

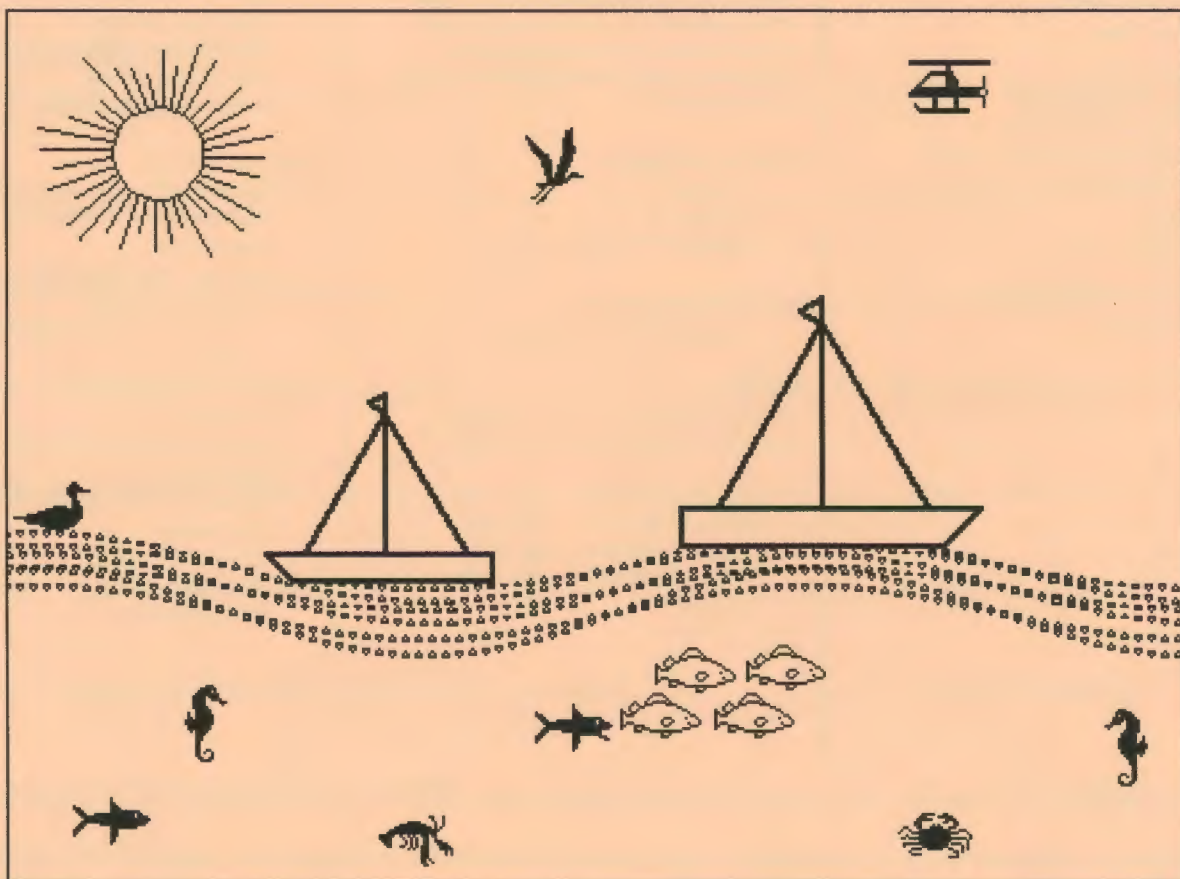


LOGO EXCHANGE

Volume 15 Number 4

Summer 1997

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



In this issue:

Sea Urchins

Estimating with Robotics

Drawing Objects for Flexibility

Language, Grammar, and Logo

The Game of Nim

The Plotting Turtle

Summer Logo Institutes

A Tale of Three Teachers

Editorial Publisher

International Society for Technology in Education

Editor-in-Chief

Dorothy M. Fitch

Copy Editor

S. M. Summerlight

Founding Editor

Tom Lough

International Editor

Dennis Harper

SIGLogo Officers

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

Production

Kerry Lutz

Director of Advertising Services

Judy Stickney

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:

Anita Best
1787 Agate St.
Eugene, OR 97403-1923

Send electronic mail to:

Anita_Best@ccmail.uoregon.edu

Deadlines

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

Volume 16, Number 2	June 1, 1997
Volume 16, Number 3	Oct. 1, 1997
Volume 16, Number 4	Jan. 1, 1998
Volume 17, Number 1	Mar. 1, 1998

ISTE Board of Directors 1996-97

David Brittain, *President*
MGT of America (FL)
David Moursund, *Executive Officer*

Executive Board Members

Lynne Schrum, *President-Elect*
University of Georgia-Athens (GA)
Terrie Gray, *Secretary*
Chico Unified School District (CA)
Terry Killion, *Treasurer*
Jefferson County R1 School District (CO)
Cheryl Lemke, *At Large*
Illinois State Board of Education (IL)
Gwen Solomon, *At Large*
US Department of Education (DC)

Board Members

Chip Kimball
Lake Washington School District (WA)
Paul O'Driscoll
Salem-Keizer Public Schools (OR)
Kathryn Edwards Price
Apple Computer, Inc., Santa Monica (CA)
Carla Schutte
Moton Elementary School (FL)
Neal Strudler
University of Nevada-Las Vegas (NV)
Harriet Taylor
Louisiana State University (LA)
Peter Wholihan
Saints Peter and Paul Catholic School (VI)
Michael Turzanski
Cisco Systems (MA)

Logo Exchange is published quarterly by the International Society for Technology in Education (ISTE), 1787 Agate Street, Eugene, OR 97403-1923, USA; 800/336-5191. This publication was produced using *Aldus PageMaker*®.

ISTE Members may join SIG/Logo for \$24.00. Dues include a subscription to *Logo Exchange*. Non ISTE Member subscription rate is \$34.00. 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.

© 1997 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; 508/750-8400; Fax 508/750-4470. ISTE members may apply directly to the ISTE office for free reprint permission.

POSTMASTER: Send address changes to *Logo Exchange*, ISTE, 480 Charnelton St., Eugene, OR 97401-2626 USA. Periodicals 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 15 Number 4

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

Summer 1997

Contents

From the Editor: Parting Thoughts	<i>Dorothy M. Fitch</i>	2
Quarterly Quantum		
Logo 101	<i>Tom Lough</i>	3
Logo Workshops: Summer 1997	<i>compiled by the editor</i>	5
Sea Urchins and Random Memories	<i>John Gough</i>	6
Robotics Fun: What's Your Best Guess?	<i>LEGO DACTA CONNEXIONS</i>	8
A Challenging Game of Nim	<i>Jim Muller</i>	13
Drawing Objects for Flexibility	<i>William J. Spezeski</i>	16
Musings: Language, Grammar, and Logo	<i>Robert Macdonald</i>	20
Seeing Algebra: My Turtle Plots Graphs	<i>John Gough</i>	27
Logo: Search and Research		
A Tale of Three Teachers, Part II	<i>Julie Sarama and Douglas H. Clements</i>	33

The cover is made from pieces of several articles in this issue. The sailboats are from Bill Spezeski's article on creating flexible shapes. The sun is drawn using the sea urchin code in John Gough's first article. The waves are made from graphing code in John Gough's second article, on algebra. The rest are shapes from Logo PLUS for the Macintosh, which the editor couldn't resist including! Have a great summer!



