

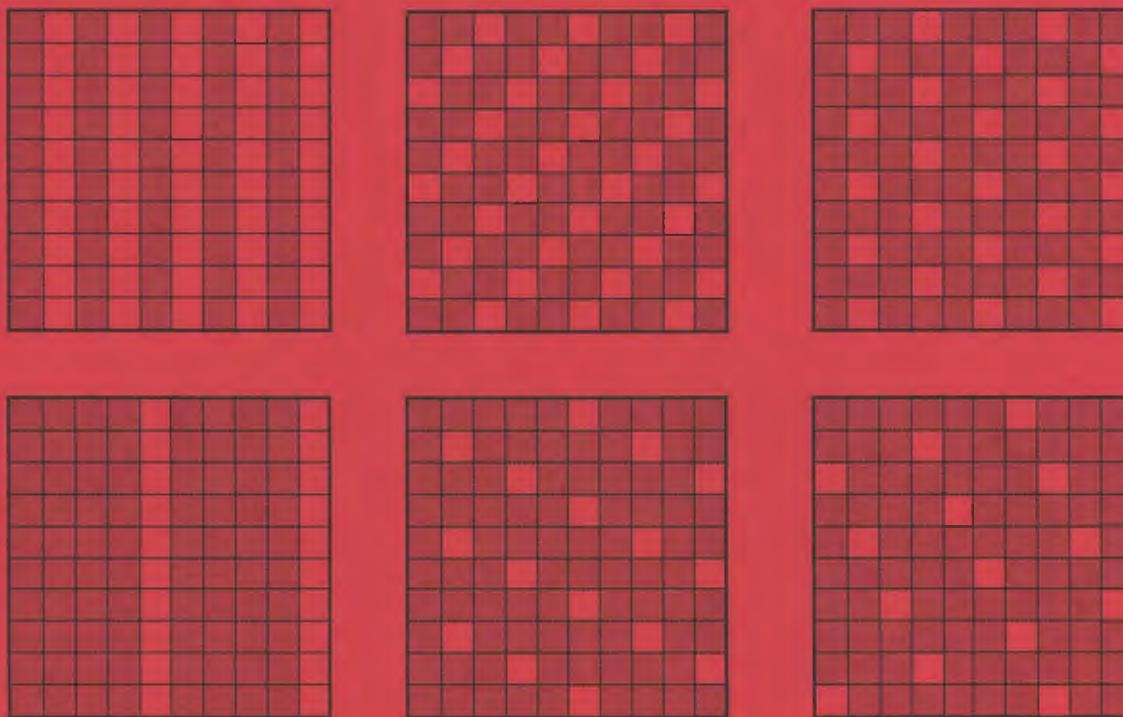


# LOGO EXCHANGE

Volume 14 Number 3

Spring 1996

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



**Fun with Hundred Board Patterns**



***In this issue:***

**Exploring Recursion  
Hundred Board Patterns  
Logo and Spatial Abilities  
A Circular Romp**

**Exploring Triangles  
Logo Action Research  
Parallel Processing  
Logo and the Internet**

**Editorial Publisher**

International Society for Technology in Education

**Editor-in-Chief**

Dorothy M. Fitch

**Copy Editor**

Jennifer Lindsey

**Founding Editor**

Tom Lough

**International Editor**

Dennis Harper

**SIG/Logo Officers**

Chuck Friesen, President  
Steve Sesko, Vice-President  
Hope Chaffin, Secretary / Treasurer  
Dorothy Fitch, Editor

**Production**

Kerry Lutz

**Director of Advertising Services**

Lynda Ferguson

**Submission of Manuscripts**

*Logo Exchange* is published quarterly by the International Society for Technology in Education Special Interest Group for Logo-Using Educators. *Logo Exchange* solicits articles on all aspects of Logo use in education.

Manuscripts should be sent by surface mail on a 3.5" disk (where possible). Preferred format is Microsoft Word for the Macintosh. ASCII files in either Macintosh or DOS format are also welcome. Submissions may be made by electronic mail as well. Where possible, graphics should also be submitted electronically. Please include electronic copy, either on disk (preferred) or by electronic mail, with any paper submissions. Paper submissions may be submitted for review if electronic copies are supplied upon acceptance.

**Send surface mail to:**

Dorothy M. Fitch  
3 Derby Road  
Derry, NH 03038

**Send electronic mail to:**

Internet: 71760.3666@compuserve.com

**Deadlines**

To be considered for publication, manuscripts must be received by the dates indicated below.

Volume 15, Number 1	Mar. 1, 1996
Volume 15, Number 2	June 1, 1996
Volume 15, Number 3	Oct. 1, 1996
Volume 15, Number 4	Jan. 1, 1997

**ISTE Board of Directors 1995-96**

David Brittain, President  
*MGT. of America (FL)*  
David Moursund, Executive Officer  
Dennis L. Bybee, Associate Executive Officer

**Executive Board Members**

Cornie Stout, Secretary  
*Texas Education Network (TX)*  
Barry Pitsch, Treasurer  
*Hearland AEA #11 (IA)*  
M.G. (Peggy) Kelly, Past-President  
*California State University - San Marcos*  
Terrie Gray, At Large  
*Chico Unified School District (CA)*

**Board Members**

Kathryn Edwards Price  
*Apple Computer, Inc., Santa Monica (CA)*  
Terry Killian  
*Jefferson County R1 School District (CO)*  
Paul O'Driscoll  
*Salem-Keizer Public Schools (OR)*  
Lynne Schrum  
*University of Georgia*  
Carla Schutte  
*Educational Technology Consultants (FL)*  
Gwen Solomon  
*U.S. Department of Education (DC)*  
Neal Strudler  
*University of Nevada - Las Vegas*  
Harriet Taylor  
*Louisiana State University*  
Brenda Williams  
*West Virginia Department of Education*

*Logo Exchange* is published quarterly by the International Society for Technology in Education (ISTE), 1787 Agate Street, Eugene, OR 97403-1923, USA, 800/356-5191. This publication was produced using Aldus PageMaker®.

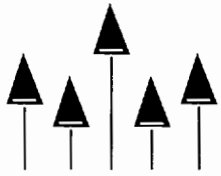
ISTE Members may join SIG/Logo for \$24.00. Dues include a subscription to *Logo Exchange*. Add \$10 for mailing outside the USA. Send membership dues to ISTE. Add \$4.00 for processing if payment does not accompany your dues. VISA, Mastercard, and Discover accepted.

Advertising space in *Logo Exchange* is limited. Please contact ISTE's director of advertising services for space availability and details.

*Logo Exchange* solicits articles on all topics of interest to Logo-using educators. Submission guidelines can be obtained by contacting the editor. Opinions expressed in this publication are those of the authors and do not necessarily represent or reflect the official policy of ISTE.

© 1995 ISTE. All articles are copyright of ISTE unless otherwise specified. Reprint permission for nonprofit educational use can be obtained for a nominal charge through the Copyright Clearance Center, 27 Congress St., Salem, MA 01970, 508/744-3350, FAX 508/741-2318. ISTE members may apply directly to the ISTE office for free reprint permission.

**POSTMASTER:** Send address changes to *Logo Exchange*, ISTE, 480 Chamblin St., Eugene, OR 97401-2626 USA. Second-class postage paid at Eugene OR. USPS# 660-130. ISTE is a nonprofit organization with its main offices housed at the University of Oregon. ISSN# 0888-6970



# Logo Exchange

Volume 14 Number 3

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

Spring 1996

## Contents

From the Editor: Spring Threads .....	<i>Dorothy M. Fitch</i>	2
Quarterly Quantum: Turtles? What Turtles? .....	<i>Tom Lough</i>	3
The Recurring Question of Recursion .....	<i>Jim Muller</i>	4
Hundred Board Patterns .....	<i>Ihor Charischak</i>	7
Using Logo as a "Building Block" to Improve Spatial Abilities in Primary Grade Students .....	<i>Nancy Flynn</i>	12
A Circular Romp with Logo Triumphant .....	<i>Robert Macdonald</i>	15
Turtle and Coordinate Geometry in a Secondary Environment .....	<i>Medhat H. Rahim</i>	22
Logo Action Research: Getting Teachers Excited About Technology .....	<i>Nan Youngerman and Donna Cutler-Landsman</i>	27
Recursion, the Problem Solver .....	<i>Jim Muller</i>	30
Windows on Logo: Logo and the Internet—The Turtle and the World Wide Web .....	<i>Glen L. Bull and Gina L. Bull</i>	33
Logo: Search and Research Parallel Programming .....	<i>Douglas H. Clements and Julie Sarama</i>	36





# From the Editor: Spring Threads

by Dorothy M. Fitch

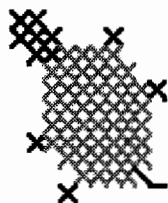
Three themes thread their way through this issue of *LX*: using Logo as a tool for exploring mathematical ideas, for investigating gender differences, and for approaching the latest technologies.

