124

Non-Profit Org.
U.S. POSTAGE
PAID
Cleveland, Ohio
Permit No. 365

TO THURSDAY, MAY 5, 1988

PRE-CONFERENCE WORKSHOPS

TIME [9:00 am - 4:00 pm]

Cleveland State University

CONFERENCE WELCOMING RECEPTION

TIME [6:00 pm - 8:00 pm]

Stouffer Tower City Plaza

(24 Public Square, Cleveland)

[Cash Bar]

END



TO PRE-CONFERENCE WORKSHOPS

[A] [ASSESSING CRITICAL ASPECTS OF LOGO LEARNING]

PRESENTERS :Dan Watt :Molly Watt

This workshop will provide hands-on experience designed to help teachers assess key learning points of Logo learning in the classroom. In order for students to gain maximum benefits from the Logo learning experience, teachers must have the knowledge to assess the student's progress. The key learning points and the accompanying assessment activities used in this workshop have been developed by teachers and researchers collaborating in a research and development effort supported by the National Science Foundation. Topics will be chosen from areas common to most Logo learning environments; introductory computer science, problem solving, and mathematical thinking. Activities will be suitable for all versions of Logo.

[B] [CREATING A LOGO TOOLBOX]

PRESENTERS :Michael Tempel :Brian Silverman

This workshop will focus on creating a collection of tool procedures to customize Logo in such areas as graphing, statistics, language processing, mathematics and physics. Some applications require sophisticated Logo programming techniques. In particular, the workshop will address questions of procedure, recursion, Logo grammar, and the use of reporters (operations).

[C] [LOGO LANGUAGE ACTIVITIES :A WORDS AND LISTS WORKSHOP]

PRESENTER :Alison Birch

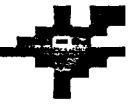
This workshop will give participants a chance to learn the fundamental principles of manipulating words and lists in Logo. By investigating Logo in a playful environment, primitives for taking Logo objects apart and putting them together will be learned. The activities will include use of tool procedures, writing simple interactive programs and examining data structures.

[D] [INTRODUCTION TO LEGO/LOGO]

PRESENTERS :Jean Presser :Bernice Sewell :Gail Sigmund

This hands-on workshop presented by practicing classroom teachers/teacher trainers will introduce participants to the use of the Lego/Logo package. Participants will learn how to build with Lego and how to write Logo programs to control their creations.

END


TO FRIDAY, MAY 6, 1988

REGISTRATION

Stouffer Tower City Plaza

TIME [8:00 am - 6:00 pm]

OPENING SESSION :Dan Watt :Molly Watt

CREATING PROBLEM SOLVING AND CRITICAL THINKING

:ELEMENTS OF AN EFFECTIVE LOGO ENVIRONMENT

TIME [8:30 am - 9:30 am]

VENDOR AREA OPEN

TIME [10:00 am - 6:00 pm]

BREAKOUT SESSIONS :1-5

[There will be 5 topics to choose from in each session]

TIME [10:00 am - 5:00 pm]

DINNER AND ENTERTAINMENT

TIME [7:00 pm -10:00 pm]

END



TO BREAKOUT SESSIONS :1-5

BREAKOUT SESSION 1 :TIME [10:00 am - 11:00 am]

BREAKOUT SESSION 2 :TIME [11:30 am - 12:30 pm]

BREAKOUT SESSION 3 :TIME [1:00 pm - 2:00 pm]

BREAKOUT SESSION 4 :TIME [2:30 pm - 3:30 pm]

BREAKOUT SESSION 5 :TIME [4:00 pm - 5:00 pm]

END

TO SATURDAY, MAY 7, 1988

REGISTRATION

Stouffer Tower City Plaza

TIME [8:00 am - 12:00 noon]

OPENING SESSION :Seymour Papert

BEYOND THE BEGINNING

TIME [9:00 am - 10:00 am]

VENDOR AREA OPEN

TIME [10:00 am - 4:00 pm]