From the Editor: Parting Thoughts

by Dorothy M. Fitch

I write this editorial with sadness. As Tom Lough celebrates his 101st column, I write my last one. For personal reasons, I must step down as editor-in-chief of *Logo Exchange*.

I have truly enjoyed working on each issue of *LX*. What I have enjoyed most is my contact with all the authors and learning how they use Logo. I have met new friends, and gotten to know others even better. I now have faces to attach to many names.

I want to thank all the *LX* authors, ISTE staff, and SIG-Logo officers. It's been a pleasure to work with all of you. As this issue goes to press, a search continues for a new editor. Hopefully not a beat will be missed for the Fall issue. Continue to send in your submissions!

Logo saves the day!

A few weeks ago I had the delight of using Logo for a very practical purpose. In fact, Logo was the only computer tool I could think of that could do the job, or at least make it easy.

The task was to make a scale drawing of a large hayfield that is part of a nature center. The center had a trail map, but it was not as carefully drawn as we needed. We needed it to show sites for mist nets for a bird-banding station. The nets were to be at least 50 meters apart; most would be situated in the woods, just inside the edge of the hayfield. Our map needed to show the area and indicate net sites.

So, we gathered friends one day and set out with a 50-meter length of string, compass, 100-foot measuring tape, and notebook. We used the string to confirm that the net sites were far enough apart. While others selected net sites, two of us set out to survey the field.

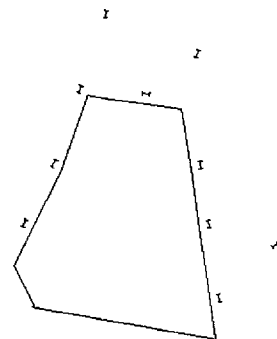
My friend Hank took compass readings along each straight edge of the field and paced out each distance. He double-checked both measurements while I took notes. When we were done, I had a rough sketch, labeled with degree head-

ings and distances for each leg. Hank measured his stride against the 100-foot tape several times to find the length of his average pace.

Someone asked if I needed to use his computer program to draw the map. I said I couldn't imagine a better tool for the job than Logo, which he didn't know about. Hank recalled that his kids had used it in school.

That evening, within a few minutes I had the outline of a perfectly scaled map. The measuring had been so carefully done that the last line came within a few dots of the starting position. All it took was a series of **setheading** and **forward** commands! Amazingly, I could use the number of paces as the input to **forward** with no scaling required.

I drew symbols for nets in the correct locations, and used Logo PLUS's **stampcircle** command to draw 50 meter circles around each one, having written a function to convert the pace measurement to meters. Here is a tiny version of the map (without the circles):



I challenge you to find another application to do this job—to handle distances, headings, and conversions—and let you make changes quickly and easily. Isn't Logo great?

Happy Logo adventures. 'Til we meet again!

Dorothy M. Fitch, LX Editor

3 Derby Road, Derry, NH 03038

E-mail: 71760.366@compuserve.com

Telephone: 603-425-2010 Fax: 603-425-6487

Quarterly Quantum: Logo 101

by Tom Lough

I just could not believe it! And yet figures do not lie (most of the time). How could it have happened so quickly? Amazing!

According to my calculations, I am now writing my 101st regular* *LX* editorial! Please permit me to ruminate on this realization for just a moment or two.

As many of you know, this journal began in September 1982 as an eight-page newsletter output from a dot-matrix printer. From the very first issue, everyone involved with *The National LOGO Exchange* (as it was known then) was enthusiastically dedicated to learning and communicating the full potential of this fascinating computer language.

In editorial number one, I tried to describe this dedication by explaining our motto, FORWARD 100.

Here at *The National LOGO Exchange*, our motto is FORWARD 100! This reflects our enthusiasm toward LOGO and its role in education. We are very excited about the potential represented by LOGO. We want to press on in a FORWARD direction with our efforts to bring LOGO to a position from which it can influence our country's children. We want to give 100% support to you, the LOGO teachers and parents, as you work with LOGO in the classrooms and homes. Please feel free to adopt this motto as your own, for LOGO, and for life!

—*The National LOGO Exchange*, Vol. 1, No. 1, September 1982, page 2.

Since then, the idea of FD 100 has spread worldwide. With the launching of *The International LOGO Exchange* newsletter in 1986 and its later combining with the *NLX* into the present *Logo Exchange* in 1987, the publication you are holding established itself as a global Logo clear-

inghouse. Logo ideas from the *LX* have influenced hundreds of thousands of children in countries all around the earth.

I did not set off on this journey with the goal of writing 100 editorials. Along the way, I have had the opportunity to work with dedicated educators such as Glen Bull and Steve Tipps, with enthusiastic colleagues such as Sharon (Burrowes) Yoder and Michael Tempel, and with gifted visionaries such as Seymour Papert. Inspirational co-workers such as these give wings to my feet (and to my typing fingers).

In actuality, this 101st editorial is a testimony to the work of you, our readers, as well as to the power of Logo to engage, to challenge, and to stimulate. You have brought this computer language to your children, you have nurtured them as they encountered amazing ideas and discoveries, and you have followed them as they built their own knowledge and then applied it in wonderful and unanticipated ways. On behalf of the children of the world, thank you!

In this 101st editorial, I would also like to thank Dorothy Fitch for the outstanding job she had done as *LX* editor for the past several years. Drawing upon her longtime involvement with Logo, she brought to the *LX* a sense of balance, providing both challenging articles for experienced readers as well as nurturing articles for those just getting started. It is always a pleasure to work with such dedicated people as Dorothy. Thanks, D!

Just like the turtle drawing the classic square, I feel as if I am turning a corner after going FD 100. What will the next 100 steps be like? One of the thoughts I had was to use the Roman numeral C and write a procedure.

```
to C
output 100
end
```



In this way, we can express our motto as FD C. We can think back over the past 100 editorials and reflect. Then we can look to the future and wonder what challenges, what triumphs, what realizations and discoveries it will bring. This suggests, then, that our motto might become

FD C more! ▲

Tom Lough

Founding Editor
20 Whitcomb Drive
Simsbury, CT 06070
70020.223@compuserve.com

P.S. Many of the articles in the early *National LOGO Exchange* newsletters contain ideas and suggestions that are still relevant and stimulating today. These newsletters are available in microfiche form through ERIC. See your media specialist for more information.

Volumes 1-3 (September 1982 to May 1985) ED
264 842

Volume 4 (September 1985 to May 1986) ED
279 305

* Astute *LX* diehards will quickly recognize that there have been more than 101 issues published. Just to set the record straight, guest editorials (Steve Tipps, December 1983; and Elaine Blitman, November 1985), the tribute to the Challenger crew (March 1986), and the spoof April Fool issue of 1991 are not included in this tally.

ISTE PUBLICATIONS

HYPERMEDIA HELP FROM ISTE!

Two, Two, Two Worlds in One



Introduction to MicroWorlds 2.0— A Logo-Based Hypermedia Environment

MicroWorlds is a powerful hypermedia application that offers many of the appealing features seen in Kid Pix, HyperCard, and LogoWriter in a single program. This Version 2.0 edition is newly revised and includes all new screen shots. Learn to use MicroWorlds objects to control turtles and add text. Learn to integrate sound, music, audio CDs, Laserdiscs, and QuickTime movies into your computer-generated projects. Each chapter discusses problem solving in the MicroWorlds environment and includes tips on techniques, activities, and self-assessment. Appendices contain a list of all the Logo primitives available in MicroWorlds.