Exploring mathematics, one of the classic ways to use the power of Logo, is the main theme of this issue. Ihor Charischak takes us on a field trip to ancient Greece, where we use Logo to experiment with **times tables** and **factors**. Jim Muller helps remove the mystery from **recursion**. In his first article, he explains recursion with a great hands-on demonstration. His second article gives us a powerful example of putting recursion to work to solve a problem. Robert Macdonald shows how to get kids excited about **circles**. Dr. Rahim helps students explore **triangles** and learn what makes a triangle a triangle.

Also in this issue, we investigate gender differences in two different contexts. Nancy Flynn explores gender differences in spatial abilities using **building blocks** and Logo. Nan Youngerman and Donna Cutler-Landsman look at an action research project focused on gender differences using **robotics**.

To help us look toward the future, Glen and Gina Bull ponder the use of the **Internet** and **World Wide Web** as a distribution channel for Logo ideas, Doug Clements and Julie Sarama explore **parallel programming** and the new types of bugs we must learn to unravel, and Tom Lough ponders the future of **real live turtles** and tells how we can help their cause.

And speaking of turtles (and threads), I couldn't resist sharing this morsel with you.



I recently attempted teaching myself the art of counted cross-stitch. The first lesson in the pamphlet showed how to cross-stitch a turtle, so

naturally I took that as a good omen. But before taking a needle in hand, there I was at the computer, writing this little automatic cross-stitch program in Logo. For a challenge, modify it for other cross-stitch diagrams.

```
to cross.stitched.turtle
cleargraphics hideturtle
pu setx -50 sety 50 pd
; 0 is no stitch (an X with pen up)
; 1 is a light green X (my color # 11)
; 2 is a dark green X (my color # 12)
; B is a custom backstitch procedure
; starting from the upper left corner
row [2 2]
row [2 2 2 0 0 0 2]
row [0 2 2 1 1 1 1]
row [0 0 1 1 1 1 1 1]
row [0 0 1 1 1 1 1 1 2]
row [0 0 1 1 1 1 1 1 1]
row [0 0 1 1 1 1 1 1 1]
row [0 2 1 1 1 1 1 1 1]
row [0 0 0 1 1 1 1 1 1]
row [0 0 0 0 1 1 1 1 B]
row [0 0 0 2]
end
```

```
to row :list
if empty? :list [pu setx -50 sety ycor - 8
pd stop]
if first :list = 0 [pu x pd]
if first :list = 1 [setpc 11 x]
if first :list = 2 [setpc 12 x]
if first :list = "B [setpc 12 bkstitch]
row butfirst :list
end
```

```
to x
right 45 forward 10 back 5
left 90 forward 5 back 10 right 45
end
```

```
to bkstitch
pu right 90 forward 7 left 90 pd
left 45 forward 10 back 10 right 45
right 90 forward 7 left 90
end
```

Happy Logo adventures! ▲

**Dorothy M. Fitch**, *LX* Editor  
3 Derby Road, Derry, NH 03038  
Telephone: 603-425-2010 Fax: 603-425-6487  
E-mail: 71760.366@compuserve.com

# Quarterly Quantum: Turtles? What Turtles?

by Tom Lough

Kim: Knock, knock.

Pat: Who's there?

Kim: Leonardo, Donatello, Michelangelo,  
and Raphael

Pat: Leonardo, Donatello, Michelangelo,  
and Raphael who?

Kim: How quickly we forget.

The Teenage Mutant Ninja Turtles are history. Gone. Poof.

I have been thinking about turtles recently. No real surprise there. Perhaps because the TMNTs have faded so quickly I am more attuned to turtles in distress. Consider these items.

In a recent issue of the *Hartford Courant*, there appeared at the top of the front page an article about Joseph Ferrara of Wallingford, CT. Joseph had learned that a 14-year-old boy had caught two huge snapping turtles, each weighing about 90 pounds and estimated to be between 75 and 100 years old. The boy and his father had declared their intention to make turtle soup of their captives. Joseph tracked them down and persuaded them to sell the turtles to him for \$200. Joseph then released the turtles back into the wild at an undisclosed location.

Not long after that, I noticed a turtle in the road as I drove to work one morning. By the time I could turn around and drive back, another car had stopped by the turtle. A man got out of the car, picked up the turtle, and placed it down gently in a marshy area a safe distance from the road.

This motivated me to call my friend Carole Allen, founder and chairperson of Help Endangered Animals—Ridley Turtles (HEART). For years, Carole has focused her efforts on helping the Kemp's Ridley sea turtle avoid extinction. She has organized public awareness campaigns for the Kemp's Ridley and has developed educational material for school use.

While Carole was happy to hear my two stories of how turtles have been saved, she is

more concerned with the thousands of sea turtles that wash up on U.S. beaches each year. Many of these turtles had been needlessly drowned when they became entangled in shrimping nets. Carole helps to educate shrimpers about how they can use Turtle Exclusion Devices (TEDs) in their nets to give sea turtles a means of escape. Although thousands of TEDs are now in use, sea turtles are still endangered.

## How can we help?

Carole's most recent initiative is a Turtle-Safe Shrimp Campaign, in which shrimpers who agree to use TEDs properly will have their catch certified as Turtle-Safe. Concerned consumers are encouraged to ask their retail grocers and restaurant managers if they are using Turtle-Safe shrimp. Such a campaign can raise public awareness about the plight of the sea turtles and also encourage shrimpers, retailers, and consumers alike to become part of the solution.

Our Logo turtles have been around a long time. We have used them in both subject-specific and interdisciplinary activities. Here is an opportunity for our students to become more aware of the living, breathing variety of turtle and perhaps help to save a turtle or two.

For more information, write to Carole Allen, HEART, P.O. Box 681231, Houston, TX 77268-1231. She would appreciate a small contribution to help with the return postage if possible. You can also contact her through her online address: [heartallen@aol.com](mailto:heartallen@aol.com). ▲

## FD 100!

*Tom Lough*  
Founding Editor  
P.O. Box 394  
Simsbury, CT 06070

P.S. Do you have a story of a turtle being saved? I would love to hear about it!



# The Recurring Question of Recursion

by Jim Muller, The Logo Forum

---

Every Logo fan(atic) knows about recursion—where a procedure calls itself as an element of the solution. At least you would think so.

Beginning Logophiles often know about recursion. They can talk about it among themselves. They can point to lots of examples. But can they explain it to others? More importantly, can you explain it to a young child?

Here's a fun procedure to help you make some sense out of recursion, even though it really isn't Logo.

```
to get.through.life
get.through.today
get.through.life
end
```

That's cute. But think about it for a moment. The procedure says that to get through life, you have to get through today. Once you are through today, you have to move on with your life.

But where?

You cannot go backward. You cannot stop time. You have to get through life. But to get through life, you have to get through today. But each day is different. Wouldn't it be nice if you could save each day so that at the end, you had a record of your life. All you have to do is add a line to the life procedure.

```
to get.through.life
get.through.today
get.through.life
print book.of.life
end
```

Now, as you go through **get.through.life**, a page is written. When the procedure stops, it will print your book of life, starting from the most recent page back to the first.

To help make sense out of this, let's use a real example.

```
to test.recursion :n
print [Is this recursion?]
if readword = "YES [test.recursion :n + 1]
print :n
end
```

[Note: If your version of Logo doesn't include **readword**, use **first readlist** or **first request**.]

In this procedure, the variable **n** is used as a counter. It is used to keep track of your answers to the question, Is this recursion?

```
test.recursion 1
```

When you start the procedure, the first thing you see on the screen is the question,

```
Is this recursion?
```

**Readword** tells Logo to stop and wait for you to type an answer. Then comes the test. If the word you typed was YES, the procedure calls itself and the question is posed again.

```
Is this recursion?
```

Type YES a few times when you see the question and then type NO. When you type NO, the procedure comes to the test. This time the word you typed doesn't match the Logo word "YES" and so Logo reads the next line,

```
print :n
```

The result demonstrates what makes recursion different from just a simple loop.