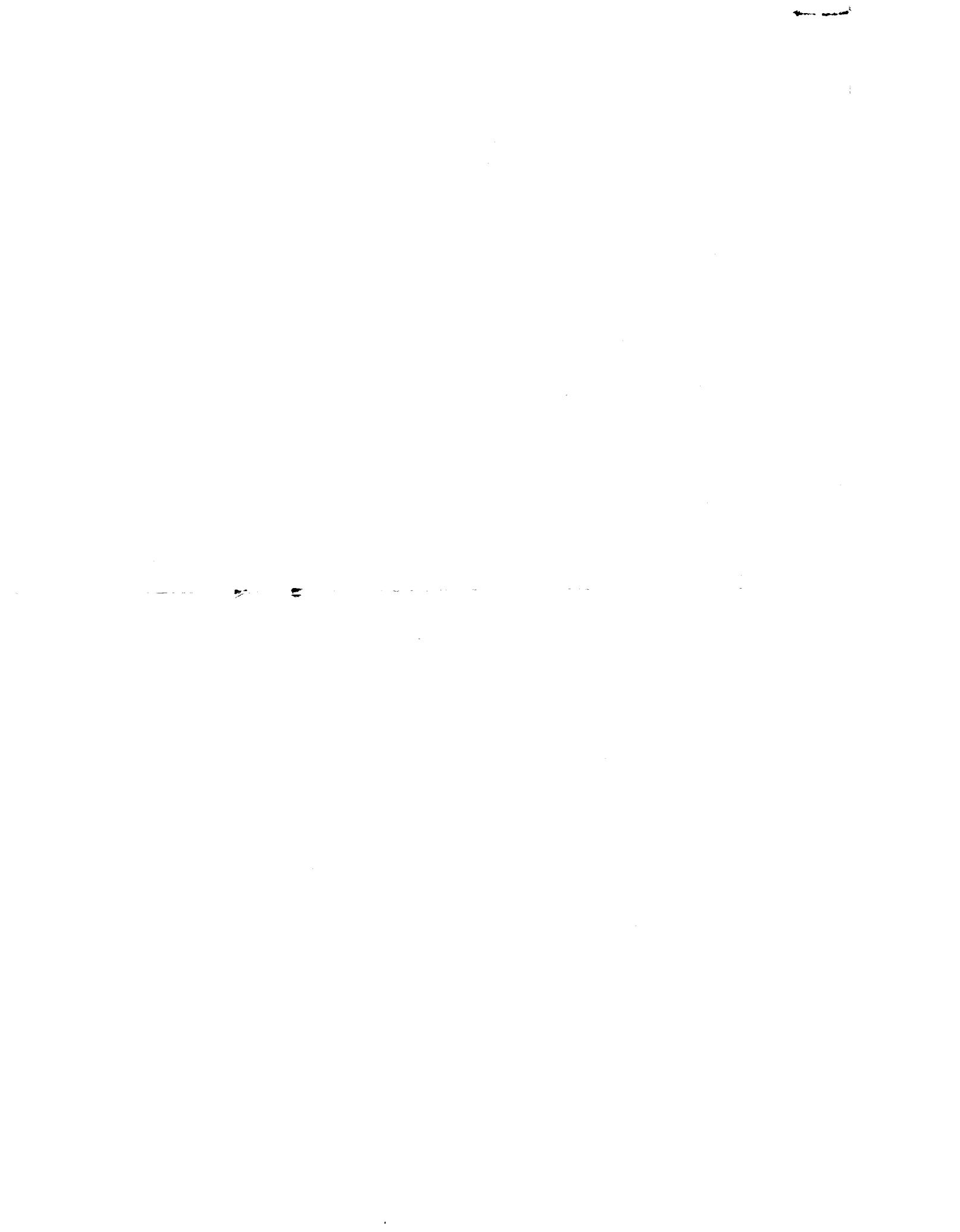
PANEL DISCUSSIONS

TIME [10:30 am - 4:00 pm]

BREAKOUT SESSIONS :6-11

[There will be 5 topics to choose from in each session]





activity card "Guess a Number" for another use of RANDOM.

Nudges by Bull, Tipps, Riordon et al. includes a number of interesting RANDOM projects which might be fun to try. (See pages 233-249.)

Recently, when we began to talk about adding to variables in a recursive procedure, I became aware of just how effective the RANDOM lesson had been. Marc asked if he could use RANDOM to select whether to add, subtract, multiply, or divide by the increment to the variable. This was clearly a question he would not have asked had he not participated in the RANDOM experiment. Marc's problem is a good one for you, the teacher, to try to solve! (What's your solution to this? We might try collecting solutions. Can you do it using symbols or is it better to use SUM, DIFFERENCE, etc. - terms vary depending on your version of Logo. If you find one, please send it to me, noting the Logo version you used, and I'll share results.)

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

Eadie Adamson is a computer teacher at the Allen-Stevenson School in New York City where she works with students at a variety of age levels.

Technical Tip

from
 Eadie Adamson

Here's a temporary solution to the overprinting problem with Apple Logo II:

Load Applewriter IIc (ProDos version)
 Insert Logo disk containing your procedures.
 Set the prefix for the Logo disk.
 Load the file containing your procedures.
 Print!

This is somewhat messy in that you have to load another program, but it gives you control over the line length. This technique will also work with Apple Logo and the DOS 3.3 version of Applewriter.

You can also use Applewriter when typing a program. Save the file with a .LOGO at the end. You can then load it as a Logo file. (Thanks to Joe Delibero of Packer-Collegiate, Brooklyn, for this idea!) This is also useful if you have a lot of editing to do: you can use Find/Replace or Copy facilities of the wordprocessor to speed the process.

You can use this same technique to create Logo II files using AppleWorks, but you must save them as ASCII files. See the column by Glen and Gina Bull for more about ASCII.

Great Lakes/ East Coast Logo Conference

The Great Lakes/East Coast Logo Conference will be held on May 6-7, 1988 with preconference workshops on May 5. This conference, held at the Stouffer Tower City Hotel in Cleveland, OH will feature numerous well-known speakers: Seymour Papert, Dan and Molly Watt, Michael Tempel, Tom Lough, Brian Silverman, Steve Ocko, Mike Battista, Doug Clements, Alison Birch, Sharon Burrowes, Brian Harvey, Sylvia Weir and many more.

Presentations describe university, district, and school based Logo projects, address issues of equitable use of computers and present uses of Logo beyond the introductory level.

See center of this issue for registration materials.

Teaching Tools

The Ins and Outs of ASCII

by Glen Bull and Gina Bull

Certain technologic phenomena only make sense in the proper historical context. In *Mindstorms*, Seymour Papert describes the QWERTY phenomenon, named after the letters on the first row of the keyboard. In the nineteenth century the arrangement of keys on the keyboard was not optimized for speed, since the levers of the early typewriters tended to jam at higher speeds. The arrangement of keys on most computer keyboards is still the same today, even though there are no levers to jam. In this column we will discuss two Logo commands you may not have used, even though they are useful for construction of a wide range of Logo tools. If you have not used these commands, the chances are because they are best understood in their proper historical context.

The Logo commands we have in mind are ASCII and CHAR. References to ASCII codes appear in a number of seemingly unrelated places. They can allow you to:

1. Use Control-Q to end a Logo procedure.
2. Use the arrow keys to control the movement of the turtle.
3. Change uppercase letters to lowercase, or vice versa.
4. Sort a list of words into their proper alphabetical order.
5. Construct cryptograms and codes.
6. Control peripherals such as printers, speech synthesizers, and videodisc players.
7. Share Logo procedures with other teachers via electronic mail services such as CompuServe.

Some of the first electronic codes were developed with the advent of the telegraph. Morse code is a well-known system that is still in use today.

The telegrapher used an actual telegraph key to generate the dots and dashes of Morse code. Later Teletypes replaced the telegraph key in most offices. The Teletype consisted of a keyboard which would generate an electronic code when the proper key was pressed. This code would be transmitted down the telegraph wire and received by a corresponding Teletype machine at the other end. One of the most widely used was the five-bit Baudot code, named after the Frenchman Baudot.