Sharon Yoder & David Moursund

*Introduction to MicroWorlds 2.0—A Logo-Based
Hypermedia Environment*

211 pages, ISBN 1-56484-106-5

©ISTE, 1996

To order this book or receive the most recent *Resources & Services for Technology-Using Educators* guide, contact ISTE.



International Society for Technology in Education
Customer Service Office
Phone: 800/336-5191 (US and Canada)
Phone: 541/302-3777 (International)
Fax: 541/302-3778
World Wide Web: <http://www.iste.org>



Logo Workshops: Summer 1997

Stonington Retreat — Maine

June 22–28 on Deer Isle

Led by founder Laura Allen, the Stonington Retreat is a week-long opportunity for educators to learn, play, and discuss ideas about technology, Logo, and learning. This experience combines hands-on time with high level discussions about the future of learning and technology. Facilitators include Fred Martin, Robbie Berg, and Wanda Gleason.

Registration: \$1,400 per person (tuition, all breakfasts, three dinners, board, scenic boat ride)

For more information, contact Laura Allen at:
303 West 66th St. 19 FE
New York, NY 10023
Telephone: 212-873-3553
E-mail: laurallen@aol.com

Summer at Spence — New York

June 23–27 in New York City

The Spence School, an independent school for girls located in the heart of New York City, just steps from Manhattan's Museum Mile, is a leader in educational technology. Share ideas with the teachers here, where educational technology is being thoroughly infused into the curriculum from grades K–12.

Registration: \$590 per person

For more information, contact:
Michael Tempel at the Logo Foundation
250 West 85th Street, Suite 4D
New York, NY 10024
Phone: 212-579-8028; Fax: 212-579-8013
E-mail: michael@media.mit.edu
<http://el.www.media.mit.edu/groups/logo-foundation/>

Logosium — Seattle

July 27, 1997 in Seattle, Washington

Site: John Hay School, 201 Garfield Street
Time: 9:00 a.m. – 4:00 p.m.
Fees: Adults (over 17) \$55.00
Children (under 18) \$6.00

This miniconference (in conjunction with NECC '97) is for Logo practitioners and people interested in finding out about Logo. We will share our thoughts and expertise on building constructivist environments and on devising projects that are educationally rich and interesting. There will also be hands-on, how-to-

do-it, and what's-new sessions. Participants should bring their own best projects and hardest questions. (Children under 12 must be accompanied by an adult.)

Logosium '97 is sponsored by ISTE SIG-Logo and The Logo Foundation. To register, see the NECC '97 materials.

For more information, contact:

Marian Rosen mbrosen@icon-stl.net
or Michael Tempel michael@media.mit.edu

Colorado Logo Institute

July 27–31 in Grand Junction, Colorado

Mesa State College hosts its first Logo Summer Institute. Join us for four days of learning and relaxation in beautiful western Colorado. Meals and recreational activities, such as rafting on the Colorado River, are included in your registration.

Registration: \$470 per person

For information, contact Michael Tempel at the Logo Foundation (see address information listed for **Summer at Spence**).

For information about housing, contact:

Ben Keefer, Continuing Education Center
Mesa State College
1175 Texas Avenue
Grand Junction, CO 81501
E-mail: keefer@wpogate.mesa.colorado.edu
Phone: 970-281-1476

Logo St. Paul — Minnesota

August 18–22 in St. Paul

Over the past 16 years, the St. Paul Logo Project has provided a comprehensive professional development program for hundreds of elementary and secondary school teachers. A limited number of places has been set aside for people from outside the St. Paul Public Schools. Here you will interact with teachers who have experience implementing Logo in a wide variety of urban school settings. Optional graduate credit is available from Hamline University.

Registration: \$590 per person; \$126 for university credit

For more information, contact Michael Tempel at the Logo Foundation (see address information listed for **Summer at Spence**). ▲

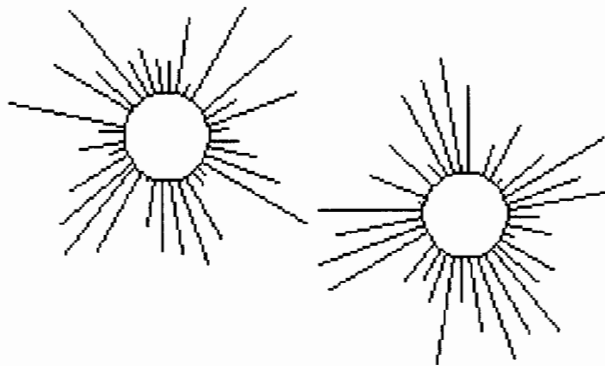


Sea Urchins and Random Memories

by John Gough

I am continually trying to find new things to do with LogoWriter. Recent seaside experience suggested drawing sea urchins. It's easy to draw a circle. And it's easy to make spikes stick out from the circle as the turtle moves—that is, it's easy if the spikes are all the same length.

But what if the sea urchin has spikes of random length?



We can turn the turtle so it faces perpendicularly away from the circumference, and then move the turtle forward some random number. But how will the turtle know how far to move back so that it can turn back and resume drawing the circle? We need a way to remember the random number, or some alternative.

At first I thought of creating a global variable using a **make** command.

```
to sea.urchin
repeat 36 [fd 3 spike rt 10]
end

to spike
make "s random 50
lt 90 fd :s bk :s rt 90
end
```

But then I wondered if it could be done without using a global variable such as "s. A sudden inspiration suggested this alternative.

```
to sea.urchin.1
repeat 36 [fd 3 spike.1 pos rt 10]
end

to spike.1 :posit
lt 90 fd random 50
setpos :posit rt 90
end
```

This stores the turtle's current position by using the command **pos** to provide a value for the local variable **:posit** in the modified procedure **spike.1 :posit**.

Here is another way.

```
to sea.urchin.2
repeat 36
[fd 3 spike.2 random 50 rt 10]
end

to spike.2 :size
lt 90 fd :size
bk :size rt 90
end
```

In this case, the command **random** is used once to provide the initial value for the length of the urchin spike when the modified procedure **spike.2** is called within the larger procedure **sea.urchin.2**, and this is remembered through **spike.2** as the value of the local variable **:size**.

Drawing random spikes around a circle suggested the following "tartan":

```
to tartan :unit
pu setpos [130 -80] pd
repeat 2
[repeat 160 / :unit [fd :unit spike.2
random 270] lt 90
repeat 270 / :unit [fd :unit spike.2
random 160] lt 90]
repeat 50 [spotty]
end
```

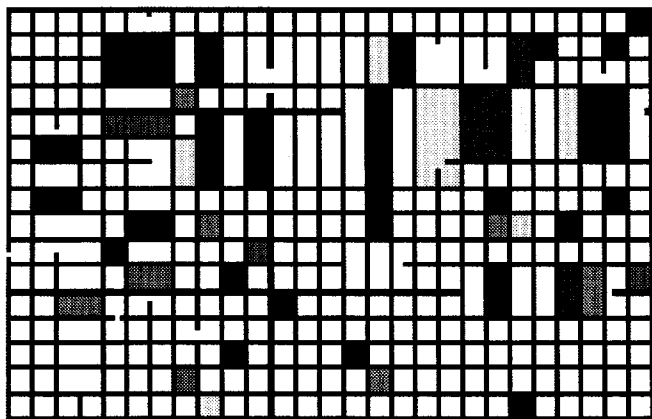


```

to spotty
pu setpos list (random 250) - 125 (random
140) - 70
ifelse colorunder = 1 [stop] [pd setc 2 +
random 6 fill]
end

```

Do you like this Scottish modern art?



