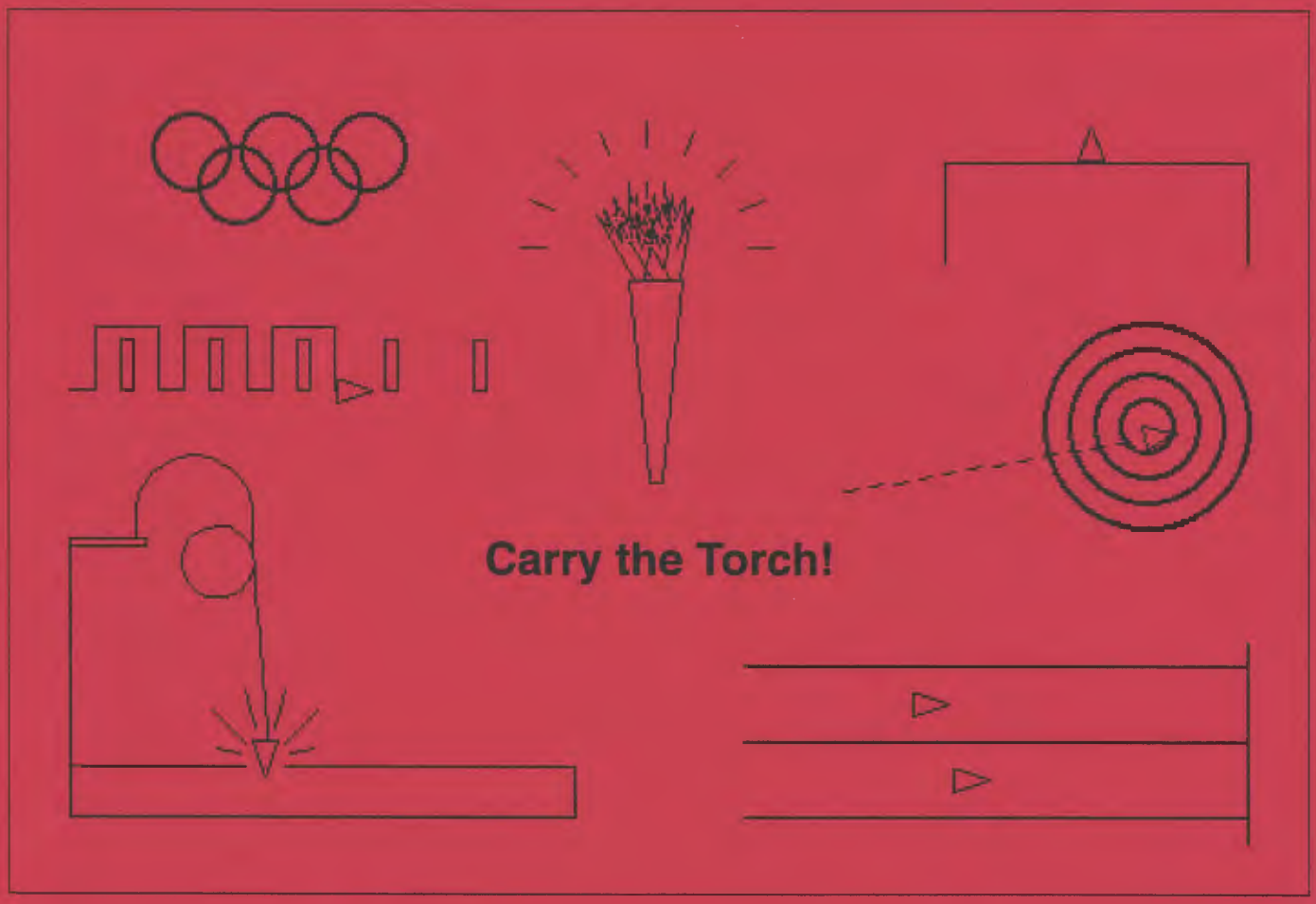
LOGO EXCHANGE

Volume 14 Number 4 Summer 1996

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



In this issue:

Editorial Publisher
International Society for Technology in Education

Editor-in-Chief
Dorothy M. Fitch

Copy Editor
Jennifer Roland

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
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
A Derby Road
Derry, NH 03038

Send electronic mail to:

Internet: 71760.3668@compuserve.com

Deadlines

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

Volume 15, Number 2	June 1, 1996
Volume 15, Number 3	Oct. 1, 1996
Volume 15, Number 4	Jan. 1, 1997
Volume 16, Number 1	Mar. 1, 1997

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*. 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.

POSTMASTER: Send address changes to *Logo Exchange*, ISTE, 480 Charnelton 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 4 Journal of the ISTE Special Interest Group for Logo-Using Educators Summer 1996

Contents

From the Editor: Carrying the Torch

by Dorothy M. Fitch

In this issue we salute torchbearers, in both the figurative and literal senses. These torchbearers light the path ahead for the rest of us. They courageously guide us, shedding new light on familiar ideas and encouraging us to explore uncharted territory. The *American Heritage Dictionary* defines a torchbearer as: "1. One that carries a torch. 2. One...who imparts knowledge, truth, or inspiration to others."

LX's founding editor, Tom Lough, exemplifies both definitions of torchbearer. He has always carried a torch for Logo. Fourteen years ago it was his idea to start a newsletter for Logo users, which has evolved into this journal. He has been a guiding light for us all since then.

And now, Tom truly will be carrying a torch! Tom has been selected as a torchbearer to carry the Olympic flame through Hartford, CT for one mile on June 16. How exciting!