American Standard Code

Another code widely used in Teletype systems was the ASCII code. ASCII stands for "American Standard Code for Information Interchange." This code is used on almost all microcomputers systems today. In the ASCII code, there is a number which corresponds to each letter of the alphabet. For example, the number 65 stands for A, the number 66 stands for B, and so on:

Letter	ASCII Code
A	65
B	66
C	67

To convince yourself that this is so, find a computer, boot Logo, and type:

```
PRINT ASCII "A
65
```

```
PRINT ASCII "B
66
```

As you can see, the computer knows the ASCII code

the ASCII code associated with a particular letter, you can ask Logo to tell you or look it up in a table. The numbers 65 through 128 stand for all of the letters and some of the other symbols as well, such as [, \, {, etc. If you want to know the ASCII code for any of these, just ask Logo:

```
PRINT ASCII "[  
91
```

If 65 through 128 stand for the letters and other symbols, what do the numbers 1 through 64 stand for? Many of these numbers represent control keys. For example, if you hold down the key labeled CTRL or CONTROL (usually found on the left-hand side of the keyboard), and press A, the ASCII code 01 is generated. In fact, if you subtract 64 from the ASCII code for any letter, the code for the same letter with the CONTROL key held down is obtained.

Key	ASCII Code	Key	ASCII Code
Control-A	01	A	65
Control-B	02	B	66
Control-C	03	C	67

In many word processors, the CONTROL keys are used for editing. For example, in many cases holding down the CONTROL key and pressing H results in a backspace. In fact, Control-H is the official ASCII code for a backspace.

Discovering ASCII Codes

The ASCII function allows us to write a Logo procedure to determine the ASCII code that goes with any key.

```
TO CODE?  
PRINT [PRESS ANY KEY TO]  
PRINT [SEE ITS ASCII CODE:]  
PRINT ASCII RC  
END
```

Use this procedure to determine the ASCII code for Control-Q.

```
CODE?  
PRESS ANY KEY TO
```

SEE ITS ASCII CODE:<— Press Control-Q Here
17

Now that we have discovered that the ASCII code for Control-Q is 17, this information can be used to stop a Logo procedure. For example, here is an INSTANT Logo procedure which makes the turtle go forward when the child presses F, and turn right when R is pressed. The only problem is this procedure goes on forever. It is possible to abort the procedure somewhat inelegantly in some versions of Logo, and other versions such as Apple Logo II make it difficult to terminate the procedure at all.

```
TO INSTANT  
MAKE "KEY RC  
IF :KEY = "F [FD 10]  
IF :KEY = "R [RT 30]  
INSTANT  
END
```

Using ASCII Codes

One solution would be to check for the letter Q, and quit if this letter is pressed. The difficulty is that the letter Q might be accidentally pressed by the child, ending the procedure unintentionally. Using Control-Q to terminate the procedure provides a controlled exit from the procedure, but lessens the possibility of accidental termination. (Note: the delimiters around ASCII :KEY are curved parentheses, not square brackets.)

```
TO INSTANT  
MAKE "KEY RC  
IF :KEY = "F [FD 10]  
IF :KEY = "R [RT 30]  
IF (ASCII :KEY) = 17 [STOP]  
INSTANT  
END
```

The ASCII codes that correspond to the arrow keys on the keyboard can be identified in the same manner as Control-Q. For example, run the procedure CODE? and press the right arrow key. What number is associated with the right arrow?

```

CODE?
PRESS ANY KEY TO
SEE ITS ASCII CODE:<—Press Right Arrow here
—————<— What code is generated?

```

(Note: If you are using a version of Logo for the IBM, you may get some surprises since control characters show up as printable characters such as smile faces!)

After the ASCII codes for the arrow keys have been identified, they can be used to control actions of the turtle just as the letters on the keyboard can. In the following example, you will have to fill in the ASCII code for the up arrow and the right arrow in the appropriate blanks.

```

TO INSTANT
MAKE "KEY RC
IF :KEY = "F [FD 10]
IF :KEY = "R [RT 30]
IF (ASCII :KEY) = _____
    [SETY YCOR + 10]
IF (ASCII :KEY) = _____
    [SETX XCOR + 10]
IF (ASCII :KEY) = 17 [STOP]
INSTANT
END

```

If someone does not know the ASCII code, Logo can be used as a "decoder ring" to send secret messages. The procedure encrypt takes each key typed, and prints its ASCII code.

```

TO ENCRYPT
MAKE "LETTER RC
PRINT ASCII :LETTER
IF (ASCII :LETTER) = 17 [STOP]
ENCRYPT
END

```

For example, this is the message generated when HELLO is typed.

ENCRYPT

72	<— type H
69	<— type E
76	<— type L
76	<— type L
79	<— type O
17	<— type Control-Q to end procedure

The CHAR Function

The ASCII function can be used to determine the ASCII code associated with any character. The CHAR function does the opposite; it returns the character associated with any ASCII code.

```

PRINT ASCII "A
65

```

```

PRINT CHAR 65
A

```

With the CHAR function, it is possible to make a more sophisticated substitution cipher. Letters as well as numbers can now be included in the code.

```

TO ENCRYPT2
MAKE "LETTER RC
PRINT CHAR (ASCII :LETTER) + 10
IF (ASCII :LETTER) = 17 [STOP]
ENCRYPT2
END

```

When ENCRYPT2 is run, HELLO comes out in the following form.

```

ENCRYPT2
R <— type H
O <— type E
V <— type L
V <— type L
Y <— type O
<— type Control-Q to end procedure

```

This type of substitution code which rotates the letters by 10 positions is easily broken. However, a variant of this code which rotates the letters by 13 positions is widely used on certain forums on USENET, a network linking several thousand com-

puter science departments and other sites across the United States. (The purpose is not scholarly, however. It is simply to encrypt dirty jokes so those who might be offended need not read them.) A more sophisticated function could be substituted for the simple rotation to make the code more difficult to break.

Shifty Characters

We have seen that the control character associated with a letter can be obtained by subtracting 64 from the ASCII code for that letter. (If the ASCII code for A is 65, the ASCII code for Control-A will be 01.) Adding 32 to the ASCII code for a letter will generate the ASCII code for the lower case equivalent of the letter.