For *John Gough's* biography, refer to his other article in this issue, "Seeing Algebra: My Turtle Plots Graphs."

NATIONAL • EDUCATIONAL • COMPUTING • CONFERENCE

POTLATCH

@NECC97.seattle.wa.edu



Don't miss your opportunity to experience one of the foremost K-16 educational technology conferences in the country!

Join your national and regional colleagues this summer at the eighteenth annual National Educational Computing Conference (NECC) for three days of top-notch sessions, networking opportunities, and internationally known Keynote and Spotlight speakers—plus two days of preconference workshops, June 28 and 29.

Seattle is a breathtaking city, and we hope that in addition to enjoying all that the conference has to offer professionally, you'll take time out for a ferry ride, take in the view from the Space Needle, and dine at some of the finest restaurants and cafés in the Northwest.

For the latest information on attending or exhibiting, check us out on the Web or call us toll-free today!

NECC '97 is hosted by Western Washington University (WU) and is sponsored by the National Educational Computing Association (NECA).



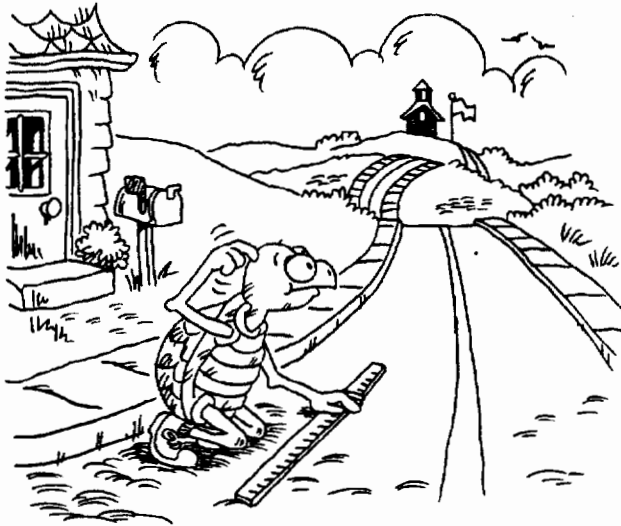
June 30–July 2, 1997 • Seattle, Washington • 1.800.280.6218 • <http://www.wce.wvu.edu/necc97>



Robotics Fun: What's Your Best Guess?

reprinted with permission from LEGO DACTA CONNEXION™, Vol. 3 No. 1, Jan. 1990.

When you estimate you make calculated guesses. For example: If someone asks you how far it is to school, you might say, "about two miles." You would not be expected to give an exact answer—"two miles, three hundred fifty-seven yards and two inches!" You would be expected to estimate.



In this project you are going to use a robotic turtle to estimate distances and degrees of rotation.

This project was originally written for use with the LEGO® TC logo program. Feel free to adapt it for use with your robotics program and device.

Materials

- LEGO® TC logo turtle (or other robotic device that you can move and turn by controlling motors)
- Ruler
- Masking tape (or something to mark distance on the floor or table)
- Inventor's Notebook or sheet of paper

Setup

1. Build a robotic turtle, using LEGO activity card 9700-8, for example.

2. You must create a new page with the **turtle** procedures on it. First get the **turtle** page.
3. Then create a page called estimate. Type **namepage "estimate** in the Command Center.
4. You will need special procedures to operate the turtle. Add the procedures below to your **estimate** page.

```
to startup
cc
reset
end
```

```
to reset
alloff
recycle
talkto [A B]
setpower 5
end
```

```
to setup.l (prepares motors to turn left)
recycle
talkto "A
seteven
talkto "B
setodd
talkto [A B]
end
```

```
to setup.r (prepares motors to turn right)
recycle
talkto "A
setodd
talkto "B
seteven
talkto [A B]
end
```

```
to rl :degrees (rotates motors to the left)
setup.l
onfor :degrees/4.7
reset
end
```

```
to rr :degrees (rotates motors to the right)
setup.r
onfor :degrees/4.7
reset
end
```

For these procedures to work correctly, be sure to connect the RED plug to port A and the BLUE plug to port B. To test your connections, type **talkto [A B] onfor 20** in the Command Center. If the turtle moves forward, everything is connected properly. If the turtle turns to the right instead of going forward, turn the plug in port A around. If it goes left, turn the plug in port B around. If it goes backward, turn the plugs in both ports A and B around.

Estimating distance in inches

You are going to estimate the number of inches the turtle moves. Follow the steps below:

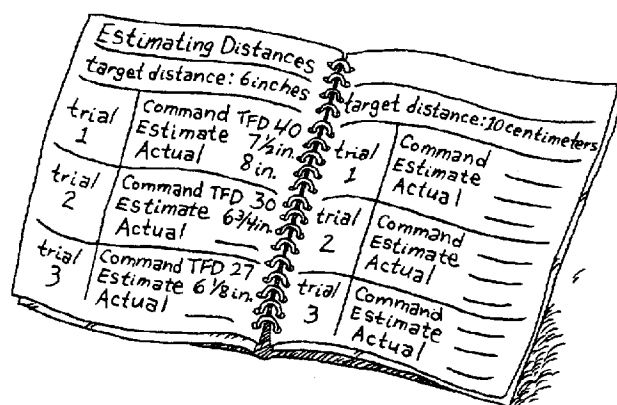
- Step 1: Mark your turtle's starting position with a piece of masking tape.
- Step 2: Use the turtle forward command (e.g., **tfd 30**) to try to move the turtle six inches.
- Step 3: Mark the turtle's stopping point.
- Step 4: Estimate how far the turtle moved.
- Step 5: Now measure how far the turtle moved.
- Step 6: Record the result in your Inventor's Notebook.
- Step 7: Return the turtle to its starting position. Repeat the exercise using one **tfd** command.

Do three trials for six inches and record your results. Then try three trials for each of the following distances: 13 inches; 20 inches; your choice.

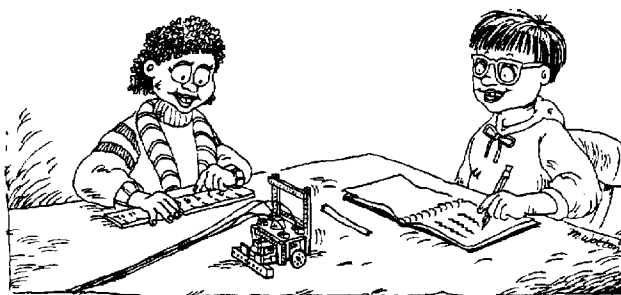
Remember to record your results!

Estimating distance in centimeters

Centimeters are a unit of measure in the metric system. Follow the steps you used for estimating distance in inches to estimate the distance your turtle moves in centimeters. Do trials for the following distances: 17 centimeters; 35 centimeters; your choice.