Not only that, but modest Tom, I have just learned, was a member of the USA Modern Pentathlon team at the 1968 Olympics in Mexico City! (The modern pentathlon includes horseback riding over jumps, fencing, pistol shooting, swimming, and running.) Having known Tom for so many years and just now finding out this exciting information was truly awesome, in the true sense of the word. As he wrote in an e-mail message, which he has allowed me to share:

> I have not really kept my Olympic participation a secret; I just didn't proclaim it to everyone. I enjoy sharing my Olympic experiences with school groups and others. I take pleasure in encouraging people to dream and then to start on the path to realize their dreams. From my Olympic experience, I learned many valuable lessons, which I still try to apply in my everyday life. And, as you might expect, having represented our country in international competition has meant a lot to me.

This year, Tom has been instrumental in getting an online project called CyberTorch up and running. "CyberTorch: An Educational Project of Heroic Proportions" is a grass-roots project facilitated by the Global SchoolNet Foundation, a nonprofit organization established by Al Rogers of FrEdMail fame.

The purpose of the project is to provide a framework for schools, teachers, and students to incorporate the 1996 Olympic Torch Relay into their classroom activities. Across the country, teachers are developing math and science projects, geography lessons, language arts assignments, and physical education events that relate to the torch relay. Some students are interviewing local torchbearers, while others are designing and making mockup torches for them to carry in their running workouts.

Schools share their experiences via the CyberTorch online resources, or search the information for torch activity ideas of their own. What Logo-related projects can you think of for the CyberTorch project?

Check out the CyberTorch home page on the Web at http://www.gsn.org/torch/. You can also sign up for the torch-talk newsgroup of CyberTorch by sending an e-mail message to majordomo@gsn.org. Leave the subject line blank. In the body of the message, type the single line: subscribe torch-talk. Then send the message. Follow the route of the torch on the Coca Cola Torch World Wide Web page at http://www.cocacola.com/olympics/torchrelay.html.

Another Olympian feat is being a columnist for a computer education journal for 14 years! Glen Bull, one of Tom's original columnists, has done just that. He, too, is a torchbearer, having guided us in exploring Logo ideas, activities, and connections for so many years. Glen and Gina Bull have decided to step down from their time-honored position at the end of this year. Their last column is a fascinating and personal reminiscence about *Logo Exchange* and how Logo has

affected them. We wish them all the best and hope to hear from them again from time to time. We thank them here (and also at the end of their column) for their great contribution to the Logo community.

And Tom, as you run with the Olympic flame on June 16, we'll all be thinking of you—every step of the way.

```
repeat 5280 [fd 100!]
```

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

P.S. Tom tells me that the May/June 1984 issue of *Teaching and Computers*, published by Scholastic, Inc., includes a Logo article called "Light the Way to Logo With Logo Olympics" by Tom Lough, Steve Tipps, and Rebecca Poplin, pages 20–28. In the "Light the Torch" event, students use the turtle to draw steps leading up to the Olympic torch stand so that the flame can be lit. In the process, they learn something about the slope of a line. Dig out this article, if you can!

About the cover

The cover pictures offer ideas for designing your own "Turtle Olympics" events. These pictures were part of a Summer 1992 *LX* article by Dorothy Fitch. It seems appropriate and timely to share them again now. As the Summer Olympics approach, help your students design Logo activities to honor our world's athletes. Write a program for your favorite Olympic event. Draw a sporting scene. Use animation to show a turtle (or several turtles) in motion.

Quarterly Quantum: Let It Snow!

by Tom Lough

You may not believe this, but as I write this column, I am surrounded by snow. This winter in Connecticut was something else! At one point, we had nearly three feet of the white stuff on the ground! And more is on the way.

Naturally, with so much of it around, my thoughts turned to snow, and to snowflakes in particular. (And for those of you reading this summer issue of *LX* in warm or warmer temperatures, maybe some Logo snow might make you feel a little cooler!)

I recalled reading about the scientists and teachers working in the Antarctic this year. One of the research activities involved analyzing cores drilled out of deep holes in the ice. From an examination of the ice crystals, researchers hoped to learn something about conditions on earth when the ice was just snow on what was then the surface.

What a fascinating concept! Information frozen in time, embedded in the snowflakes.

I began to wonder how we might encode information in snowflakes drawn with the Logo turtle. Hmmm.

I started with the assumptions that every flake has six identical spokes and that all angles between snowflake parts are 60 degrees. After a lot of thinking and messing around, I finally set up a snowflake representation using a list of number lists.

Each number list corresponds to a spike on each of the six snowflake spokes. The numbers in the number list determine the size of the snowflake.

(Don't worry if this doesn't make much sense yet. Just wait until you try the procedures.)

Let's focus on some ideas for using the procedures in the classroom to help students understand how information can be encoded in designs such as snowflakes.

After you define the procedures (listed at the end of this column), you can draw a snowflake by simply typing **flake** followed by a list containing one or more lists of numbers. Here are a few examples. Notice that the lists of numbers are grouped by square brackets into a single list (called **spokelist** by the procedures).

```
flake [ [2 2]  [4 1] ]
flake [ [4 3 2 1]  [7 5 3]  [9] ]
flake [ [5] ]
```

Students like to figure out how the numbers affect the size and design of the snowflake. For convenience, each number in a list is multiplied by a factor of 10 in the **spike** procedure. This allows students to enter small numbers (even decimal numbers, such as 2.4) and still get reasonably large results.

Eventually, the snowflake drawing activity can be turned inside out, often by a student who asks another, "Betcha can't guess my numbers by looking only at my snowflake."