When you first look at this procedure, it seems as if it were going to go around in a loop. Every time it passes the **test.recursion :n + 1** line, the counter is going to add 1. Then, when you type NO instead of YES, you'd think the procedure would simply print the current value of **n**.

Now the question becomes how to demonstrate this so that it makes sense.



## How recursion works, hands-on!

You'll need some blank paper, a pencil, and scissors for this demonstration.

Fold the paper in half. Then fold it in half again, and again, then one last time. Crease the edges, then open up your piece of paper. Cut the paper along the folds. You should end up with 16 small pieces of paper. Now number these "pages" from 1 to 16 by writing a small number at the bottom of the page.

Get your pencil ready and type:

```
test.recursion 1
```

First, you see the question: Is this recursion? So write a big 1 on your first piece of paper and put that piece off by itself.

Type YES. The question appears on the screen again and  $n$  becomes  $n + 1$ , or 2. Write 2 on your second piece of paper and put that on the pile with your first page, the one with the 1 on it. Type YES again. Write 3 on the next piece of paper and put that piece on the number pile. Do this three more times, writing the new number for  $n$  each time. Put each piece of paper on the top of your growing pile of papers.

Now, when you type NO, what happens on the screen? You see a list of numbers counting backward from 7 to 1.

Why? Look at the screen. There are seven answers displayed there. You typed YES six times and NO once. You typed seven answers.

You have two stacks of paper now. You should have some blank pages left over in one stack, pages 8 to 16. The other stack has the pages you numbered from 1 to 7.

The stacks of paper are like the computer's memory. Each time you answered YES, Logo filed the current procedure in memory. The current procedure carried the current value of  $n$ , referred to in the procedure as  $n$ . When you answered NO, Logo went to the memory stack and asked for all the values of  $n$  and displayed them on the screen. In doing so, Logo emptied the memory stack.

This is like putting your pages back in order.

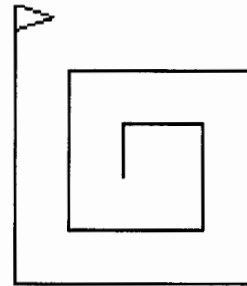
To demonstrate this, erase the 7 from page 7 and put it on top of page 8. Put page 6 on top of page 7, page 5 on top of page 6, and so on until you have all the pages back in a single stack again, erasing the large number each time.

## Tail-end versus embedded recursion

Another way to demonstrate recursion is to use the following two procedures. The first uses tail-end recursion, where the recursive call is at the end of the procedure. The second procedure does much the same thing, only the recursive call is embedded in the procedure.

The **maze** procedure definitely looks like a simple loop that stops when the value of  $n$ , or  $n$ , is greater than 100. But is it?

```
to maze :n
  if :n > 100 [stop]
  forward :n right 90
  maze :n + 10
end
```



When you type **maze 20**, the turtle moves **forward 20 right 90** and then the procedure calls itself. Only  $n$  is now 30. The procedure continues with  $n$  adding 10 each time until it becomes greater than 100. Then it stops.

Now take a look at an example of "embedded" recursion. Will the **amaze** procedure below produce the same picture as the **maze** procedure, or will it be different? Try to picture what will happen before you run it.

```
to amaze :n
  if :n > 200 [stop]
  amaze :n + 10
  forward :n right 90
end
```

Think about how recursion works, about how Logo reads and acts on procedures. Start with  $n$



equal to 20. This is how Logo reads the procedure the first time.

```
to amaze 20
if 20 > 100 [stop]
; Since 20 is smaller than 100, Logo goes to
the next line.
amaze 20 + 10
; The next line is written to memory as
"amaze 30" is called.
forward 20 right 90
end
```

Next, you have:

```
to amaze 30
if 30 > 100 [stop]
; Since 30 is smaller than 100, Logo goes to
the next line.
amaze 30 + 10
; The next line is written to memory as
"amaze 40" is called.
forward 30 right 90
end
```

Each time Logo runs the procedure, it never gets the chance to run the last line. So it puts each last line on a "page" of the memory stack. It will keep going, adding pages for each line it did not run from **amaze 40** through **amaze 100**. Then it stops.

Then Logo reads and executes the memory stack: forward 100 right 90, then forward 90 right 90, forward 80 right 90, and back to where it started, forward 20 right 90.

Are the pictures produced by **maze** and **amaze** the same? The end results look the same. The methods demonstrate what makes tail-end recursion different from embedded recursion.

Before we leave the discussion of tail-end versus embedded recursion, look at the **maze** procedure again.

```
to maze :n
if :n > 100 [stop]
forward :n right 90
maze :n + 10
end
```

What will happen if you change the second line to read:

```
if :n > 100 [print :n stop]
```

What will be printed at the end? Why?

This procedure does not result in a long list of numbers being printed. As it is run, every step is executed. There is nothing left over to hold in memory. At the end, Logo will print 110 and stop. Why 110? Only when **n** is greater than 100 does the procedure stop.

## Putting recursion to work

Fractals were once thought to be math monsters. No one could figure out what to do with them. But thanks to computers, we now know that these recursive monsters help make beautiful computer graphics.

Here is a very short procedure. It certainly looks simple enough. But how can it produce such a complex graphic? Type **c 5 10** and take a look.

```
to c :size :level
if :level = 0 [forward :size stop]
c :size :level - 1 right 90
c :size :level - 1 left 90
end
```

The **c** curve is fun to watch. But how does it do what it does?

Start off with **c 5 0**. Because **:level** equals 0, the turtle moves forward 5. Once I get beyond this, I need a pad of paper to keep track of what is happening. Those ubiquitous sticky-back notes on the board are a big help. By the time you get to **c 5 3** or **c 5 4**, you will have a lot of notes on the board—and a pretty good idea of how recursion works.

Special thanks to Dr. Kathleen Martin, co-author of *The Turtle's Sourcebook*, for her stack-of-paper demonstration. It made that proverbial light go on so I could begin to see what recursion was all about. ▲

**Jim Muller** has had a lifelong interest in translating various technologies into understandable and persuasive programs. In 1981, Muller and his son organized the first Logo users group, the Young Peoples' Logo Association, which eventually grew into a worldwide 6,000-member organization. Among other things, the YPLA published *1,2,3 My Computer and Me* and *The Turtle's Sourcebook*. Both of these books are now being expanded for use with current versions of Logo. In 1985, the YPLA merged with CompuServe, where it became The Logo Forum. Today, Muller is a computer training and marketing consultant in the Dallas/Fort Worth metroplex. You can reach him via e-mail at 76703.3005@compuserve.com or on CompuServe at 76703,3005.





# Hundred Board Patterns

by Ihor Charischak

[Editor's note: This article is reprinted with permission from CLIME (Council for Logo & Technology in Math Education). It was originally published in 1990 in *Clime Microworlds: Volume II*, a collection of Logo-based math activities. The Logo program provided here is a revision of the original.]

Children enjoy games of "Let's Pretend." Here is an activity that involves children in a historical fantasy.

One of the ways students can have a meaningful learning experience is to go on a math field trip to a place many of the ideas that we study in math were first discussed. That place is Ancient Greece. Now that would be quite a trick, given that Ancient Greece hasn't been around for over 2,000 years. So how can you possibly go back in time? Why, you would take the trip in a time machine!

To get students ready for this adventure, you might start by distributing to your class the following letter:

Dear student,

You are about to embark on a journey back in time. Your destination will be Ancient Greece. When you arrive, you will join an affluent merchant's family as an adopted son or daughter. Being rich in Greece means you don't have to do any back-breaking work that would take up all your time. So you can spend your idle hours on contemplation and study, which, of course, you do not have to do. But before you plan to head for the beaches with your ice cream and candy in hand, you must remember one thing: When in Greece, you will do as the affluent Greeks do, which is to pursue intellectual challenges of the mind! You do have one advantage over the Greeks. You have your computer with you. Your guide will be Eratosthenes. You can call him Beta for short.

Regards,  
Your teacher

Next have your students take home and have signed a parent permission form.