Key	ASCII Code	Key	ASCII Code	Key	ASCII Code
Control-A	01	A	65	a	97
Control-B	02	B	66	b	98
Control-C	03	C	67	c	99

The relationship between upper and lowercase letters can be used to construct a Logo function which converts uppercase to lowercase, or vice versa. This function is built into some of the more recent releases of Logo, but is easily constructed for those which do not have it.

```
TO LOWERCASE :LETTER
  OUTPUT CHAR (ASCII :LETTER) + 32
END
```

The function works like this:

```
PRINT LOWERCASE "A
a

PRINT LOWERCASE "B
b
```

The function can readily be extended to convert the letters in an entire word:

```
TO SHIFT :ITEM
```

```
IF EMPTYP :ITEM [OUTPUT " ]
  OUTPUT WORD LOWERCASE
    (FIRST :ITEM) (SHIFT BF :ITEM)
  END
```

The procedure works like this:

```
PRINT SHIFT "CAT
cat
```

Controlling Peripherals

ASCII codes can also be used to control peripherals. For example, the official ASCII code for generating a form feed (which makes a page eject from the printer) is 12. If you have a printer attached to your computer, try this. First put the computer in a mode in which output is directed to the printer. The precise way in which this is done varies from dialect to dialect of Logo.

```
DRIBBLE 1      ← Apple Logo II
            (if printer card is in slot 1)
```

```
DRIBBLE "PRN      ← IBM Logo
```

```
DRIBBLE "PRINTER   ← LCSI Logo II
            for the Apple
```

```
OUTDEV 1      ← Terrapin Logo
```

Once output is directed to the printer by the means appropriate for your version of Logo, send the ASCII character 12 to the printer.

```
DRIBBLE 1
PRINT CHAR 12
```

Did a sheet of paper eject from the printer? If so, you could make a Logo procedure to generate a formfeed, rather than walking over to the printer and rolling up the paper manually.

```
TO FORMFEED
  DRIBBLE 1
  PRINT CHAR 12
```

```
PRINT CHAR 12
NODRIBBLE
END
```

It is amazing how much more convenient this becomes once the FORMFEED command is installed as a working Logo tool. We usually abbreviate the procedure title as FF.

Extension

References to ASCII codes appear in a surprising number of other contexts. For example, as this is written the letters in this file are being stored in memory in ASCII format. This is the most widely used standard for storing and exchanging files among different programs and different computer systems. For example, a Logo file written in Apple Logo II or in LCSI Logo II for the Apple can be read into an

Appleworks file. Just select the ASCII option when reading the file into Appleworks. Standard ASCII formats can also be used to exchange files between otherwise incompatible systems. For example, an Apple Logo II file can be transferred to an IBM and read into IBM Logo in this manner. (In fact, rather than retyping Logo procedures for different dialects of Logo, we often move them from one system to another in this way.) This format is also often used on electronic services such as CompuServe. The developers of one telecommunications program designed to transfer files in this way even called their program *The ASCII Express*.

The ASCII and CHAR functions are surprisingly versatile, but often go unnoticed. ASCII references are more comprehensible with the proper historical context. In a sense, history and context are sometimes among the best teaching tools of all. ▲

Site Licensing of Terrapin Logo

Terrapin, Inc. announces site licensing for its three versions of the Logo language - Terrapin Logo for the Apple, Commodore Logo, and Terrapin Logo for the Macintosh.

A Terrapin Logo site license includes copyable language and utility disks as well as complete documentation. The *LogoWorks: Lessons in Logo* student book and teacher's guide are included with the Commodore and Apple site license packages, along with a supply of Quick Reference Cards and a subscription to Terrapin's Logo newsletter, the *Terrapin Times*. The *LogoWorks* package, also sold separately for \$42.90, contains 47 lessons and over 250 activities for students.

The one-time site license fee is \$299 for Commodore Logo and \$399 for either Terrapin Logo for the Apple or Macintosh. Schools may obtain site licenses for two versions of Logo for \$450 or all three site licenses for \$500. (Although counted as a single version, the Apple site license contains copies of both the 64K and 128K Logo languages.) Schools that have previously purchased lab packs of Terrapin Logo will be offered a reduced site license fee ranging from \$100 to \$200. Terrapin is also offering a \$100 site license for the Logo Data Toolkit, a set of four application programs for creating databases, surveys, charts and association games.

For more information about site licensing contact Ann Ciara, Terrapin, Inc., 376 Washington Street, Malden, MA 02148; ph. 617/322-4800.

LXIONARY

By Bill Craig

"Computer Criticism Vs. Techocentric Thinking" by Seymour Papert, Logo 85 Theoretical Papers

This is Papert's now famous "ask not what Logo can do for you, but what you can do with Logo," paper. The paper is an attempt at encouraging a new way of thinking about computers in schools, a less technocentric way. The computer itself or the computer language we use should not be the basis for evaluating the utility of computers in schools. Papert does not believe that it is reasonable to expect Logo or computers to deliver great change or benefits to students. Neither the computer nor Logo is a change agent,

"The context for human development is always a culture, never an isolated technology."

Rather than evaluating Logo, we should be evaluating the way in which Logo is used. Papert uses this point to criticize his critics, specifically the Pea study from the Bank Street College of Education. Pea reported that children given Logo instruction failed to show improvement on a test for cognitive change. Papert claims that this treatment method (comparing students who received Logo instruction with those who had not) for measuring the impact of Logo instruction is not valid if "... you see Logo not as a treatment but as a cultural element ... something that can be powerful when it is integrated into a culture but is simply isolated technical knowledge when it is not."

In place of treatment studies, Papert proposes the development of writing he calls computer criticism, analogous to literary criticism. While Papert was very specific about the shortcomings of then existing Logo research, he is less specific in this paper about what computer criticism really is. Computer criticism should examine not only what hap-

pens to students after they use computers but it would examine the "social structures and cultures that existed before the computer." He provides examples of Logo and computers being used in different school culutures for different purposes and with different outcomes.

Papert's thesis that we must examine our uses of Logo and not just Logo is an excellent one. What his description of computer critcsim lacks are criteria for evaluating our Logo and computer uses.

"Logo Needs Research: A Response to Papert's Paper."

by Decker F. Walker,
Educational Researcher June-July, 1987.