When this happens, the students are doing some thinking similar to that of the Antarctic researchers. They are trying to make sense of what they are seeing and to extract information (or even estimates of information) from their observations. Unlike the polar scientists, students can test their hypothetical data by drawing a snowflake based on it and comparing it with the original.

After students get acquainted with the **flake** procedure, some of them will prove to be amazingly adept at estimating what the input list looks like.

What other activities and thinking processes can you develop around these ideas? I'd love to hear about them.

If you are interested particularly in the programming aspect, feel free to dissect and adapt

the procedures in any way you like. I would enjoy learning what you do with them.

Above all, keep cool! Until the next snowstorm,

FD 100!

Tom Lough
Founding Editor
P.O. Box 394
Simsbury, CT 06070
E-mail: 70020.223@compuserve.com

P.S. As an extension, develop a procedure to generate randomly the required list of number lists for drawing a different snowflake each time. I would appreciate seeing a copy of what you did. If you would like a listing of this procedures I developed, send me a self-addressed stamped envelope and request "Snowflake Design" procedures. Overseas subscribers please send equivalent international postage-return coupons.

```
to flake :spokelist
repeat 6 [spoke :spokelist right 60]
end

to spoke :spokelist
if :spokelist = [ ] [stop]
spike first :spokelist
spoke butfirst :spokelist
end

to spike :spikelist
if :spikelist = [ ] [stop]
forward tenfirst :spikelist
left 60
spike butfirst :spikelist
right 120
spike butfirst :spikelist
left 60
back tenfirst :spikelist
end

to tenfirst :spikelist
output 10 * ( first :spikelist )
end
```

The Historical Turtle

by Jim Muller

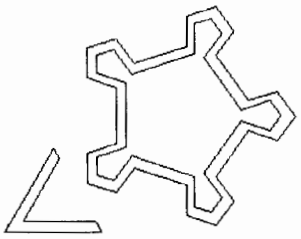
The day always seems to arrive in any Logo fan(atic)s life when you seem to run out of ideas. You've done every type of geometrical exercise and word game you can think of. There has to be something more than just another graphics project.

Geography is a subject that offers many possibilities. What about history?

While working with middle school classes some time ago, we looked at several famous battlefields from the early days of this country. In addition to the Alamo—what else would Texans study—we had a lot of fun with the Battle of Baltimore in the War of 1812.

The British brought a large naval force up the Chesapeake Bay and into Baltimore Harbor. They knew that their ships had more and larger guns than the rather small fort that guarded the entrance to the city.

Fort McHenry is small when compared to many other forts built in the 18th and 19th centuries. It is a star-shaped structure that sits on a point of land that juts out into Baltimore Harbor, which makes it the perfect spot to protect the city.



The barracks and other buildings line the inside walls of the fort. In the center is a large flagpole that made the fort and the battle famous. It was on that pole that the Star Spangled Banner flew.

Earlier in the war, the British had captured William Beanes, an American from Maryland. He was being held on one of the British ships moving into Baltimore. Francis Scott Key and John Skinner, both from Washington, DC, received the permission of President James Madison to negotiate for the release of Beanes.

The British agreed to release Beanes. However all three men would have to remain on board the prisoner exchange ship until after the battle so they would not be able to warn the American patriots that they were coming.

The bombardment started on Tuesday, September 13, 1814, and continued all that day and almost all night. Key and his friends knew that Fort McHenry was no match for the huge guns of the British fleet. Sitting well back from the battle line, they paced the deck of their ship all night. Even when dawn broke, they did not know who had won the battle because the smoke and haze was so thick.

Suddenly there was a break in the smoke and haze and the Americans could see the American flag still flying over the walls of the fort. Key was so excited, he pulled an unfinished letter from his pocket and began to express his feelings.

> Oh! Say can you see, by the dawn's early light,
> What so proudly we hailed at the twilight's last gleaming?

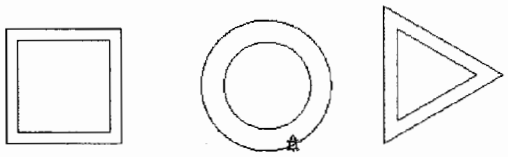
Key completed most of the poem within a few minutes. Shortly after that, the British gave up the fight and left Baltimore Harbor, leaving the three Americans behind. The poem was printed on handbills the next morning and distributed throughout the city as the citizens celebrated their victory. A few days later, it was sung to the tune of an English marching song that the Americans had adopted and named "Adams and Liberty." "The Star-Spangled Banner" became famous immediately and was sung at the Battle of New Orleans three months later.

"The Star Spangled Banner" is certainly an inspirational story for the children. But what

they couldn't understand was why the British lost the battle. They had many battle ships with more and larger guns than the fort. They had a large force of ground troops, many more than the rather small American force that was defending the city.

One of the things we talked about was the shape of the fort. It was shaped like a five-pointed star. The inner and outer walls of the fort were high and very thick. At each point of the star, there was a gun emplacement that allowed the gunners to cover every possible angle. Protected by high, thick walls and hidden in a small place that was difficult to hit, the guns were well protected.

In our Logo sessions, we had gotten to the point where we were discovering how to put shapes within shapes: squares spaced within squares, triangles spaced within triangles, for example.