Dear school,

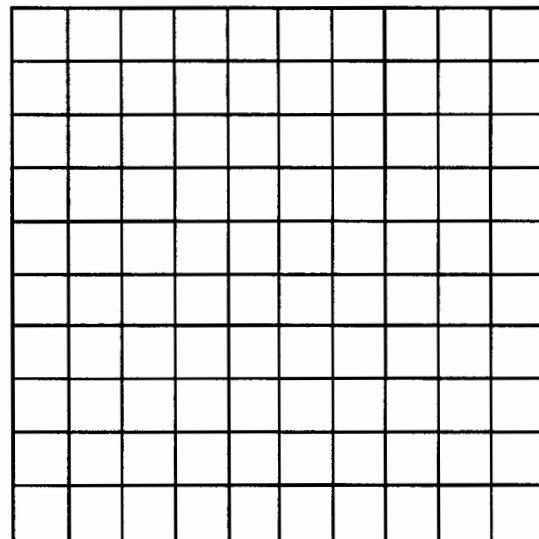
I hereby give permission for my son/daughter  
to travel to Ancient Greece.

I waive all rights to sue the school if my son/  
daughter gets lost in a time warp.

Sincerely,  
A brave parent

So having agreed to the mission and permission granted, the students climb into the time capsule and take off. The machine is fast, and before too long you arrive at your destination (200 BC). There standing in front of the machine waiting to greet you is Beta. He takes everyone into his house and begins.

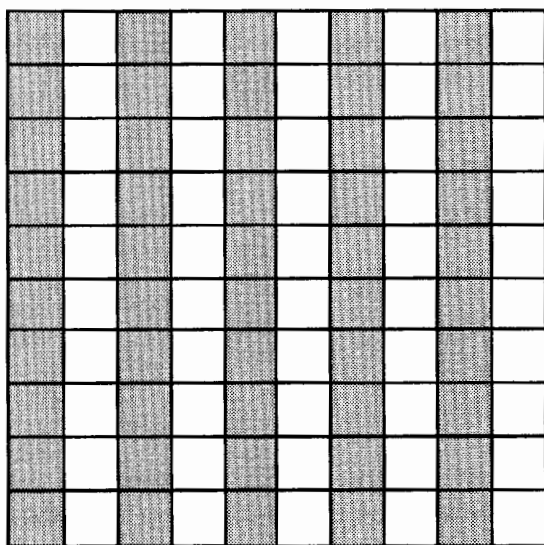
"Hi there! Beta's my name and math is my game. Let's start off with something that's fun and easy. We are going to explore patterns on the hundred boards."



A 100 Board



Beta then passes out hundred boards made out of wood to all his guests. He next asks a question, "Do you know what numbers belong to the 2 times table or club?"



The 2 Times Table

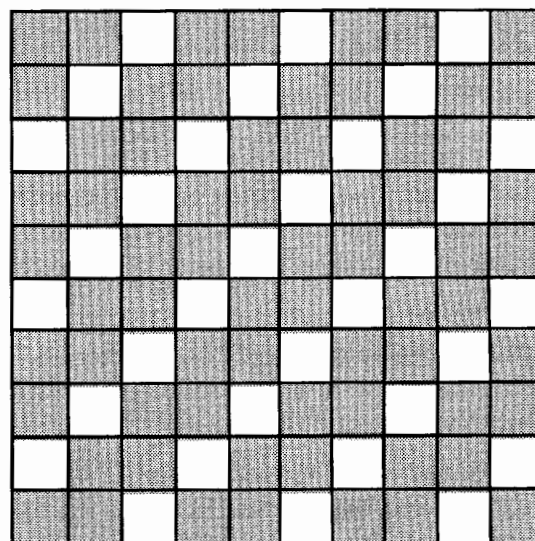
"Sure," says one of the students. "Those are all the even numbers."

"Yes, that's it! Now use a cutting tool to cut out the even squares. When you finish you will have a family portrait of the two club."

Now Beta was noticing something odd. Some of the students brought with them a strange machine. "What is that?" he asked.

"Why, it's a computer and I can do this hundred board activity with it. Watch," responded a student.

The student calls up 100BOARD on the disk, then types **grid** followed by **board 3** to create the 3 times table. Beta was intrigued. He grabbed the computer from the student.



The 3 Times Table

"I'm going to make the 3 times table," he said. Beta and his guests continued until they had completed the 2, 3, 4, and 5 tables.

"I suppose you want us to make the 6 times table as well," asked a student.

"Yes, but you don't have to make one. You can create it from the boards you already have," Beta responded.

He then placed his wooden 2 board on top of the 3 board and held them up to the light. Sure enough, only the 6s emerged.

A student then tried the same experiment on the computer. He cleared the screen and typed **board 2** followed by **board 3**. As expected, the 6s appeared.

Beta continues, "Can you tell me what will happen if you put a 3 board on top of a 5 board?"

"That's easy," said a student, "it's the 15 times table."

"Good. You think you've got the pattern, don't you? Well, what about the 3 table on top of the 6 table?"

The students were surprised to find out that it was *not* the 18 table! It turned out to be the 6 table. They scratched their heads and went back to the drawing board to try and figure out why.

Can you help them? Make the "boards" on the computer and see if you can predict what hap-

pens. Extend your guessing to three boards. For example, what happens if you put the 2s on top of the 3s on top of the 8s? What boards would you use to make the 12 times board or the 18 times board?

### The secret of the 100 boards revealed...

You may have already figured out that the specific times table that results is the least common multiple of the tables that created it. So if you should place board 4 on top of board 6 and board 8, you should see the 24 times table.

### Creating a 100Board program

Presented here are sets of procedures for two different methods of creating times tables. (These procedures were written using Terrapin's *Logo PLUS for the Macintosh*, but should work with most versions of Logo with little modification.)

Using Method A, students will determine which boxes should be erased to create a times table. Method B is an automated process that doesn't require students to generate the times tables themselves. Choose whichever method you wish to use with your students, depending on your goals and available time (Method B is handy when time is limited).

#### Method A

After entering the following procedures, type **grid** to display the 100 x 100 grid. Then show your students how to edit the **setup** procedure to make one that will create the times table for any number, as explained below.

```
to grid
penup setpos [-100 -100] pendown
repeat 4 [forward 200 right 90]
repeat 10 [forward 200 back 200 right 90
  forward 20 left 90]
penup setpos [-100 -100] pendown
right 90
repeat 10 [forward 200 back 200 left 90
  forward 20 right 90]
penup setpos [-100 -100] pendown
left 90
penup setpos [-100 80] pendown
end
```

```
to do :list
if empty? :list [nextrow stop]
if first :list = "X [fillsq]
nextsq
do butfirst :list
end
```

```
to fillsq
; use the necessary commands for your
; Logo to fill a 20x20 square
; a quick Terrapin/Mac shortcut is:
; (stamprect 20 20 "true)
setpc blue ; or any color you choose
fillsh [repeat 4 [fd 20 rt 90]]
setpc black
end
```

```
to nextsq
penup
right 90 forward 20 left 90
pendown
end
```

```
to nextrow
penup
left 90 forward 200 right 90 back 20
pendown
end
```

After entering the "housekeeping" procedures listed above, show your students how to edit the following **setup** procedure to make a procedure for any times table.

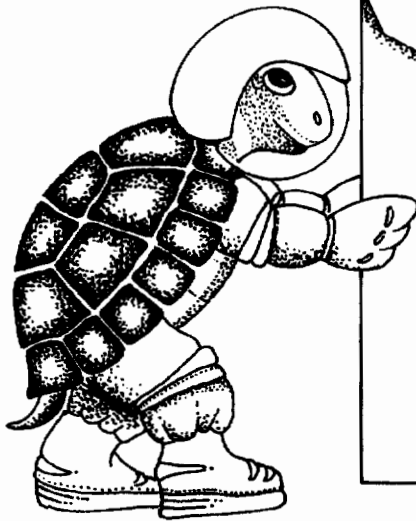
The **setup** procedure calls the **do** procedure 10 times, with an input that specifies which boxes should be erased in each row in order to draw the correct table. An X stands for a filled square. An O stands for an open or empty square.