Traveling turtle



1. Mark the turtle's starting point.
2. Have one member of your team use **gofd** or **gobk** to move the turtle forward or backward a desired distance. (The command **ot** turns off the turtle.)
3. Mark the turtle's stopping point.
4. Have everyone on the team make two estimates of the distance the turtle moved—one estimate in inches and one estimate in centimeters. Team members record each of their estimates in their own Inventor's Notebook.
5. Measure the actual distance in inches and in centimeters. Which team member came closest to estimating the actual distance the turtle traveled?
6. Record the actual distance the turtle moved.

Try at least three trials. Estimate both in inches and in centimeters. Remember to record all the data in your Inventor's Notebook.

Traveling Turtle			
trial 1		trial 2	
estimate	12"	30cm	estimate
actual	10"	25.4cm	actual
difference	+2"	+4.6cm	difference
			estimate
			actual
			difference
			7 1/2"
			8 1/4"
			-3/4"
			17cm
			21cm
			-4cm



All the way there and halfway back

Here's a harder challenge!

1. Follow steps 1–3 in "Traveling Turtle."
2. Now measure the distance from the starting point to the stopping point. Let's call this the "all the way distance."
3. Record the "all the way" distance.
4. Estimate the "half way" distance. Mark your estimated distance.
5. Now measure the distance from the "all the way" stopping point to your half-way mark. Record your result.
6. Calculate exactly half of the "all the way" distance. (Divide by 2.) Record your result.

How close was your estimate to the actual "half way back" distance?

Do a few more trials. Let everyone on the team move the turtle.

TRIAL 1	
Half way estimate	8 inches
"All the way" actual	18 1/2 inches
Half way actual	9 1/4 inches

1. Place a small object (LEGO brick, piece of chalk, etc.) near the turtle, but off to one side.
2. Use the **rr** (rotate right) procedure to turn the turtle so that it points directly at the object.
3. Record the command you typed.
4. If you don't reach the object or if you pass it, use **rr** or **rl** (rotate left) to try again.
5. Remember to record each command you type.

You can try the same activity using the **rl** command first.

Putting it together

1. Position your turtle in one location and a target object (LEGO brick) in another location.
2. Use the **rr**, **rl**, **tfd**, and **tbk** commands to move your turtle as close to the target as possible without hitting it. Try to use as few commands as possible.
3. Record each command as you go.
4. If you hit the target, record a "hit" and count the hit as one extra move.
5. Move the target object to another position and try again!

What other rotation/distance activities can you design to give yourself more practice?

Turtle turns

In this activity, you are going to estimate degrees of rotation.

Teaching Notes

Key ideas

1. Estimating and measuring in inches.
2. Estimating and measure in centimeters.



3. Estimating and measuring degrees.
4. Learning about acute and obtuse angles.

Materials

In the procedure **reset**, you could change the input to **setpower** to any number between 1 and 7. But remember to recalibrate angles in **rr** and **rl**.

Excess friction on gears and variations in the speed of the turtle's motors affect measurements. The surface on which the turtle travels affects its speed.

Estimating distance in inches

It is very important that students record the results of each trial as it takes place. Students should take turns controlling the turtle and marking the starting and stopping points.

Estimating distance in centimeters

This activity is a good opportunity to address students' working knowledge of the metric system, specifically regarding the relationship between a centimeter and an inch. For example, once they have measured the distance in inches, they could calculate the distance in centimeters, rather than measuring separately (1 inch = 2.54 cm).

Which was easier: estimating in inches or in centimeters? Why?

Were longer distances harder to estimate than shorter ones?

Can you think of instances when you had to estimate distance?

Traveling turtle

Students will find that it takes practice to control the turtle with the **gofd**, **gobk**, and **ot** (off turtle) commands. It may be helpful to have the practice before actually doing the activities. **Gofd** and **gobk** simply turn on the motors. **Tfd** and **tbk** turn on the motors for a specified time.

The number of estimates recorded for each trial in "Traveling Turtle" will depend on the number of students per team. You may want to have each student make estimates in both inches and centimeters. Or, if you have four students per team, you may want to have two students

give estimates in inches and two give estimates in centimeters.

All the way there and halfway back

"Halfway back" provides a good opportunity to review fractional quantities and basic arithmetic skills.

As an extended or advanced activity, students could move the turtle a different fractional quantity (e.g., one third the original distance).

Students can use the **show** command to calculate the exact half distance. For example: If the turtle travels 11.5 inches, type **show 11.5/2** in the Command Center to calculate half the distance.

Which was easier: estimating the original distance (in "Traveling Turtle") or half the distance back?

For what occupations are estimation skills important?

Can you think of any situation in which estimation isn't enough and accuracy is necessary?

Turtle turns

In this activity, students are required to use a specific command first so that they will increase their ability to rotate more than 180 degrees. They will observe that turning a large angle in one direction can be equivalent to turning a small angle in the opposite direction.

You may find that the **rr** and **rl** commands are not very accurate. If that is the case, you will need to change the calculations in one or both procedures. (This is necessary because not all models are built exactly the same, and may rotate at different speeds.)

If the turtle doesn't rotate far enough, change the number 4.7 in the procedure to 4.9 or 5.1. If the turtle rotates too far, change the number 4.7 to 4.3 or 4. Continue to adjust these numbers until the turtle approximates closely the actual rotation in degrees.

You may want to make a practice page with different angles drawn on it (e.g. 45°, 90°, 135°, 180°, and so on). Then you could test your turtle's accuracy by putting it in the center of the prac-



tice page, typing **rr 45** to see if the turtle turns 45 degrees.

Rr and **rl** may not work with numbers less than 5 or 6, depending on your calibration. The numbers may simply be so small that the motors turn on and off too quickly to move the turtle.

Once the results have been recorded, they provide valuable discussion material. Students should see that their accuracy increased with successive trials; their first trial was probably always less accurate than their third trial.

Discussion

Which do you estimate more accurately, rotation or distance? Why?

Do you think the people who make the LEGO building materials estimate or use exact calculations to make LEGO bricks?

What other kinds of estimations do we use besides rotation and distance? ▲

STEP PUBLICATIONS
A FUN WAY TO TEACH MATH!

Simply Amusing!

Logo Musings—Ten Mathematical Encounters Using LogoWriter

These encounters use manipulatives, calculators, worksheets, and the *LogoWriter* and *MicroWorlds* programs to help students practice and have fun with numerical patterning. Topics covered include the numerical value of words, probability, magic squares, topology, and square, rectangular, and triangular numbers. Each chapter includes files, references, and worksheets. *LogoWriter 2.0* and *MicroWorlds Project Builder* are required, but not included. Grades 4–8 (ages 9–14).

Robert Macdonald

Logo Musings—Ten Mathematical Encounters Using LogoWriter

180 pages, 1 disk, ISBN 1-56484-101-4, ©ISTE, 1996

To order this book or receive the most recent *Resources & Services for Technology-Using Educators* guide, contact ISTE.



International Society for Technology in Education
Customer Service Office
Phone: 800/336-5191 (US and Canada)
Phone: 541/302-3777 (International)
Fax: 541/302-3778
World Wide Web: <http://www.iste.org>



A Challenging Game of Nim

by Jim Muller

The game of Nim is an old favorite that you may remember from your youth. When you first began to play, you could never win. However, after you learned the secret, you could never lose.