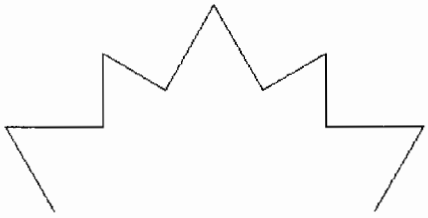
Looking at the diagram of the fort, it became a challenge to recreate the star image.

```
to fort.mchenry
outside.wall
move
inside.wall
move2
ravelin
end

to outside.wall
repeat 5 [outside.corner]
end

to outside.corner
forward 46 left 72
repeat 3 [forward 25 right 72]
forward 25 left 72
end

to move
pu right 90 forward 8
left 90 back 3 pd
end

to inside.wall
repeat 5 [inside.corner]
end

to inside.corner
forward 53 left 72 forward 25 right 72
repeat 2 [forward 15 right 72]
forward 25 left 72
end

to move2
pu back 70 left 90 forward 20 pd
end

to ravelin
repeat 2 [forward 80 right 120]
forward 10 right 60 forward 60
left 120 forward 60 right 60 forward 10
end
```

Ravelin? What's a ravelin? A ravelin is a triangular fortification outside the main defensive structure; in this case, outside the fort.

There were actually several ravelin embankments that faced the harbor as a first defense against an invasion from the sea. It shielded men and guns from an advancing enemy.

One portion of the ravelin is shown with the diagram of the fort. Here's another to show what the actual fortification looked like.



Here's a challenge for you and your students: Add the actual ravelin shapes to the picture of the fort.

Those who want to take a look at the Battle of Baltimore from the British perspective and why they failed to silence the American guns can read the companion article starting on page 22 of this issue, "The Artillery Battle of Baltimore."

Jim Muller is the moderator of the Logo Forum on CompuServe. You can read more about him at the end of his other article in this issue, "The Artillery Battle of Baltimore." You can reach him via e-mail at 76703.3005@compuserve.com or on CompuServe at 76703,3005.

Logo Workshops: Summer 1996

Logosium — Minnesota

June 14 in St. Paul

The third annual Logosium in St. Paul, Minnesota 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.

Registration details appear on page 27.

Logo St. Paul — Minnesota

June 17–21 or August 19–23 in St. Paul

Over the past 14 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.

Registration:

$490 per person; $126 for university credit

For more information, contact:

Michael Tempel
Logo Foundation, 250 West 57th Street
New York, NY 10107-2228
Telephone: 212-765-4918; Fax: 212-765-4789
E-mail: michaelt@media.mit.edu

Summer at Spence — New York

June 24–28 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. Join the Spence staff for a week of Logo exploration and creation.

Registration:

$790 per person

For more information, contact:

Michael Tempel
Logo Foundation, 250 West 57th Street
New York, NY 10107-2228
Telephone: 212-765-4918; Fax: 212-765-4789
E-mail: michaelt@media.mit.edu

Stonington Retreat — Maine

June 30–July 6 on Deer Isle

Led by Laura Allen and Michael Tempel, the Stonington Retreat is an experiment in creating a reproducible learning experience for teachers interested in using computers and in thinking about their implications for the future of education. Each day will include a morning meeting, reflection and discussion, hands-on lab time, a local culture talk, and free time to explore Logo or the area.

Registration:

$1,500–1,600 per person, includes tuition, bed and breakfast lodging, all breakfasts, three dinners, and a scenic boat ride.

For more information, contact:

Laura Allen
P.O. Box 20616, Cherokee Station
New York, NY 10021-0071
Telephone: 212-744-2447 or 617-354-2370
E-mail: laurallen@aol.com

Pepperdine University Summer Logo Institute & Conference

August 5–10 in Culver City, CA

Led by Gary Stager, the Summer Logo Institute (August 5–9, plus the Logo Conference) provides a unique collegial environment in which educators can learn Logo, enhance their programming skills and reflect upon the roles computers may play in the creation of constructive learning environments for children.

The Logo Conference (August 10) is an opportunity for those people unable to attend the Institute to recharge their batteries, learn Logo, share classroom project ideas, and attend minds-on workshops led by Institute participants and exciting guest speakers.

Registration options:

1) Institute and Conference, $499
2) Logo Conference on August 10, $75 (includes lunch)

Local hotel accommodations are available.

For more information, call: 310-568-5704
E-mail: GStager@Pepperdine.Edu

The Hudson River Project: Computing the Water Quality Index using *MicroWorlds*

by Orlando Mihich and Students

Introduction

The goal of this ongoing project is to help students develop into citizens who possess the knowledge, skills, and attitudes to engage in environmentally responsible behaviors. Booker T. Washington, Middle School 54, is located on the Upper West Side of Manhattan. The Hudson River, one of the most beautiful sites in New York, is within walking distance from the school. The question is: How is the Hudson River doing? The project began three years ago with the support of an enlightened administration and a yearly budget of $250–$300.

Project Description.

Once a week, a group of seventh- and eighth-grade students, "The Water Watch Team," visits the Boat Basin on the Hudson River, at the 79th Street Marina. Here, students record air and water temperatures, take pH readings, determine the water clarity and dissolved oxygen levels, and prepare a sample for the five day "biochemical oxygen demand" test. Sewerage pollution is tested using a Coli-count sampler. A water sample is brought back to school to determine levels of nitrates, phosphates, and solids.

In their work, students follow an "Internal Manual" compiled from different sources: the literature that comes with the suppliers' test kits, various electronic encyclopedias, and mainly, the "Field Manual for Water Quality Monitoring." The "Internal Manual" was written in part by students and contains a general section on the world's rivers, their history, the watershed, and some information on the utility and history of the Hudson River. For the most part, our "Internal Manual" conforms to the procedures outlined in the "Field Manual for Water Quality Monitoring." Other procedures requiring open-flame burners, filtration, and the use of sulfuric acid or other harsh chemicals have been modified. For example, the Coliforms test is performed using the pocket-size Millipore Coli-count sampler. Prepared on location, the sampler is incubated at 35° C for 22–24 hours in the school's lab.