Give the students this starting **setup** procedure:

```
to setup
penup setpos [-100 80] pendown
; X for filled, 0 for open (empty) box
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
do [X X X X X X X X X X]
grid
end
```

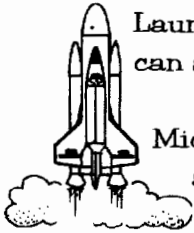






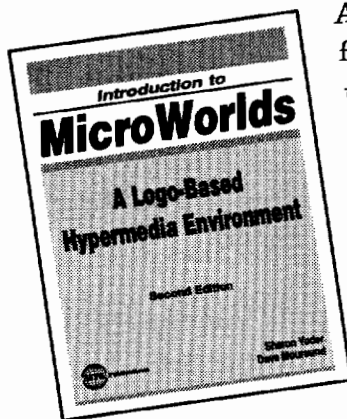
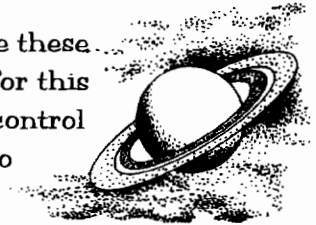
## Push the envelope with MicroWorlds

Launch your students toward stimulating new classroom challenges and adventures. They can soar to a galaxy of new skills using MicroWorlds.



MicroWorlds is an exciting application that offers in a single program many of the features available in HyperCard, Kid Pix, and LogoWriter.

You're the pilot. Get ready now to grab the controls, blast off, and explore these exciting new worlds with your class! To help you prepare your crew for this fantastic voyage, Sharon Yoder and Dave Moursund back at mission control have developed a 222-page flight manual titled *Introduction to MicroWorlds—A Logo-Based Hypermedia Environment*.



As a computer application, MicroWorlds contains a wide range of features that make it easy and fun to work with color graphics, sound, text, and animation. As a programming environment, MicroWorlds includes a powerful and modern version of the Logo programming language.

It's time to broaden your universe. The countdown is underway. Have a nice trip!



**International Society for Technology in Education**  
*Customer Service Office*  
480 Charnelton Street, Eugene, OR 97401-2626 USA  
Phone: 800/336-5191 (US and Canada)  
Phone: 541/302-3777 (International)  
Fax: 541/302-3778 Fax-on-Demand: 800/891-0212  
America Online: ISTE CompuServe: 70014,2117  
Internet: iste@oregon.uoregon.edu  
World Wide Web: <http://isteonline.uoregon.edu>

HYPERCARD® IS A REGISTERED TRADEMARK OF CLARIS CORPORATION, SANTA CLARA, CA, U.S.A. KID PIX® IS A REGISTERED TRADEMARK OF BRØDERBUND SOFTWARE, NOVATO, CA, U.S.A. LOGO®, LOGOWRITER® AND MICROWORLDS® ARE REGISTERED TRADEMARKS OF LOGO COMPUTER SYSTEMS, INC., HIGHGATE SPRINGS, VT, U.S.A.





# Using Logo as a "Building Block" to Improve Spatial Abilities in Primary Grade Students

by Nancy Flynn

"Can we do *Kid Pix*? Yeah, I want to do *Kid Pix*!" came a chorus of first graders upon my first meeting with them in the computer lab.

"No," I said casually. "We're going to learn Logo."

"Logo? Is that a new computer game? What do we do with it?" they asked.

Typical of first graders, they were full of questions and many doubts, as this was something brand new. I began the first lesson with a story that I wrote about the Logo turtle, to introduce what we would be doing for the next four weeks.

I then proceeded to introduce programming using *KidTool*, which is included with *Primary LogoWriter for the Macintosh* from Logo Computer Systems, Inc. (LCSI). This program very closely resembles a toolkit I wrote for kindergartners and first graders a few years ago. We began with the basic commands of F for forward, B for backward, R for right turn, and L for left turn. All the children had to do was type a letter, a space, and any number from one to ten, which would be multiplied by ten in *KidTool*.

## Turtle orientation

After my first session of Logo with the first graders I was a bit discouraged. I found that in most cases, the girls had a lot more difficulty with the spatial concept of turtle orientation, and in general, taking charge of the computer. Most of the girls needed constant attention to stay on task. They asked a lot of questions relating to orientation, the most often of which was "How do I make the turtle go down? Is it D for down?"

For example, a child would get two sides of a square ending with the turtle heading 90 degrees. By starting with the turtle in the home position, the child would be able to program **F 4 R 90 F 4** because the first right turn is visualized as a right turn when initially heading at 0 de-

grees. The next logical move would be **R 90** again, but a right turn was not what most of the children, especially the girls, could comprehend because it looked to them as if the turtle had to turn up or down instead of right or left, as it had in the first turn.

I would then explain that there were only four commands that we would be using to move the turtle and I would ask the children to recite those four commands. After I heard F, B, R, and L, I would ask the child who asked me how to make the turtle go down which of those four commands he or she would use. This was a very puzzling concept for many of the students because they could not see how turning the turtle right or left would make the turtle go down. I used the analogy of riding a bike. I asked the children what they would do if they were riding their bike and came to the end of the street. Their responses were either turn right, turn left, or turn around and go back. By using analogies of turning I was trying to get the children to think in terms of turns for heading in the desired direction.

I devoted a considerable amount of time teaching turtle orientation through both physical movements of the children themselves and turtle moves because this concept of spatial orientation was frustrating for so many children. I used a maze and a target practice in which the students tried to "hit" a second turtle, had them draw squares and rectangles, and had them trace over pre-programmed shapes.

## Spatial abilities: girls versus boys

After a few sessions I began to see a change in the children's abilities to work independently, figure out the spatial orientation on their own, and take charge of the computer. I especially saw a very significant change in how the girls approached programming. They began to feel

more comfortable working independently and with peers without constantly asking for help.

I began to look for explanations as to why there was such a difference in the spatial abilities between the boys and the girls. By observing how they worked through their programming challenges I decided that it could be based on how much time the children spent on spatial activities outside of school, such as building with blocks.

### Logo building blocks

Given this assumption, one Logo challenge I gave the children, once they had mastered drawing squares and rectangles, was to "build" with them. At first I noticed that most of the boys had no trouble building with the "blocks" they drew with the turtle, while the majority of the girls had trouble deciding how to build with their Logo blocks. (*LogoWriter for the Macintosh* allows the user to place the arrow on the turtle and, by clicking the mouse button, move the turtle anywhere on the screen without drawing a line, thus eliminating the **pu setpos pd** steps.)

### Results of surveys and research

With my new hypothesis linking building experience with spatial ability, I sent a survey form home with the children asking their parents to indicate approximately how many hours per week their child spent building with blocks, Legos, Tinker Toys, and/or Lincoln Logs prior to entering first grade. My assumption of linking building experience with spatial ability proved to be reasonably accurate. I found that, on average, the female child spent no more than two to four hours per week engaged in a building activity, whereas the male child spent a minimum of four to seven hours per week. Some parents indicated that their male child spent more than ten hours per week building.

In her book *Sex Differences in Cognitive Abilities*, Diane Halpern cites a study conducting a direct test of the hypothesis that boys develop better spatial skills because of their experiences with spatial activities. In this study, Sprafkin, Serbin, Denier, and Connor (1983) gave 3½- to 4-year-old children special training with pre-

school toys that are typically preferred by boys, including blocks, dominoes, Tinker Toys, and paper cut in geometric shapes. They believed that these typical "boys' toys" were important in the development of spatial ability. They found that the children who received the special training with these toys scored higher on a test of spatial ability than a control group of children who did not receive the special training. From this study it seems clear that spatial ability can be taught, and that it can be taught to preschool children by increasing their exposure to typical male-preferred toys. While I do not consider Logo a male-preferred "toy," I do advocate using it as a tool to improve spatial ability in all children in the primary grades.

In another study, Schackman and Serbin (1978) tested first-grade children using a special children's version of the Embedded Figures Test, a spatial ability test that sometimes (but not reliably) produces sex differences. They found the usual sex difference, with the boys performing more accurately than the girls. They then provided both the girls and the boys with an "overlay training procedure" in which they used transparent overlays to highlight the way smaller figures were embedded within the boundaries of the larger figures. They found that after this training both girls and boys improved significantly. Most importantly, they found that the sex differential disappeared after the training. They believe that the reason the girls benefited more from the training than the boys is because the girls typically have less experience with spatial activities and therefore would improve more with specific instruction.