The game is easy enough. Two players start out with a small pile of toothpicks or stones. The players decide how many pieces each can take on a turn and how many turns each will have. The whole idea is not to get stuck taking the last piece.

One nice thing about Nim is how easy it is to write in Logo. Here is the game in PC Logo.

```
to nim
  cleartext textscreen
  repeat 5 [pr "]
  pr [Welcome to the game of nim!]
  pr "
  pr [This is a game where you and the]
  pr [computer take turns picking stones]
  pr [from a pile. The challenge is not]
  pr [to get stuck with the last stone.]
  pr "
  pr [You decide. How many stones will]
  pr [each player be able pick on each turn?]
  make "pick first readlist
  make "picks :pick + 1
  pr [How many turns will we have to pick?]
  make "turns first readlist
  make "total :picks * :turns + 1
  game
  over
end

to game
  (pr [There are] :total [stones in a pile.])
  (pr [Take from 1 to] :pick [stones.])
  make "key get.number
  pr "
  (pr [I take] word :picks - :key ".)
  pr "
  make "total :total - :picks
  if :total > 1 [game]
end

to get.number
  make "key rc
  if :key > :pick [pr " pr [That's too many.
    Try again!] output get.number]
```

```
if :key < 1 [pr " pr [Pick at least one.]
  output get.number]
output :key
end

to over
  repeat 3 [pr "]
  pr [There is one stone left. I win again.]
end
```

Take a good look at this procedure. Do you see why the computer always wins?

The answer is right there in the simple formula that follows, "I take" in the **game** procedure. The computer always takes **:picks - :key**. Do the math yourself. You will soon see why there is no way the computer can lose.

PC Logo uses classic Logo, even in its Windows package. The procedure shown below, however, was written using the freeware package, MSW Logo.

This package is unique in that it adds a full range of Windows features to explore. A few of these are included in the version of the game listed below:

```
to nim
  ct
  pr [Welcome to the Game of NIM!]
  pr "
  pr [This is a game where you and the]
  pr [computer take turns taking stones]
  pr [from a pile. The challenge is not]
  pr [to get stuck with the last stone.]
  pr "
  pr [You decide. How many stones will each]
  pr [player be able to take on each turn?]
  selectbox [Pick a number.]
  make "pick :key
  make "picks :pick + 1
  pr [How many turns will each player have to]
  TAKE stones?]
  selectbox [Pick a number.]
  make "turns :key
  make "total :picks * :turns + 1
  game
  over
end
```



```

to game
(pr [There are ] :total [ stones in a
  pile.])
(pr [Take from 1 to ] :pick [ stones.])
selectbox [Pick a number.]
if or (:key > :pick)(:key < 1) [correct
  stop]
(pr [I take] word :picks - :key ".)
make "total :total - :picks
if :total > 1 [game]
end

to over
pr [There is one stone left. I win again.]
make "ans yesno [nim] [Want to play
  again?]
ifelse :ans = "true [nim] [ct pr [Bye for
  now!]]
end

to selectbox :question
pr :question
dialogcreate "root "Selector "SelectBox 250
  20 100 180 [selectbox.setup]
end

to selectbox.execute
if radiobuttonget "one [make "key 1 del
  stop]
if radiobuttonget "two [make "key 2 del
  stop]
if radiobuttonget "three [make "key 3 del
  stop]
if radiobuttonget "four [make "key 4 del
  stop]
if radiobuttonget "five [make "key 5 del
  stop]
if radiobuttonget "six [make "key 6 del
  stop]
if radiobuttonget "seven [make "key 7 del
  stop]
if radiobuttonget "eight [make "key 8 del
  stop]
if radiobuttonget "nine [make "key 9 del
  stop]
end

to selectbox.setup
groupboxcreate "Selector "nim 10 10 80 140
radiobuttoncreate "Selector "nim "one [One]
  20 15 60 15
radiobuttoncreate "Selector "nim "two [Two]
  20 30 60 15
radiobuttoncreate "Selector "nim "three
  [Three] 20 45 60 15
radiobuttoncreate "Selector "nim "four
  [Four] 20 60 60 15
radiobuttoncreate "Selector "nim "five
  [Five] 20 75 60 15
radiobuttoncreate "Selector "nim "six [Six]
  20 90 60 15

```

```

radiobuttoncreate "Selector "nim "seven
  [Seven] 20 105 60 15
radiobuttoncreate "Selector "nim "eight
  [Eight] 20 120 60 15
radiobuttoncreate "Selector "nim "nine
  [Nine] 20 135 60 15
radiobuttonset "one "true
radiobuttonset "two "false
radiobuttonset "three "false
radiobuttonset "four "false
radiobuttonset "five "false
radiobuttonset "six "false
radiobuttonset "seven "false
radiobuttonset "eight "false
radiobuttonset "nine "false
buttoncreate "Selector "game "OK 35 150 25
  20 [selectbox.execute]
end

```

```

to correct
messagebox [nim] [That number won't work.
  Try again!]
game
end

```

```

to del
dialogdelete "Selector
end

```

Challenges

Before exploring the opportunities of Windows programming, why not consider some of the challenges open to you regardless of the Logo dialect you are using?

Why not make it a graphical game? Create pictures of the pieces used in the game—things like stones, toothpicks, rabbits, or cars. This really doesn't change the game. It just makes it easier to look at.

How can you change the procedure to make it more of a challenge? In other words, can you change it so that the computer doesn't always win?

The easy way is simply to add a random element. The computer takes a random number of pieces on each turn. If you do your arithmetic correctly, you have a good chance of turning the tables on the computer. Of course, many games are fun because they have a degree of randomness that keeps things interesting. However, randomness does not do much to develop strategic thinking.



How can you make it tough on both the player and the computer?

One way to add a strategic element to the game is to distribute the pieces on multiple rows. Have the players select pieces from just a single row on any one turn. For example, a real challenge can be developed by using pieces placed on three rows. However, as you can guess, you will get to the point where you have one or two pieces on each row. Your original algorithm does not work in such situations, so you'll have to add new rules for the end of the game.

Writing game procedures is fun, regardless of the Logo dialect you use. Games are fun to play, but the novelty can soon wear off. This is where a good game design challenge can end up being more fun than the game itself.

Why windows programming

Windows programming supports two modes, Modal and Modeless. Modal programming is similar to non-windows programming. When running Logo, the procedure controls the action. For example, during processing, you (as the programmer) decide to ask the user for information using **readlist**, **readword**, or **readchar**. Everything stops until the user enters the requested information.

In Modeless programming, the tables are turned. The window (actually the user) is in control. This takes some getting used to; however, it is a very important concept to learn. Using the example above, rather than halt everything while waiting for the user to respond, Logo is merely idle.

While Logo is idle, you can change the colors being used, for example. You can even write an execute another procedure. The original procedure continues to wait until you trigger an appropriate event such as pushing a button. For example, if the game of Nim was done using Modeless programming, you go off and test out a turn before entering it in the real game.