Following is a brief description of the importance of the nine tests.

1. *Dissolved Oxygen (DO).* Dissolved oxygen is a measure of the health of a body of water. The absence of oxygen indicates severe water pollution.
2. *Fecal Coliform.* Fecal coliform levels are monitored because of the correlation between viruses that cause diseases and fecal coliform and pathogenic bacteria.
3. *pH.* Natural waters have a pH value from 5.0 to 8.5. Higher or lower levels are unsuitable for most organisms.
4. *Biochemical Oxygen Demand (BOD) [5-days].* Biochemical oxygen demand is a measure of the quantity of the oxygen that disappears from the water because of the decomposition of organic matter.
5. *Temperature.* Thermal pollution from industries, urban runoffs, soil erosion, and deforestation affect the amount of dissolved oxygen and rate of photosynthesis.
6. *Turbidity.* Increased amounts of suspended solids reduce the transmission of light and cause waters to become warmer, with a consequent drop in oxygen levels.
7. *Phosphates.* The amount found in healthy water is generally small, not more than 0.1 ppm. Larger amounts of phosphates usually wipe out the river's fish population.
8. *Nitrates.* The presence of excessive amounts of nitrogen promotes plant growth and decay, which in turn increases biochemical oxygen demand.

9. *Total Solids.* High concentrations of suspended solids reduce water clarity, and contribute to a decrease in photosynthesis.

Students enter all test results into a *ClarisWorks* spreadsheet and generate charts for a discussion of results. Once all nine tests are completed, students use *MicroWorlds* to find the Water Quality Index.

Computing the Overall Water Quality Index

The Water Quality Index (WQI) is a standard index developed by the National Sanitation Foundation to compare rivers' waters. The WQI ranges from 100 for Excellent Water Quality to 0 for Very Bad Water Quality. To obtain the WQI number, it is necessary first to find the **Q-value**, the quality value, for each test. This is done with a series of nine "weighting curves," one for each test. Once found, the Q-value is multiplied by a "weighting factor" to obtain the **Total Q-value** for the test. The sum of all Total Q-values gives the **Overall WQI** for the testing site.

During the first year of testing, students computed the Q-value by consulting the charts provided in the "Field Manual for Water Quality Monitoring." *ClarisWorks* was employed to find overall WQI.

The second year, students recreated all nine diagrams in *LogoWriter*, and wrote a program to find the Q-value by positioning the turtle on the abscissa and typing "find" in the Command Center. At the same time we realized that several "weighting curves" could be changed and adapted to local conditions; e.g., Total Solids never exceeded 20 mg/L, so we changed the existing diagram's range of 0 to 500 mg/L to a range of 0 to 100 mg/L; the curve was redrawn accordingly. Now, we were able to position the turtle more accurately and achieve more precise results.

With the introduction of *MicroWorlds* it is now feasible to find the Q-value, the Total Q-value, and the overall Water Quality Index without switching to *ClarisWorks*.

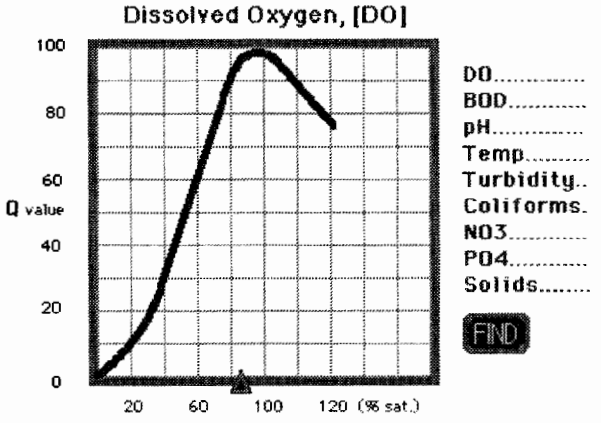
Discussion of results

For the school year 1994–1995, the WQI ranged from MEDIUM to GOOD. Fecal coliform bacteria exceeded 200 colonies/100 mL. The Hudson River's water does not meet the state standard for total body contact such as swimming. Dissolved oxygen levels were healthy, above 4 mg/L. Biochemical Oxygen Demand levels were higher than expected, indicating the presence of large amounts of oxygen-consuming microorganisms.

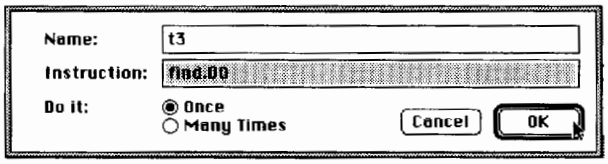
On May 18, 1995, the "Water Watch Team" students were awarded the New York Citywide and Boroughwide first prizes for their work in The Hudson River Water Watch Project as part of the Citywide "Team-Up-To-Clean-Up" Competition. In December 1995, "GREEN," the journal for the Global Rivers Environmental Education Network, published an article describing the students' work.