The results of these studies provide a strong case for the importance of sex differentiated environmental factors in the development of spatial skills. They also demonstrate that spatial skills are trainable. Because not all children are exposed to toys that enhance the development of spatial abilities, I believe that exposing young children to programming can be very beneficial to their learning development. Logo already applies skills in math, science, and language, and it is actually improving spatial skills as well.



## Final results

At the end of four weeks I asked the children to come up with their own programming challenge. I kept a copy of each child's work and when I displayed their work on the wall of the lab it was impossible to distinguish pictures done by children who caught on quickly from those done by children who struggled at the beginning. I'm convinced that those who struggled early on with turtle orientation have improved spatial abilities because of their Logo programming experience.

On the last day I told the children they could do *Kid Pix*. Shouts of glee rang out. They all selected *Kid Pix* from the menu, and within 10 minutes, most were bored, asking if it was all right to do Logo. "I guess so," I said, grinning ear to ear. ▲

**Nancy Flynn** is a computer specialist in the Saint Paul (Minnesota) Public Schools. She is currently writing a book on multicultural projects using *LogoWriter* and how they can be integrated into the classroom curriculum.

## References:

- Halpern, D.F. (Ed.) (1992). *Sex differences in cognitive abilities*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc. Publishers.
- Sprigle, J.E., & Schaefer, L. (1984). Age, gender, and spatial knowledge influences on preschoolers' computer programming ability, *Early child development and care, Vol. 14*, pp. 243-250.

# Logosium '96

St. Anthony Park Elementary School  
St. Paul, Minnesota  
Friday, June 14, 1996

## Call for Participation

The third annual Logosium will be a day of Logo workshops, discussions, and presentations hosted by the world renowned St. Paul Logo Project. This year Logosium will include sessions conducted by students sharing their projects with other students and with adults.

Sessions may be one-hour presentations or panel discussions, or two-hour, hands-on workshops on any topic of interest to the Logo community.

If you wish to offer a session, submit a one-page description of what you have in mind to:

Marian Rosen & Michael Tempel  
c/o Logo Foundation  
250 West 57th Street, Suite 2228  
New York, NY 10107-2228  
Ph: 212/765-4918  
Fax: 212/765-4789  
e-mail: mbrosen@oui.com  
michaelt@media.mit.edu

**The deadline for submissions is  
March 1, 1996**

For registration and hotel information contact:  
NECC '96  
1787 Agate Street  
Eugene, OR 97403-1923  
Ph: 503/346-2834  
Fax: 503/346-5890  
e-mail: necc@oregon.uoregon.edu

Logosium is sponsored by the Logo Foundation and ISTE's SIGLogo.



# A Circular Romp with Logo Triumphant

by Robert Macdonald

---

This article details an educational adventure with fourth-grade students enrolled in two elementary schools. The school system has been investigating the possible educational benefits that might accrue with the philosophy of cooperative learning. With this in mind, a mathematics teacher in a separately housed high school decided that one of her second-year unified math classes (comprising algebra, geometry, statistics, and probability) might investigate the value of pi ( $\pi$ ) with fourth graders.

One of the fourth-grade teachers asked me to suggest ways in which she might prepare her students for the learning experience. I suggested that she collect compasses and straight edges. With these, the students would construct circles and investigate the relationship of the circumferences of their circles to the diameters and radii. As it is sometimes difficult for fourth graders to manipulate compasses, I suggested that she use tongue depressors in which some holes were drilled. Use one of the holes as the center of a circle by holding it in place with a pencil or pin. One of the other holes could be controlled with a pencil, and the circumference could be sketched out by moving the pencil in a circular path.

## Some hands-on activities

I have always delighted in introducing fourth graders to constructions with a compass and straight edge. Perhaps you might introduce such basic geometric constructions as:

- making a line segment equal to a given line segment
- bisecting a line segment
- constructing an angle equal to a given angle
- bisecting an angle
- constructing a line parallel to a given line through a given point off the line
- erecting a perpendicular to a line from a point on the line
- dropping a perpendicular to a given line from a point off the line.

With these basic constructions, students gain some control over their tools and are empowered to build a number of interesting geometric figures.

## Measuring what we have drawn

Having drawn a circle with a compass and inserting a diameter with a straight edge, measure the distance of the circumference with a string. Using a metric ruler determine this distance in decimeters. Now measure the diameter with that string. Again determine the length in decimeters.

With a calculator divide the length of the diameter into the length of the circumference. What is your answer? Compare your answer with others in the classroom. You will note something of great interest.

Now take the fraction  $22/7$ . This improper fraction indicates that 22 is divided by the 7. What is the answer? With all of the values, you have uncovered the value of pi ( $\pi$ ). It is the relationship of the diameter of any circle to its own circumference.

As the students work through these activities, they should become more and more familiar with the terminology applied: circumference, diameter, radius, and pi. It is an awareness that has been brought about through active participation in the activities.

## A pi day

The high school math class arrived at noon at the first elementary school they were to visit. To accommodate scheduling, the high school students had missed their lunch period. They quickly set up shop with 10 workstations suitably labeled and staffed. Within an hour they were able to service two fourth-grade classes, giving the young students a number of work sheets and handouts in addition to treats in the form of slices of pizza pie. (Leave it to a group of teenagers to dream up a yummy conclusion.)

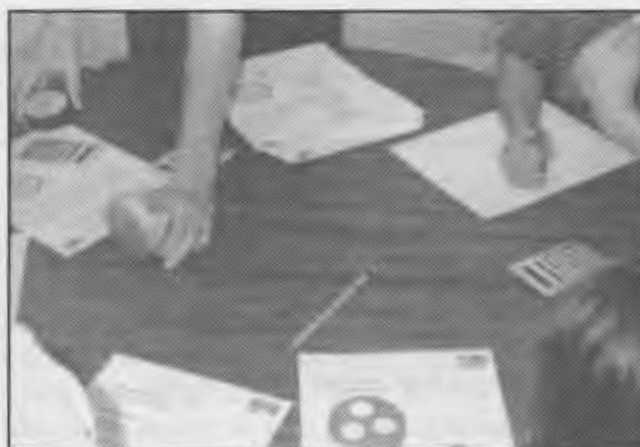




*A typical selection of circular material*

It was interesting to note the diversity of the offerings at the different stations. The 10 booths provided:

- a review of the concepts of circumference, radius, and diameter
- ample visual aids reviewing the areas of variously sized rectangles
- explorations involving the parts of a circle, then an introduction to the method of determining its area
- an examination of the Greek and English alphabets (this could have been a wonderful Logo project)
- a vast number of balls and much string for measuring many circumferences
- a comparison of the shape of a circle to the shapes of various polygons
- a circle word search
- hands-on experiments for finding the volume of a cylinder
- jelly bean counting contests (all of the jelly beans were in circular jars)
- a student-generated video in which Euclid, the father of geometry, appeared from the mists of time detailing the need for earth measurements.



*Let's get down to calculating!*

To focus interest, the high school students were costumed and bore name tags providing them Grecian identities, such as Euclid, Eratosthenes, and Aristotle. I enjoyed the presentations as much as the fourth graders. I also recognized a number of former students who seemed to enjoy their teaching tasks.

Several days later one of the fourth-grade teachers and I discussed the effectiveness of the presentation. She thought that, at least for the short term, her students had derived benefits from the visit, although, she cautioned, she didn't know how long it would last or if the students would be able to expand on the experiences on their own.

During our conversation a thought entered my mind: Why hadn't my former students applied computer-generated programs to forward their investigations? The answer was simple enough: They had not used Logo since I had had them in the fourth grade, and the school in which they were guests had never used Logo in its instructional programs, even though the school has a computer lab with 14 Apple IIGSes and two Apple IIs in each classroom.

One factor was in our favor. The second elementary school that the high schoolers were to visit did use Logo as an instructional tool. With that in mind, I got together with one of the fourth-grade teachers at that school to see how we could further her students' investigations.



## Logo to the rescue

After first engaging in the hands-on activities described above, the students were soon able to discover that there are different ways of expressing the size of the same circle. There is a fixed relationship between the measurements of a circle. Therefore the students could define the size of the same circle by providing the circumference, the diameter, or the radius.

In their earliest experiences with the turtle, all the students had constructed a circle with the following Logo program, where the circle is drawn with the turtle moving on the circumference.

```
to circle
repeat 360 [forward 1 right 1]
end
```

We can supply programs based on this model that will draw circles using the value we provide for the circumference, the diameter, or the radius.