Walker's article is one of several responses to Papert's paper. Walker begins by acknowledging the validity of several of Papert's points. There were design flaws in some of the Logo studies. But it is not reasonable to then claim that,

"...these shortcomings of conventional experiments are not reason enough to rule them out, apparently with all other forms of systematic inquiry except case studies..."

There are two messges in Walker's article. The first is that there is a happy medium between treatment studies and computer criticism, specifically Piagetian interviews. The second, and much more passionately stated, is that some system of sound research is needed to give credibility to the claims of the Logo community.

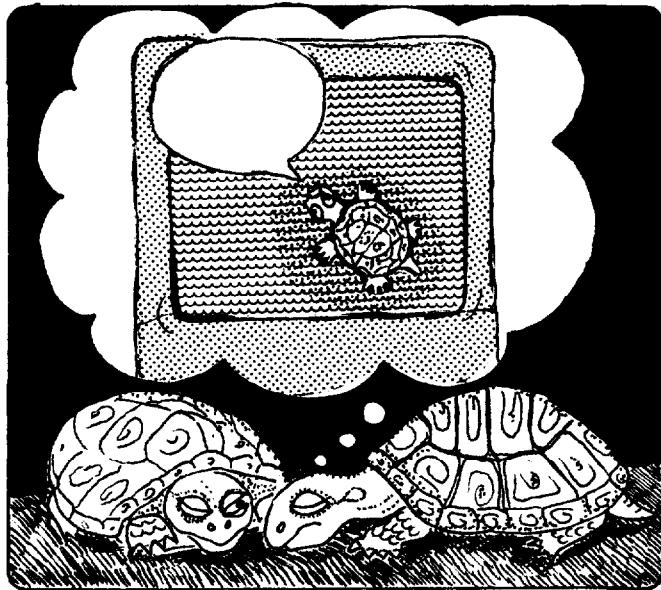
"The broad center constituency is not enthralled by the intrinsic charms of the Logo culture and it will not be won over by anecdotes. But it might by won over by convincing evidence that children benefitted from Logo in some objectively demonstrable way."

I agree. ▲

The Adventures of Jacques and Elsie

drawings by
Linda Sherman

"Whoever said turtles were slow had obviously never met the Logo turtle or Jacques and Elsie! Help Jacques demonstrate how quick-witted turtles can be by sending in his comment on the speedy Logo turtle."



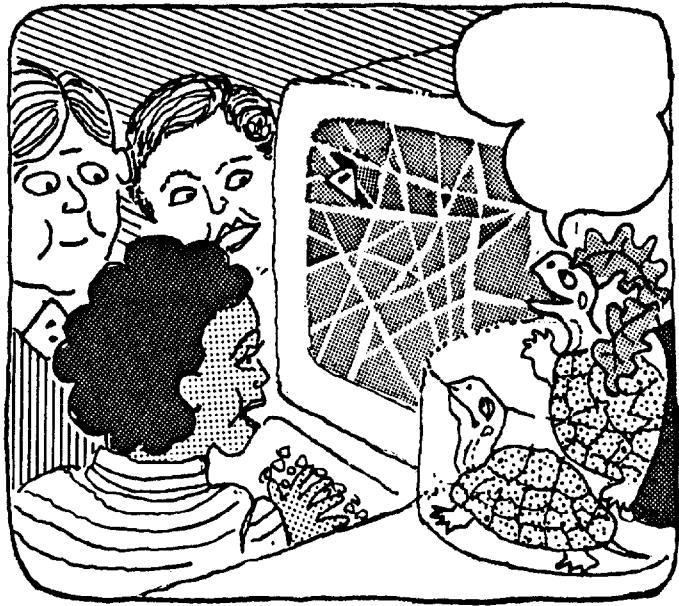
Send your caption to:

Linda Sherman
Rt. 1 Box 292-A
Shipman, VA 22971

October Cartoon

October's caption was submitted by ... Jacques himself!

"Maybe if you stuck your head out of your shell you'd be able to see where you're going!"



Linda Sherman is a freelance author and artist living in Shipman, VA., with her husband and two-year old son.

Book Review

by
Marie Balet and Lou DeLucia

Burke, M. P. & Genise, L. R. (1987). *Logo and Models of Computation*. Menlo Park, CA: Addison Wesley.

If you're looking for a computer science book to develop procedural thinking and algorithmic design, then Logo and Models of Computation may be for you. Burke and Genise use a Logo environment to present many essential topics in the study of computer science. Some of these topics include structured programming, functions and procedures, the development and refinement of algorithms to solve problems, and error handling routines.

As a text for teaching Logo, this book offers a refreshing change of pace by beginning with word and list processing rather than turtle graphics. The authors present an overview of the Logo language followed by programming with graphical objects. A solution to the Towers of Hanoi problem provides an excellent visualization of mathematical induction. A thorough explanation of recursion, procedures, and functions is presented with substitution and simplification models and level diagrams discussed in depth. Global variables, programming with multiple turtles, and the use of mathematical induction to prove the recursion procedures are also included. The appendixes include some computation theory, a glossary, differences encountered in Apple II Logo or IBM Logo versions, Logo primitives, and answers to selected problems.

If this book were to be used as a text for a high school computer science course, the teacher would probably need to develop additional materials and resources and might wish to modify certain aspects of the presentation. For example, the authors explain order of operations using left and right

binding powers rather than the more traditional approach. This seems somewhat complicated and unnecessary, although the binding power terminology is used later on in the text to explain evaluation of more complex Logo expressions. The second chapter provides an interesting overview of the Logo language which could be covered in a relatively short period of time.

The book is well organized and well thought out. A nicely designed format enhances readability. The exercises are appropriate and "do-able" if the student understands the preceding text. Although Chapter 9 provides several interesting projects and application exercises, the book could not easily be used as a reference for additional problems. The problems and exercises build quite heavily on material presented in the text and on previous problems in the text.

The authors' approach almost ensures that a student will write well structured programs and, as such, would be an appropriate preparation for LISP or other computer science courses. The book may not be appropriate for the average high school student but could be used successfully with above average students or in a college course.