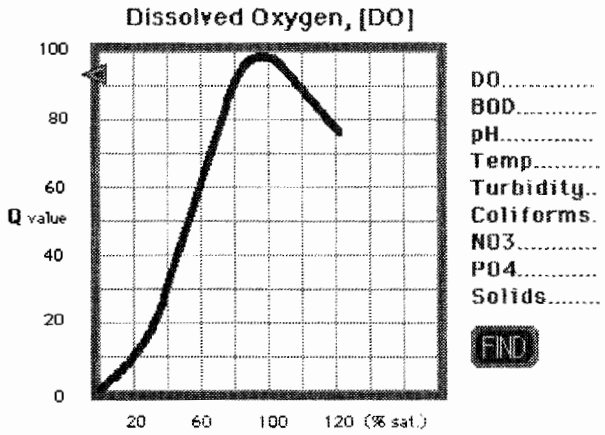
MicroWorlds as a programming tool

Following are sample pages for DO and WQI:

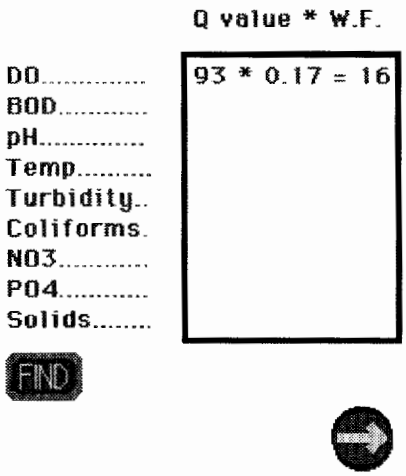


Students position the pointer for the % of DO on the abscissa, then click the FIND button. This turtle, t3, finds the Q-value for DO, using the procedure, **find.DO**.





The pointer travels to the curve, turns toward the ordinate, and travels to it. The Q value shows in the active dialog box.



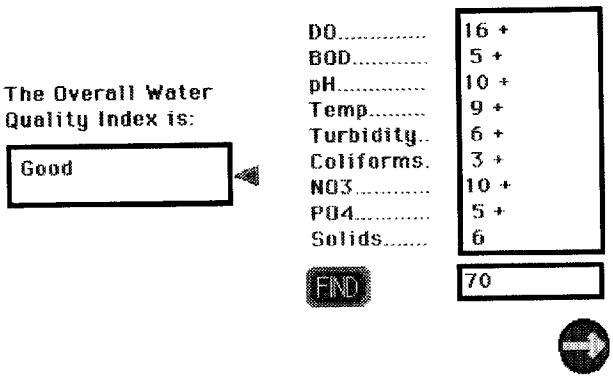
Here, the Q-value is multiplied by the Weighting Factor (W.F.) to obtain the Total Q-value. To continue, students click the ARROW button. The new Biochemical Oxygen Demand page comes up. In the dialog box the Q-value for Dissolved Oxygen is displayed with a plus (+) sign. The sum signs are added throughout with each new result. The last page computes the Overall Water Quality.

Dissolved Oxygen (DO) Procedures

```
to start.DO
; Positions t1 at the origin of the diagram;
  creates the reference turtle "zero.Q;
  creates the textbox "Q-DO.
t1, st
pu setpos [-185 -95]
setsh "pointer1
setsize 20
make "a pos
newturtle "zero
ht
pu setpos :a
newtext "Q-DO [125 100] [120 155]
settc "125
setstyle "bold
end

to find.DO          ;button
; Prevents the accidental clicking of the
  "Find" button when the Total Q-value is
  already found, and when the Total Q-value
  is not yet found. Finds the Q-value.
if alert.DO = 1 [alert.sound stop]
ifelse (ask "t1 [pos]) = :a [announce [In-
  troduce your result on the abscissa.]]
  [DO]
end

to DO
t1,
seth 0
if colorunder = 15 [seth 270 pu fd 10 go
  q.DO stop ]
pu fd 1
DO
end

to Q.DO
; Rounds and adjusts the Q-value for the
  existing 20 x 20 grid; inserts the weight-
  ing factor; gives the Total Q-value.
talkto "Q-DO
insert round value / 2
insert char 32
insert "|* 0.17|
insert char 32
insert "=
insert char 32
insert round (value / 2) * 0.17
end
```

```
to alert.DO
talkto "Q-DO
output textcount "Q-DO
end

to move.value.DO
; Selects and copies only the Total Q-value.
top
select
cd
repeat 2 [cb]
cut
top
select
cd
copy
end

to end.DO              ;button
; Prevents getting the next page before Q-
  value is found; copies the Total Quality
  Value; removes the textbox; gets and
  starts the next page.
ifelse alert.DO < 1
  [announce [Please, find Q-value first.]]
  [move.value.DO remove "Q-DO getpage
  "|Biochemical O2 Demand| start.BOD]
end

to go
setsh "pointer2
if colorunder = 55 [stop]
pu fd 2
go
end

to value
t1, output distance "zero
end

to alert.sound
; plays the Mac "sosumi" sound
sosumi
end
```

After all of the test results have been computed, the final page gives students the Overall Water Quality Index when they click the FIND button.



Water Quality Index (WQI) Procedures

```
to start.WQI
; Creates two new textboxes: "Total displays
  the numerical sum of all "Total Q-values"
  generated throughout the project, and
  "|Water Quality| displays the water qual-
  ity range, from "Excellent" to "Very Bad,"
  according to the numerical value.
t1,
setsh "pointer2
pu setpos [0 0]
newtext "Q-WQI [125 100] [120 155]
settc "125
setstyle "bold
paste
newtext "Total [125 -55] [120 30]
newtext "|Water Quality| [-150 18] [140 40]
end

to find.WQI            ;button
if alert.WQI = 1 [announce [Please,
  continue.] stop]
settotal run Q-WQI WQI
end

to WQI
name total "result
talkto "|Water Quality|
if :result = 100 [print "Excellent]
if and :result > 90 :result < 100 [print
  "Excellent]
if :result = 90 [print "Excellent]
if and :result > 70 :result < 90 [print
  "Good]
if :result = 70 [print "Good]
if and :result > 50 :result < 70 [print
  "Medium]
if :result = 50 [print "Medium]
if and :result > 25 :result < 50 [print
  "Bad]
if :result = 25 [print "Bad]
if and :result > 0 :result < 25 [print
  "|Very Bad|]
if :result = 0 [print "|Very Bad|]
end
```

```
to alert.WQI
talkto "Total
output textcount "Total
end

to end.WQI              ;button
ifelse alert.WQI < 1 [announce [Please, find
  the WQI Index.]]
  [dolist [i [zero Q-WQI Total |Water
  Quality|]] [remove :i]
     Solids
     getproject "WWT+1]
end
```

The last instruction, getproject "WWT+1, brings up a page with a picture of the Water Watch Team.