## Drawing circles on the circumference

All of the following programs were written in the Macintosh version of *LogoWriter*. If you do not have a primitive that reports the value for pi, you can write the following tool procedure:

```
to pi
output 3.1416
end
```

Place the following programs on a page that you name **circle.circum**, denoting that the circles are drawn directly from the circumference. (Later in the article we will look at programs that draw circles from a center, as you would expect to do with a compass.)

The programs are very straightforward and should be easily understood by all students who have done the work described previously. The command **circle.c** requires the input of the value of the circumference. The command **circle.d** requires the input of the desired diameter. The command **circle.r** requires the input of the desired radius.

```
to circle.c :circumference
repeat 360 [forward (:circumference / 360)
right 1]
end
```

```
to circle.r :radius
repeat 360 [forward (:radius * 2 * pi) / 360
right 1]
end
```

```
to circle.d :diameter
repeat 360 [forward (:diameter * pi) / 360
right 1]
end
```

One of the first tasks the students undertook was to see if they could construct identical circles by varying the program used. If students thoroughly understand the relationship of the diameter to the circumference, they will have little difficulty and should enjoy themselves immensely.

Early on, the students decided to change the color of their turtle with a **setc** command so that the circumferences generated with each program could be differentiated.

Let's play around with these programs. In the Command Center enter the following commands to turn the turtle (and its pencolor) blue and draw a circle:

```
setc 2
circle.c 300
```

The turtle draws a circle with a circumference of 300 turtle steps.

We wish to enter a diameter that will create a circle whose circumference will be drawn on top of our previous circumference. If we divide 3 (an approximation for the value of pi) into 300 (the value of our circumference), we will get the value of 100 (an approximation for the diameter). These commands change the color to green and draw the circle.

```
setc 3
circle.d 100
```

Our estimate is close. However, the green circumference is a trifle too big. Now let's try the input of a radius. We will divide 6 into the circumference of 300 or divide 2 into the diameter of 100. The estimate of the radius with either calculation is 50. First change the color to fuchsia. Then enter the command with the input of the radius.



```
setc 6
circle.r 50
```

The fuchsia circumference fits right over the green. Our estimate based on our calculations was quite good. But why not have Logo do the mathematics? Try:

```
setc 2
circle.c 300
setc 3
circle.d 300 / pi
```

Our input has caused the circumference to be divided by the value of pi.

```
setc 6
circle.r (300 / pi) / 2
```

We have taken the previous input and divided it in half.

You might need to do some preliminary work on the order of operations with fourth graders. But I have found that they do not consider it a difficult concept. Now try some constructions on your own.

Because people do make mistakes, the students wanted to be able to erase them. So back to the drawing board. We altered each of the three procedures provided above with a penerase (**pe**) command. (The command **pd** cancels the penerase mode.) The following procedure shows the changes to the procedure **circle.c** to make an erasing procedure. You can change the other two procedures using this as a guide.

```
to erase.circle.c :circumference
pe
repeat 360 [forward (:circumference / 360)
  right 1]
pd
end
```

For example, give the command:

```
circle.c 200
```

Now erase it with the command:

```
erase.circle.c 200
```

## Drawing circles from a center

Within a very short time, some students wanted to draw circles on the screen in the same fashion as they had drawn circles with a compass. It did not take long to decide how that could be accomplished. We put these programs on a page named **circle.center**.

The commands for these procedures, listed below, are similar to those provided for circles drawn from the circumference. However, we add one extra feature: we print out the values of the circumference, radius, and diameter for each circle as a form of mathematical reinforcement.

```
to circle.c :circumference
make "diameter (:circumference / pi)
make "radius (:diameter / 2)
pu forward :radius pd
right 90
repeat 360 [forward (:circumference / 360)
  right 1]
left 90
pu back :radius pd
print.out
end
```

```
to circle.d :diameter
make "radius (:diameter / 2)
make "circumference (:diameter * pi)
pu forward :radius pd
right 90
repeat 360 [forward (:diameter * pi) / 360
  right 1]
left 90
pu back :radius pd
print.out
end
```

```
to circle.r :radius
make "diameter (:radius * 2)
make "circumference (:diameter * pi)
pu forward :radius pd
right 90
repeat 360 [forward (:radius * pi * 2) / 360
  right 1]
left 90
pu back :radius pd
print.out
end
```

```
to print.out
ct
(print [The Circumference is] :circumference)
(print [The Diameter is] :diameter)
(print [The Radius is] :radius)
end
```

Experiment with these new procedures. For example:

```
setc 11
circle.c 350
```

You quickly notice that the turtle moves from the center of the circle and draws a circumference of 350 turtle steps and returns to its center. In addition, you note that the following information is displayed:

```
The Circumference is 350
The Diameter is 111.4085
The Radius is 55.7042
```

If you wish to take advantage of the **print.out** procedure provided in the previous programs, you have the information for the inputs for the programs that use the diameter and the radius. If you do not wish to be provided with this aid, remove the instruction **print.out** from each program. However, its inclusion offers some interesting advantages. For example, enter:

```
setc 46
circle.d 111.4085
```

(We have borrowed the value of the diameter from the printout above.)

A red circumference is drawn and you get a printout on the screen of

```
The Circumference is 350.0001
The Diameter is 111.4085
The Radius is 55.7043
```

Look at the numbers carefully. Logo is doing its math precisely. Now enter:

```
setc 15
circle.r 55.7042
```

(We have borrowed the value of the radius from the first printout.)

A yellow circumference is drawn and we note the following information:

```
The Circumference is 349.9998
The Diameter is 111.4084
The Radius is 55.7042
```

Notice that numbers may vary slightly because of computer rounding algorithms.

I discovered that the students found this type of experimentation fascinating. Many of them took advantage of the opportunity.

We then provided procedures to erase any errors.

```
to erase.circle.c :circumference
make "diameter (:circumference / pi)
make "radius (:diameter / 2)
pu forward :radius pe
right 90
repeat 360 [forward (:circumference / 360)
  right 1]
left 90
pu back :radius pd
ct
end
```

```
to erase.circle.d :diameter
make "radius (:diameter / 2)
pu forward :radius pe
right 90
repeat 360 [forward (:diameter * pi) / 360
  right 1]
left 90
pu back :radius pd
ct
end
```

```
to erase.circle.r :radius
pu forward :radius pe
right 90
repeat 360 [forward (:radius * pi * 2) / 360
  right 1]
left 90
pu back :radius pd
ct
end
```

These procedures mirror the first procedures we introduced for erasing. Therefore no samples of their use will be provided here. You might notice what was omitted or added in these procedures as you compare them with the procedures they are to erase.

### "Earth and Moon" game

The following activity may be used to help students firm up their knowledge of circumferences and radii. Two turtle shapes are used: the predesigned shape 12 for the earth and a newly created smaller circular shape 1 for the moon. A knowledge of Cartesian coordinates is assumed.



```

to earth
clearpage
tell [0 1]
pu
setpos [0 0]
setsh 12
setc 11
st
show [Enter the coordinates of the point
      where you wish to locate EARTH.]
setpos readlistcc
pd stamp
end

to clearpage
if not front? [flip]
rg
ct
cc
end

to moon
cc
tell 1
pu
setsh 1
setc 4
st
show [What distance do you wish your moon to
      travel from the surface of the earth?]
make "distance readnumber
show [What circumference do you wish the
      moon to travel?]
make "circumference readnumber
forward :distance right 90
pd
circle :circumference
end

to readnumber
output first readlist
end

to circle :circumference
repeat 360 [forward (:circumference / 360)
            right 1]
end

```

First give the command **earth** and answer the question provided. Then give the command **moon** and answer both questions. There is a very important relationship to be discovered between the two numbers. You will soon find that the circumference must be a little more than six times the radius. Do you see why?

Here is a sample run-through. You would enter the underlined commands. The program displays the bracketed questions.

```

earth
[Enter the coordinates of the point where
you wish to locate EARTH.]
20 40 (a green earth appears there)
moon
[What distance do you wish your moon to
travel from the surface of the earth?]
(In other words: at what radius from the center
of the earth do you wish your orbiting moon to
travel?)