In the forward to this book, the authors state that the goals are

...to provide an introduction to computer science, to illustrate the relationship of computation to mathematics, to aid in the development of a useful problem solving discipline, and to provide an intellectual framework in which new concepts in computer science can easily be assimilated. The Logo programming language is considered by us to be the ideal vehicle to reach these goals.

These are significant goals and, in our opinion, the authors have been successful in attaining them. The book is an excellent resource for any computer science teacher. ▲

IntLXual Challenges

by Robs Muir

The long-standing goal of this column is to provide a forum to provoke, engage, excite, and expand the horizons of the "sophisticated" Logo user. It is our belief that adults need "objects to think with" at least as much as do children. While much of the literature surrounding the Logo movement has been focused on the role of a computer in children's lives, it is this author's belief that the best insurance for computer literate children (for want of a better term) is computer literate adults. Passion encourages passion. Computer Education demands adults with more than a superficial understanding of computers; Computer Education needs adults who are actively engaged in computing as a meaningful part of their own lives. As Papert has said, "What's good for thinking is good for thinking." If Logo is good for kids, it is similarly good for the rest of us.

Programming, especially in a procedural language like Logo, provides an evocative medium that can serve as a metaphor remarkable in its inclusiveness. Observe how quickly computing terminology has become idiomatic. Few activities and processes can escape being compared with "computing engines" and their workings; the computer is the canvas on which our changing culture is portrayed.

A (Logo) program is a collection of procedures each of which, in turn, play a role in the larger plan. In a well-structured program, individual procedures may rely on other procedures to accomplish their individual sub-goals. Masterfully written programs include procedures that are multipurpose in nature—that is, they can be called upon by a variety of different procedures to "help out." It is difficult not to personify procedural programs and imagine that our programs are composed of "actors" with goals, needs, demands, and "lives" of their own. Whole governments, social systems, and even small

groups of friends can be seen as behaving like computer programs, so it is only fair that complex software can be anthropomorphized. Indeed, debugging an unruly programming project often feels like trying to manage a classroom full of students—achieving unity of action among (sometimes) incompatible units!

Logo includes powerful debugging tools intended to help understand and thereby solve difficulties amongst the individual procedures that comprise our programs. STEP and TRACE are examples of such tools and were available in Logo on our personal computers since 1980. These can be written within Logo. (For examples, see LCSI's Apple Logo Toolkit Diskette.) Digital Research's DR Logo was one of the first commercial implementations of Logo to include really powerful debugging facilities beyond STEP and TRACE. Primitives such as WHOUSES (who-uses) and WHOCALLS (who-calls) allow us to "inquire" as to which procedures require the services of a specific variable or other procedure.

```
SHOW WHOCALLS "RECTANGLE
[HOUSE DOOR WINDOW]
```

```
SHOW WHOUSES "SIZE
[CIRCLE HOUSE BOX]
```

Such debugging tools allow us to orchestrate large collections of procedures, thus ensuring harmonious programs.

Agents and agencies; procedures and programs. How easy it is to apply these ideas to our world! Yet, it is ironic that the one area where we are loath to see such parallels is the "place" closest to home—that place which we call our mind. In this

world of psycho-babble, New Age mysticism, artificial intelligence, superstition, and fuzzy thinking, it is difficult to unravel what we know about the workings of our inner selves. As Logo-ists (and as Educators!) we should be the most interested in how the mind learns, grows, and works. Is there a Soul in the machine? Or is DNA the Operating System? Is it reasonable to use a computer as a metaphor for "Me," or even to think of it as being creative?

Enter the most important book since *Mindstorms*; Marvin Minsky's *The Society of Mind* [Simon and Schuster, 1986. ISBN:0-671-60740-5]. Perhaps it is the format (large paperback), or the illustrations (were they all done on a Macintosh?), or the author's daunting reputation (MIT professor, artificial research pioneer, etc.), or the topic (Mind?! Whew...). For some reason, I have yet to see a mention of this profound book in any of the Logo literature. Yet it represents the thoughts of a brilliant thinker's life work; a career spent focused on arguably the most important question ever asked by Humankind. This book is simply too important to miss!

One blurb: "Marvin Minsky's society theory of mind is as revolutionary a conceptual breakthrough for today as Piaget's theory was in his own time."—Professor Barbel Inhelder, Foundation Archives Jean Piaget, University of Geneva.

If you are interested in educational theory, computer science, artificial intelligence, childhood development, psychology, cognitive science, philosophy, Logo, or all of the above, this book belongs on your reading list. Consider it this month's principal challenge. My Summer has been lost to this readable, yet challenging theory; I will be forever changed.

If you are in need of a Logo programming challenge, let me suggest a debugging tool patterned after WHOCALLS. By a broad stretch of the concept, this tool is somewhat like Minsky's concept of a K-line (or Knowledge-line)—a mechanism conjectured to help us recreate particular brain-states. How might you build a procedure that inspects every procedure in the workspace and reports whether or not a specific procedure is therein called? ▲

LogoPals

by Barbara Randolph

Why do you enjoy using Logo? Letters from LogoPals often reflect the wonderful enthusiasm students share about learning to use Logo. Perhaps you have heard comments similar to these in your classroom:

Jill L.: "I love Logo...and I have fun doing Logo in school. I always look forward to that one day of the week when I get to go to the computer lab and work on Logo for a little while."

Gina G.: "My favorite activity in Logo is when you

can make a shape and change its sizes."

Tricia M.: "In Logo we use different colors. We use lots of squares and make designs."

Pete B.: "My favorite in Logo is when I do a string of commands and I get a neat picture."

Sharon B.: "I am in advanced Logo and I love it! I like making up my own programs."

Steve D.: "My favorite thing to do in Logo is just having lots of long program lines and guessing what

will happen."

Vita L.: "We've been making designs in Logo. I like the surprises Logo gives me."

And these LogoPals would love to "meet" someone new and talk about Logo:

Kara Kownacki (McKees Rock, Pennsylvania, USA): I am 10 years old and in the 5th grade. I am very interested in computers...and would very much like a penpal from Poland, Germany, or Japan.