This project owes its continuing success to the enthusiastic involvement of **Michael Estrada** as student in charge, and the following students: Carolina Alcantara, Nestor Amaya, Socrates Brito, Marilyn Carabajo, Alex Estevez, Rufino Estevez, Lilliana Estrella, Tracy Fields, Laura Guzman, Loamy Hodge, Sam Lazarus, Alexandria Matos, Wilfredo Pichardo, Patrick Mohaney, Haydee Pimentel, Aaron Riccio, Wilmer Ruez, and Maria Solares.

The following page shows *ClarisWorks* sample charts with a brief discussion by the students.▲

Bibliography

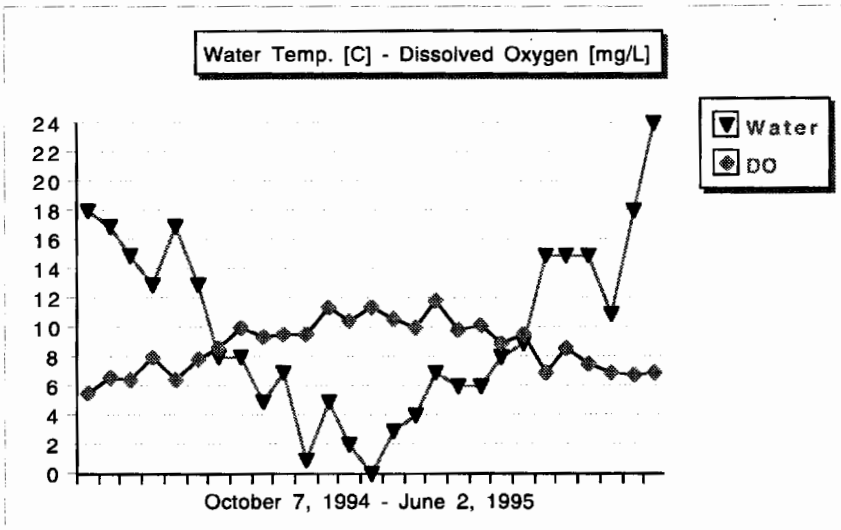
Mitchell, M.K., & Stapp, W.B. (1994). *Field manual for water quality monitoring*. (Available from GREEN, 721 E. Huron Street, Ann Arbor, MI 48103.)

Yoder, S. (1996). *MicroWorlds—Hypermedia project development and Logo scripting*. Eugene, OR: ISTE.

Orlando Mihich is a computer teacher at Booker T. Washington, MS 54, Community School District 3, New York City. He has taught science and Logo for the past 11 years. Prior to that, he was a chemical engineer in the surfactants and cosmetics industries. He can be reached by e-mail at Omihich@aol.com.

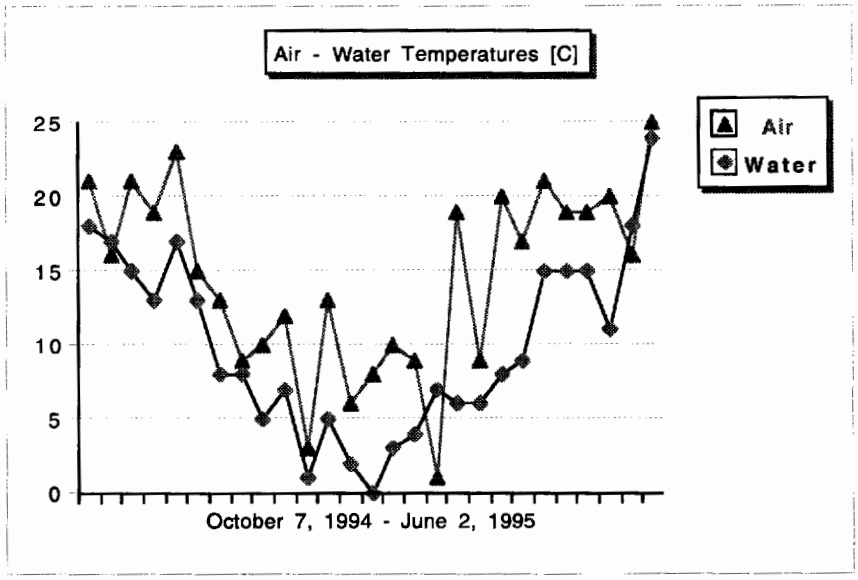
This chart clearly shows that gases, like oxygen, dissolve more easily in cooler water than in warmer water. In addition, healthy levels of DO, above 4 mg/L, were found for the Hudson River at the 79th Street location during the testing period.