Why program with windows? Try to imagine what a child's imagination could do in such a multi-event environment. ▲

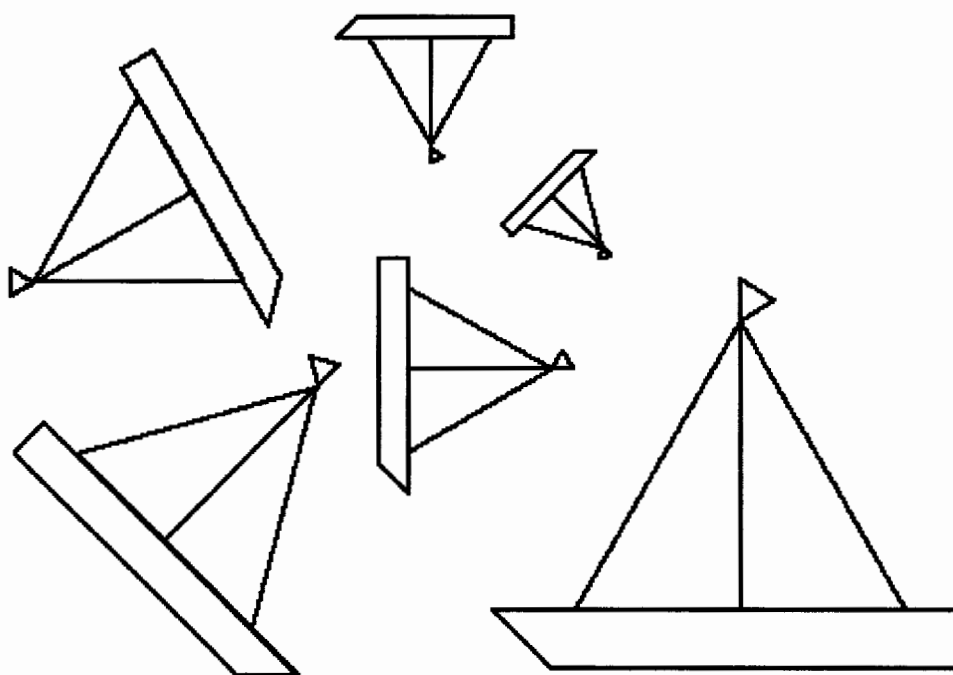
Jim Muller has had a lifelong interest in translating various technologies into understandable and persuasive programs. In 1981, Muller and his son organized the first Logo users group, the Young Peoples' Logo Association, which eventually grew into a worldwide 6,000-member organization. In 1985, the 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.

E-mail: 76703.3005@compuserve.com



Drawing Objects for Flexibility

by William J. Spezeski



Consider the sailboats shown above. They appear in different sizes and orientations. But most important, these sailboats were drawn using a single Logo drawing procedure. Writing procedures that allow this kind of flexibility is the subject of this article.

Making objects movable

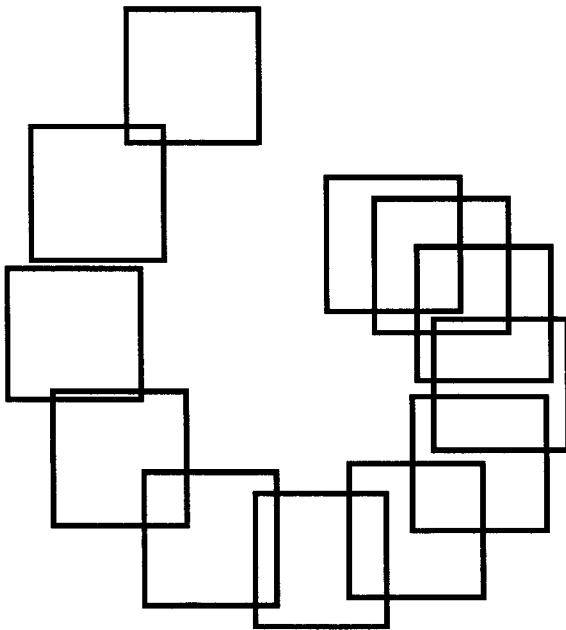
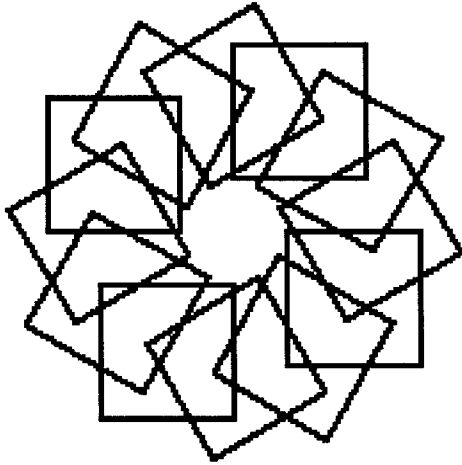
Creating objects that can be positioned anywhere on the screen is relatively easy. The secret is to restrict the commands in the drawing procedure to those that result in relative turtle movement (i.e., **fd**, **bk**, **rt**, **lt**). Avoid commands that result in absolute turtle movement such as **setxy**, **seth**, and **home**. The effects of these two categories of commands can easily be seen in the two procedures below:

```
to square
; relative turtle movement
repeat 4 [fd 50 rt 90]
end
```

```
to square1
; absolute turtle movement
pu home pd
setxy [ 0 50]
setxy [50 50]
setxy [50 0]
home
end
```

Both procedures draw squares of side length 50. Procedure **square** uses only relative turtle commands, while procedure **square1** uses absolute commands to accomplish the task. But only procedure **square** has the flexibility to draw the square at various locations and alignments on the screen.

Once an object is created by a procedure using relative turtle movement, it can be repositioned anywhere on the screen by simply changing the initial drawing coordinates of the turtle. Likewise, the object can be rotated to any heading by selecting a new drawing heading for the turtle. The following two pictures show the same square procedure drawn at various positions and headings.



Another sound practice to follow when drawing objects is always to return the turtle to its original starting position in order to complete the drawing. This is not a requirement, merely a strong suggestion. But it certainly makes life easier to know where the turtle is when the task of drawing the object is complete. Returning the turtle to its starting position becomes particularly important when you work with animation. In the interest of brevity, the procedure that draws the sailboat in this article does not return the turtle to its starting position.

Making objects scalable

Objects that are movable have a useful level of flexibility. But there are other levels as well. Creating objects whose size can be changed is another desired drawing feature. This can be done in a straightforward manner by adding a parameter to the drawing procedure that controls the size (scale) of the figure being drawn. Let's first take a look at the technique for scaling a figure and then discuss why it works. The process is easy and mechanical and done with the Logo editor.

To add a parameter to a procedure that draws an object so that the object can be effortlessly resized, do the following:

Step 1. Add a parameter to the title line (say **:x**)

Step 2. Append the expression *** :x** to each number in the body of the procedure that is related to either forward or backward turtle movement. For example, **fd 20** would become **fd 20 * :x**.

Example:

to example	becomes	to example :x
fd 50	becomes	fd 50 * :x
rt 45		rt 45
bk 20	becomes	bk 20 * :x
end		end

Explanation:

The parameter **:x** is actually a multiplier. It will modify the linear turtle movement by a factor of **:x**. If **:x** has a value of 2, the figure will be drawn twice as large as when originally created. If **:x** has a value of .5, the figure will be half of its original size.