Suzanne Bamrick (Orchard Park, New York, USA): I like skiing and swimming. I am eight years old. Please send me a LogoPal from Australia.

Steve Dries (Wauwatosa, Wisconsin, USA): I would like a penpal from Germany. I am a boy, age 10 in 5th grade. I like Logo alot. I play basketball, soccer and tennis. (Or a penpal from Austria, Turkey, England or Alaska.)

Joe Cal (Orchard Park, New York, USA): I like baseball and karate, tennis and skateboarding. I like Logo alot too. I'm 8 years old and in third grade. Please send me a LogoPal from France.

Tricia Miller (Wauwatosa, Wisconsin, USA): My favorite hobbies are skiing, biking, tennis, swimming, skating, art and dancing. I also love reading. I would like my LogoPal to be from Italy.

Sara McCloskey (McKees Rock, Pennsylvania, USA): I like playing the piano, baseball and drawing. I am ten and in the 5th grade. I would like a penpal from France, Ireland or Egypt. (I almost forgot) I know beginner's Logo.

Kim Whistler (Orchard Park, New York, USA): Please send me a LogoPal from Hawaii or Alaska. I like skating and skiing and swimming. I am eight years old and in third grade.

Jodi Stenz (Wauwatosa, Wisconsin, USA): I'm in third grade. I like soccer and ice skating. My favorite

thing in Logo is making designs with the turtle. I would like my penpal to be from Canada because I want to know what they do in Logo.

Jessica Fitzpatrick (Orchard Park, New York, USA): I like art, swimming, gymnastics, soccer, volleyball, and I enjoy writing and drawing. I like Logo mazes. I would like a girl Logo penpal from Australia, Italy or Mississippi.

Dria Kozik (Milwaukee, Wisconsin, USA): In winter I like to go skiing. In spring and summer I like to go camping. During fall I like to collect leaves. In Logo I have made neat pictures. I also like to print my stories out from the computer. I would like my LogoPal to be from someplace in Wisconsin.

Do you have students who would like to become Logo penpals? They can be matched with these boys and girls or others we have in our network. Have them write to me, telling their age and grade, their favorite Logo activities, and hobbies and interests. Their ages can range from primary school through high school levels. They may even enjoy seeing their "ads" (excerpts from their letters) printed here.

Students in the USA also need to send a self-addressed stamped envelope with their letter. Those outside the USA should enclose international postal coupons (purchased at the post office) for a 1-ounce or 28-gram reply.

Spread the word to your friends. LogoPals are seeking to make friends in every corner of the world. If you like, you may send me names and addresses of Logo educators in other countries and other parts of the United States. I'd be delighted to write to them about LogoPals. Thanks!

Write to:

LogoPals

c/o Barbara Randolph
1455 East 56th Street
Chicago, Illinois 60637
USA



Barbara Randolph is a librarian and instructional media center teacher in the Chicago Public Schools.

Africa	Asia	Australia	Europe	Latin America
Fatimata Seye Sylla Lab Informatique et Ed BP 5036 Dakar Senegal, West Africa	Hillel Weintraub Doshisha Int. Sch. Tatara, Tanabe-cho Twuzuki-gun Kyoto-fu, Japan 610-03	Anne McDougall Faculty of Education Monash University Clayton, Victoria 3168 Australia	Richard Noss London U. Logo Group IE, Dept. of Math 20 Bedford Way London, UK WC1HOAL1425 Argentina	Eduardo Cavallo & Patricia Dowling Instituto Bayard Salguero 2969 Buenos Aires

International News

edited by Dennis Harper

Two European Logo Conferences

by
Ken Johnson

**British Logo User Group (BLUG) and InterAction joint Summer School,
London, 24 to 27 August 1987**

The Summer School was a joint venture between the British Logo User Group and InterAction, a charity concerned with youth work outside schools. The venue was Southlands College, a Methodist teacher training college in Wimbledon, south London. About sixty people attended, nearly all of them from southeast England. Roughly half of them were BLUG members and the other half used, or wanted to use, computers in youth work. There were therefore two main types of sessions: those run by BLUG members tutoring others in the Logo language, and those run by InterAction members whose interests in using computers were broader. Hence, sessions on spreadsheets, desktop publishing and computerized administration balanced those on writing Logo adventure games, starting list processing, and 3-D Logo. The organizers had arranged that there would always be a choice between Logo and the more general sessions. Students attended sessions of both types freely. Commercial manufacturers had been generous in lending equipment for students to use.

The session that I found most memorable was titled "*Logo and childrens' games*" by Bruce Edmonds. He introduced half a dozen of us to simple games intended to break down the barriers between people and help a group of strangers to gel into a group of friends. Other games were meant to illustrate some point about how Logoworks that beginners might otherwise find confusing: "playing Turtle" is a well-known example; "playing stack," which illustrates how a simple recursive programs works, was new to most of us.

I was also struck by walking into a laboratory full of people making simple Lego models of fair-ground rides and other machines and hooking them up to computers to control them. There seemed enormous potential for producing absorbing work for small children as well as for teaching the basics of electricity, electronics, and computing.

The Annual General Meeting of the British Logo User Group took place on Wednesday afternoon. Mike Doyle, Bruce Edmonds, Henry Leibling, Ros Penny, and Chris Robinson were elected to the

vacant places on the Committee. I was elected Chairman for the year; Allan Martin remains as Vice Chairman, Janice Staines as Secretary and Mike Doyle as Treasurer. We are continuing the slow progress towards turning BLUG into an educational trust.

First European Logo Conference

This was a meeting of about a hundred people, drawn from throughout the EEC but with a couple of Americans, too, held in St. Patrick's College, Drumcondra, on the northern outskirts of Dublin. It was hot and sunny and the grass really is greener than you have ever seen elsewhere.

The principal themes dominating Logo in Europe were clearly visible in the activities here: on the one hand, the developments of Control Logo (Lego and Logo), 3-D Logo and LogoWriter generated a lot of excitement; on the other, constant discussion of the desperate shortage of money which prevents schools from making these developments available to their pupils in enough quantity or at enough places.