```

```

20 (a small blue moon appears)
[What circumference do you wish the moon to
travel?]
140

```



The blue moon safely orbits the green earth. Note that my entry is seven times greater than my radius of 20. I played it safe. A number of children did seem to enjoy getting the moon to "crash" into the earth. This was an easy enough task. Make certain that your radius is somewhat small and then diminish the size of the circumference.

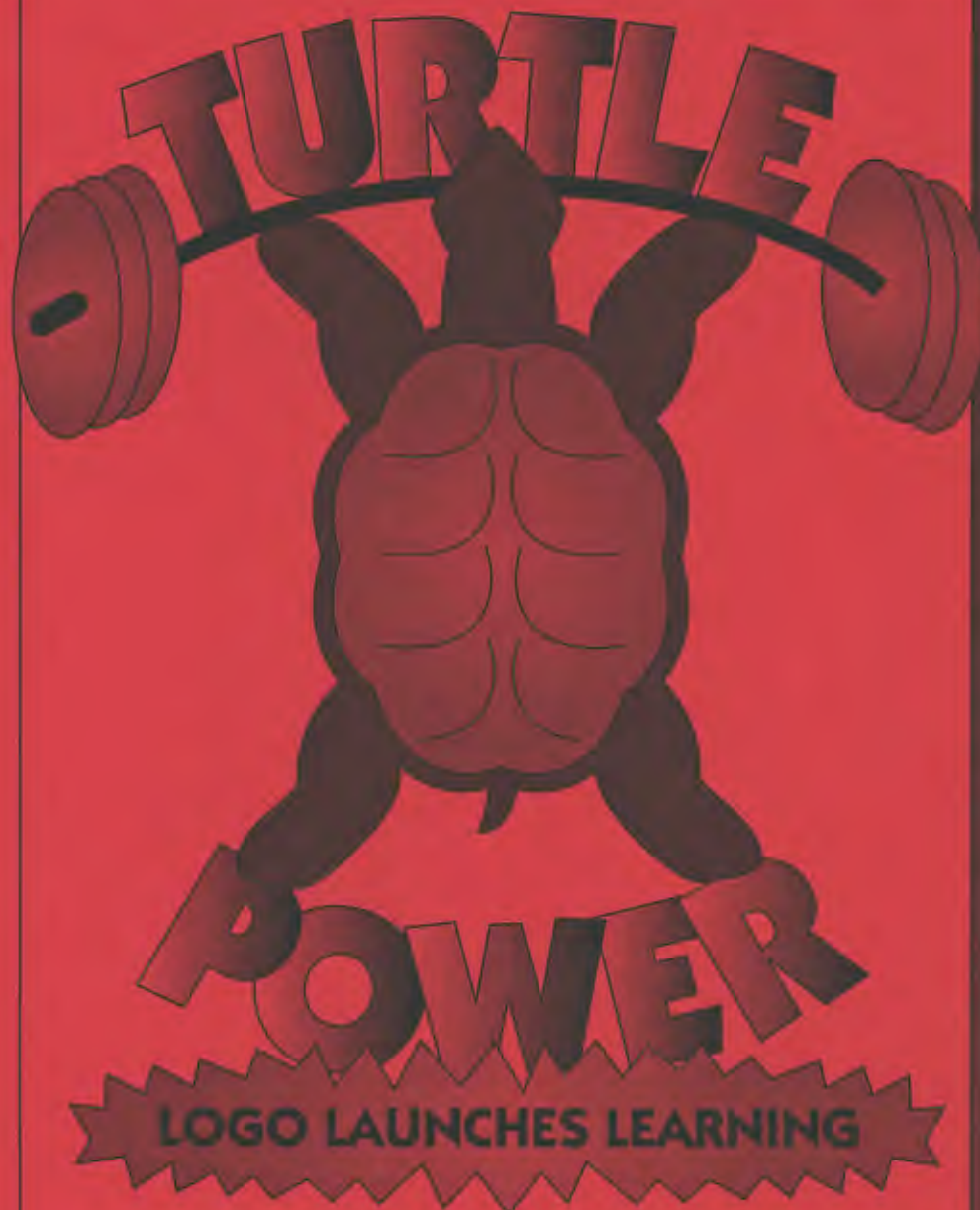
One young fellow expanded upon this and constructed a sunburst that appeared with printed statements announcing detonation when inputs were smaller.

### Some concluding observations

In working through this material, students were firming up what they had discovered with the hands-on material. The firming up was the result of enjoying what they were doing. Trying to impose the circumference of a circle exactly upon another with the use of different inputs (circumference, diameter, or radius) was a task many students found fascinating.

Logo had provided the second group of fourth graders with opportunities for learning that had been denied the first group. In a way, this points out some of the problems inherent in American education. When a mode of learning has been proven beneficial, why shouldn't it be shared? But what if the opportunities are ignored? Alas, proselytizing is not always the answer. ▲

**Robert Macdonald**, Hawthorne Meadows, 10225 Nancy's Blvd., Grosse Ile, Michigan 48138



Many educators around the globe have been using turtle power in their classrooms with very positive results. As a beginning programming language, Logo is your logical choice.

ISTE's Logo in the Classroom series gives teachers fun, easy, and innovative methods for using Logo programming to improve the thinking and problem-solving skills of precollege students.

Logo concepts and tools can be used to teach a variety of subjects. The series offers interactive ways to use Logo for teaching mathematics, language arts, art, modeling and simulations, and computer science.



**International Society for Technology in Education**  
*Customer Service Office*  
 480 Charelton Street, Eugene, OR 97401-2626 USA  
 Phone: 800/356-5191 (US and Canada)  
 Phone: 541/302-5777 (International)  
 Fax: 541/302-3778 Fax-on-Demand: 800/891-0212  
 America Online: ISTE Computer: 70034,2117  
 Internet: [iste@oregon.uoregon.edu](mailto:iste@oregon.uoregon.edu)  
 World Wide Web: <http://isteonline.uoregon.edu>

The ISTE Logo in the Classroom Books include the following:

**101 Ideas for Logo**

**The Children's Machine—  
 Rethinking School in the  
 Age of the Computer**

**Exploring Recursion With LogoWriter**

**Graphing With LogoWriter**

**Introduction to MicroWorlds—  
 A Logo-Based Hypermedia  
 Environment**

**Introduction to Programming in Logo  
 Using LogoWriter, Third Edition**

**Introduction to Programming in Logo  
 Using Logo PLUS**

**Introduction to Programming  
 Using Terrapin Logo for the  
 Macintosh**

**LogoWriter for Educators—  
 A Problem Solving Approach**

**Logo PLUS for Educators—  
 A Problem Solving Approach**

**Logo Learning—  
 Strategies for Assessing Content and  
 Process**

**Math Activities Using LogoWriter  
 Series:**

**High School Math  
 Investigations  
 More Investigations  
 Numbers and Operations  
 Patterns and Designs  
 More Patterns and Designs  
 Probability and Statistics**

**Turtle Power—Beginning Graphics  
 With LogoWriter, Revised Edition**

**More Turtle Power—Text, Graphics,  
 and Sound With LogoWriter**

**Turtle Toss Design—  
 Working With Patterns in LogoWriter**

**Weekly Words for Traveling Turtles—  
 Logo Applications in Language Arts**

LOGO, LOGOWRITER, AND MICROWORLDS ARE REGISTERED TRADEMARKS OF LOGIC COMPUTERSYSTEMS, INC. HONOLULU OFFICE: (808) 524-8111. WWW.LOGO.PLUS ARE REGISTERED TRADEMARKS OF TERRAPIN SOFTWARE, INC., CHAMBERS, MA, USA.



## ISTE BRINGS THE WORLD OF TECHNOLOGY CLOSER TO YOU.

By drawing from the resources of committed professionals worldwide, ISTE provides support that helps educators like yourself prepare for the future of education.

ISTE members benefit from the wide variety of publications, specialized courseware, and professional organizations available to them.

They also enjoy exciting conferences, global peer networking, and mind-expanding in graduate level distance education courses.

So if you're interested in the education of tomorrow, call us today.



**International Society for Technology in Education**  
*Customer Service Office*  
480 Charnelton Street, Eugene, OR 97401-2626 USA  
Phone: 800/336-5191 (US and Canada)  
Phone: 541/302-3777 (International)  
Fax: 541/302-3778  
Internet: [iste@oregon.uoregon.edu](mailto:iste@oregon.uoregon.edu)

WE'LL PUT YOU IN TOUCH WITH THE WORLD.