The key element of this technique is that it preserves all of the proportions of the original figure. Once the figure has been created, the hard work has been done. Modifying the corresponding procedure is an exercise in using the Logo editor. Note that all angles must remain the same. If they are changed, then the figure will be changed and in some cases totally unrecognizable. An example of a procedure that draws a sailboat and its modified version that scales the size of the boat are shown below.



```

to sailboat
rt 30 fd 80 rt 120 fd 80
lt 240 fd 80 bk 40 rt 90
rt 90 fd 60
rt 135 fd 20
rt 45 fd 100
rt 90 fd 14
rt 90 fd 114 bk 60 lt 90
fd 80 bk 10 lt 60
fd 10 rt 120 fd 10
end

```

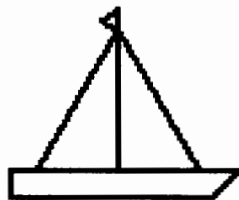
```

to sailboat :x
rt 30 fd 80*x rt 120 fd 80*x
lt 240 fd 80*x bk 40*x rt 90
rt 90 fd 60*x
rt 135 fd 20*x
rt 45 fd 100*x
rt 90 fd 14*x
rt 90 fd 114*x bk 60*x lt 90
fd 80*x bk 10*x lt 60
fd 10*x rt 120 fd 10*x
end

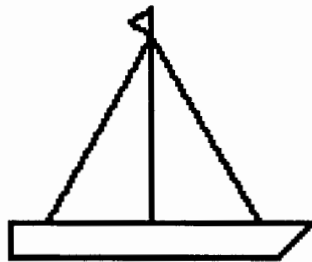
```



SAILBOAT .5



SAILBOAT .75



SAILBOAT 1

Flipping: changing the direction of an object

Many objects, like a sailboat or an airplane, tend to point in a specific direction, and it is often desirable to draw a similar sailboat or airplane that points in the opposite direction. This change of direction is sometimes referred to as a horizontal flip. The object is flipped horizontally about an imaginary vertical axis.

The ability to flip an object adds to its versatility. It can be accomplished in one of two ways. One technique is to take the drawing procedure

and replace all of the **lt** commands with **rt** and replace all of the **rt** commands with **lt**. Thus the instruction **rt 60** would be replaced by **lt 60**, and the instruction **lt 45** would be replaced by **rt 45**.

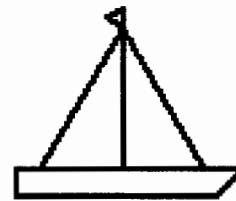
Alternately, we can multiply the inputs of all **rt** and **lt** instructions by **-1**. Using this technique, we would replace the command **rt 60** with **rt -60** and **lt 45** with **lt -45**. The second technique lends itself nicely to using a parameter to handle the multiplication (change of +/- sign).

Shown below is procedure **sailboat**, modified with a second parameter that can be used to flip the boat horizontally.

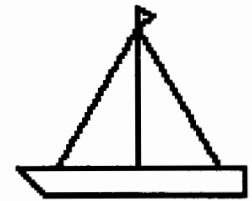
```

to sailboat :x :y
rt 30*y fd 80*x rt 120*y fd 80*x
lt 240*y fd 80*x bk 40*x rt 90*y
rt 90*y fd 60*x
rt 135*y fd 20*x
rt 45*y fd 100*x
rt 90*y fd 14*x
rt 90*y fd 114*x bk 60*x lt 90*y
fd 80*x bk 10*x lt 60*y
fd 10*x rt 120*y fd 10*x
end

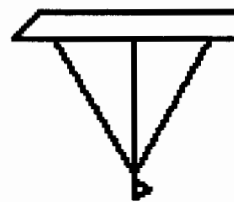
```



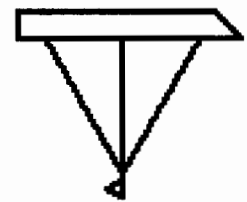
SAILBOAT 1 1



SAILBOAT 1 -1



SAILBOAT -1 1



SAILBOAT -1 -1


The orientation of these sailboats can be summarized by the following:


- :x > 0, :y = 1 original figure
- :x < 0, :y = 1 figure rotated 180°
- :x > 0, :y = -1 horizontal flip
- :x < 0, :y = -1 vertical flip


PREPARE FOR YOUR TEL•ED


MIGRATION THIS FALL.



 ISTE's SIXTH INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS AND MULTIMEDIA IN EDUCATION is migrating to Austin, Texas, and Mexico City, Mexico, in November — and as an education stakeholder of the '90s who understands the dramatic, positive changes interactive technologies have made in the classroom, you should be too.

 Exponential growth on the Internet. Transformations in the use of the World Wide Web. These developments have made a revolutionary impact in education over the past decade — literally opening up the classroom walls to new collaborations and innovative ways of communication.

 ISTE's Tel•Ed has been at the forefront of this educational metamorphosis since 1989, providing practical solutions to better integrate multimedia and telecommunications into our diverse learning environments. ISTE's multisite Tel•Ed conference in 1997 will offer hands-on workshops, classroom-tested sessions, professional networking, a cutting-edge trade show and much more, inspiring you to take your students of today into the learning opportunities of tomorrow.

 So don't let the Tel•Ed migration pass overhead. Spread your wings and Take Flight in the Digital Age.

Austin, Texas, USA  Mexico City, Mexico
November 13-16, 1997

To receive your FREE Advance Program
or for more Tel•Ed information, please contact ISTE:

Phone: 541/346-2472 • Fax: 541/346-5890

WWW: <http://www.iste.org>

Internet: laurie_thornley@ccmail.uoregon.edu

Presented By:


International Society for Technology in Education (ISTE)

Texas Computer Educators Association (TCEA)

Instituto Latinoamericano de la Comunicación Educativa (ILCE)



TEL•ED '97

Taking Flight In the Digital Age 

ISTE BRINGS THE WORLD OF TECHNOLOGY CLOSER TO YOU.

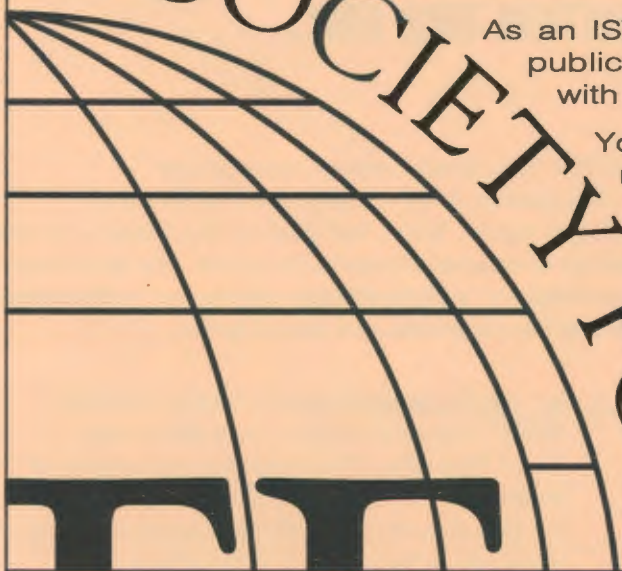
By drawing from the resources of committed professionals worldwide, ISTE provides support that helps educators like you prepare for the future of education.

As an ISTE member, you benefit from a wide variety of publications, national policy leadership, and our work with Teacher Accreditation.

You also enjoy exciting conferences, global peer networking, and graduate-level Distance Education courses.

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

AL SOCIETY FOR



TECHNOLOGY



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: cust_svc@ccmail.uoregon.edu
World Wide Web: <http://www.iste.org>

WE'LL PUT YOU IN TOUCH WITH THE WORLD.