Two lectures in particular attracted a lot of attention and discussion. Bob Sparkes of Stirling University, Scotland, demonstrated Control Logo in his opening address as well as showing simple Logo procedures with which he had been able to teach the rudiments of wave mechanics to young children in school. Mike Doyle gave a full demonstration of the surprising and entertaining programs you could write for his special home-made Turtle which has senses of touch, sight, and hearing.

We heard of new dialects of Logo: version 2 Nimbus Logo and Apple IIe LogoWriter were on show to be played with; we were told of LCN Logo from Holland and of Logo in the minority languages Welsh (available) and Irish Gaelic (possible). LogoWriter, incidentally, is the first dialect of Logo I've seen in which the brackets in *if* and *repeat* statements etc. can be formatted.

We've heard reports of Logo work going on in Ireland, in the English cities of Lancaster and Liverpool, in Nijmegen and in Madrid. Controversy

about the exclusion of Logo from examinable courses in Scotland continues: a paper called "The Brain Damaged Programmer: effects of exposure to BASIC" added fuel to the fire. And we heard of miracles: one speaker reported a spastic child who learned to walk in order to get to the terminal before the others; another, told of seriously disruptive children who visited the local college to play with Logo and learned to concentrate and to do creative and attractive work with the Turtle.

Irish hospitality and friendliness and Guinness are justly renowned. Both the Minister for Science and Technology and the local shinty team dropped in to drink with us at the *ceili* on Saturday night. We all sang and danced and listened to a traditional Irish band of two fiddlers and a concertina player, well into the small hours of Sunday morning. ▲

4th Annual Logo Conference

"Grow with Logo"

**Saturday, March 26, 1988
Education Centre
London, Ontario**

Sponsored by the Logo Special Interest Group of the Educational Computing Organization of Ontario, Conference registration is \$25 and includes four sessions and lunch. For more information or to register, contact Rudy Neufeld, Program Dept., Board of Education for the city of London, 1250 Dundas St., London, ONT N5W 5P2. London is situated between Detroit and Toronto.

***The ECCO Logo Project:
Materials for Classroom Teachers
and Teacher Trainers***

Edited by
Theodore C. Burrowes and Sharon K. Burrowes

An ICCE Publication

This new booklet presents Logo activities for use in grades four through eight. Use it for teacher training or take it directly into the classroom to enhance the teaching of language arts, social studies, science and math.

**The ECCO Logo Project
Includes:**

*Student worksheets
Teacher information sheets
Teacher-training materials
Logo II disk version of all materials
Apple Logo disk version of student materials*

The Educational Computer Consortium of Ohio (ECCO) developed these materials over the course of an academic year, in conjunction with its extensive series of Logo workshops.

To order your copy, use the order form below or call ICCE at 503/686-4414.

Name _____
Address _____
City/State _____
Postal Code/Country _____
Phone _____

Payment enclosed
 Bill me (add \$2.50 handling)
 Bill my:
 Mastercard Visa (circle one)
card number: _____
name: _____
exp. date: _____

Please send me _____ copies
of *The ECCO Logo Project*
at \$20 each. Total price \$_____

Add \$2.50 for shipping \$_____

Add \$2.50 handling
for billed orders \$_____

Total Amount \$_____

Mail to: ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403 USA



June 15-17, 1988
Loews Anatole Hotel
Dallas, Texas

8th National Educational Computing Conference

INVITES YOU TO ATTEND
Papers, Panels, Projects, Exhibits, Full-Day Pre-Conference Workshops

For information contact:

Jim Poirot
Computer Education & Cognitive Systems
North Texas State University
P.O. Box 13886
Denton, TX 76203-3886

For information on Exhibits contact:

Paul Katz
Continuation Center
University of Oregon
Eugene, OR 97403

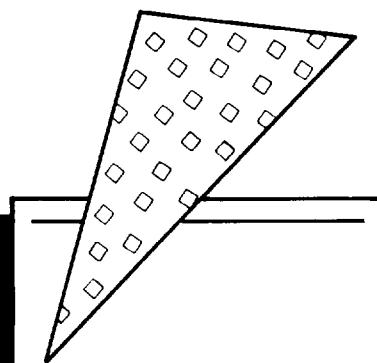
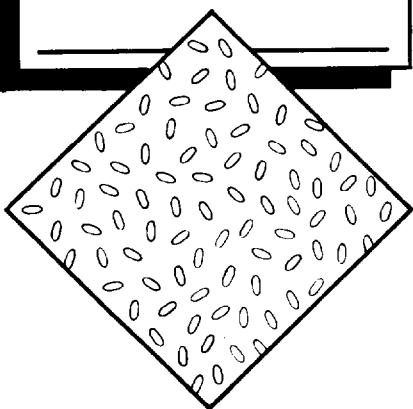
NECC '88 is sponsored by the International Council for Computers in Education (ICCE) in cooperation with member organizations of the NECC Steering Committee:

- Association for Computers and the Humanities (ACH)
- Association for Computing Machinery (ACM) Special Interest Groups on:
 - Computer Science Education (SIGCSE)
 - Computer Uses in Education (SIGCUE)
 - University and College Computing Services (SIGUCCS)
- Educational Computing at Minority Institutions (ECMI)
- IEEE Computer Society
- EDUCOM/EDUNET
- Society for Computer Simulation (SCS)
- Association for Small Computer Users in Education (ASCUE)
- AERA
- International Association for Computers in Education (IACC—formerly AEDS)

Join the Leading Professional Organization for Computer Educators!

Whether you are an administrator, curriculum specialist, classroom teacher, media specialist or special educator, you will benefit from membership in ICCE.

ICCE is the leading U.S. and international professional organization for computer educators. It is non-profit, supported by more than 50 organizations of computer-using educators worldwide.



Membership in ICCE includes a subscription to *The Computing Teacher* journal. The journal has long been respected as an important source of information for computer educators, providing accurate, responsible and innovative information. Nine times per year, its articles, departments and reviews keep you in touch with the constant changes taking place in educational technology.

Members will also receive discounts on all ICCE publications and SIG memberships, and information on ICCE committee activities.

As educational technology continues to grow and change, look to ICCE and *The Computing Teacher* journal to keep you on the leading edge.

Join today!

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