By: Laura Guzman
Loamy Hodge



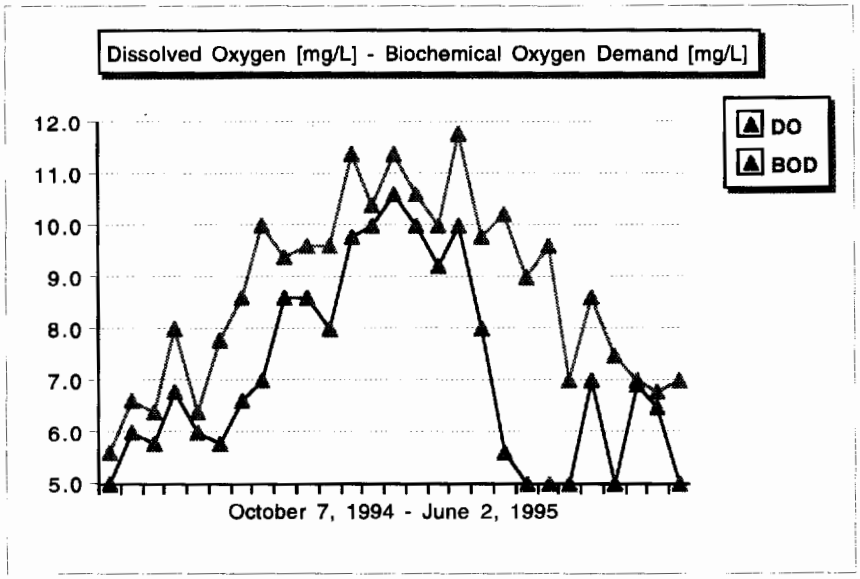
Water temperatures, in general, follow air temperatures, but water, a heavier fluid, responds less drastically to variations in temperature.

By: Alejandro Cruz
Ruez Wilmer



This diagram shows that, with several exceptions, most of the oxygen is used up in the five-day test by organic matter and microorganisms in the water.

By: Michael Estrada



Reaction Time: Fun with Robotics!

[Editor's note: This article was originally published in the premier issue (Spring 1989, Vol. 1, No. 1) of *LEGO DACTA CONNEXION*™, a newsletter for *LEGO® TC logo* Users. Reprinted with permission of LEGO Dacta, the educational division of LEGO Systems, Inc.]

To the Teacher:

Key ideas

1. Learning about the brain and voluntary muscle functions.
2. Practicing the scientific method.
3. Introducing the primitive **random**.

Teaching notes

A *stimulus* is something that causes a reaction. Light, sound, and electrical impulses are examples of stimuli.

The scientific method is often broken down into four or five parts. We will use four parts in this exercise: stating a hypothesis, designing and conducting an experiment, recording results, and reaching a conclusion.

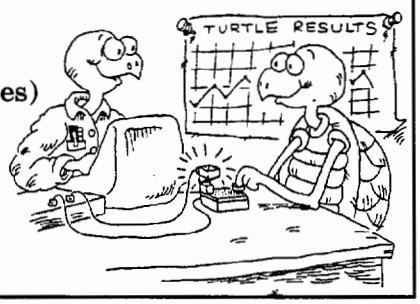
Students may need to be reminded that a hypothesis is an educated guess and that the experiment may disprove the hypothesis. Students should not be concerned about having their original hypothesis be "right." It is more important that experimental results become the basis for new ideas.

Your brain is the busiest, most complex organ in your body. It acts as a message center, sending and receiving messages through the nervous system to all parts of your body. Even the simplest task, such as standing, requires an incoming "balance" message from the ear and an outgoing command message to your legs and feet. The "message center" contains 10 billion neurons, each processing as many as 50,000 incoming and outgoing messages per minute, traveling 100 miles per hour through your body! Although everyone has a message center, we do not all react to stimuli at the same rate of speed.

In this activity, you are going to use the scientific method to examine how quickly people respond to a light stimulus.

Materials needed

1 *LEGO TC logo* system
(or other robotics system with appropriate capabilities)
1 touch sensor
1 light brick
miscellaneous elements
Inventor's Notebook or sheet of paper



Designing an experiment

Before you can set up an experiment, you need a hypothesis—a statement that you can test. For example: Because girls mature faster than boys do, they will react to a stimulus faster than boys will.

1. Record your hypothesis in your Inventor's Notebook or on a sheet of paper.
2. Build a reaction machine.

   The machine pictured on the next page uses a LEGO light brick for the light stimulus. A touch sensor serves as the button that subjects will press to turn off the timer when they notice the light come on. Design your own reaction machine using a light and a touch sensor or an optosensor.

Students may invent different reaction machines. For example, they might replace the touch sensor with an optosensor and light brick, creating an "electric eye" to stop the timer.

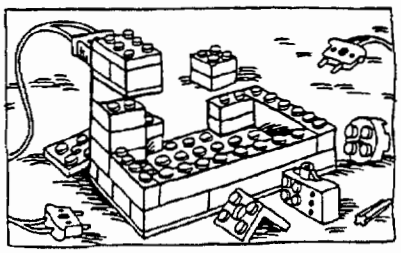
Including **wait random 250** in the procedure adds an element of surprise. Subjects are less likely to learn when the program starts and when the light will turn on.

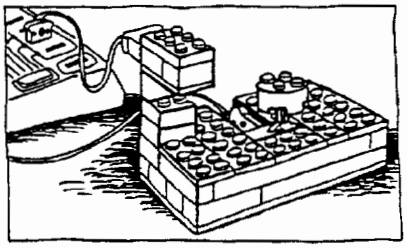
If the student conducting the experiment wishes to control when the light turns on, another touch sensor, in addition to the touch sensor controlling the timer, can be added to the program. Plug another touch sensor into port 7 and try the following procedure.

```
to touch.light
listento 7
waituntil [sensor?]
talkto 1 on
resett
listento 6
waituntil [sensor?]
off
show timer/10
end
```

To make the experiment more impartial, assign a number to each subject. Have students record the name and number of each subject in their Inventor's Notebooks. Example: Jane–34; Sarah–56; Robert–23; and so on. Then response times can be recorded as Subject 34: 5, 4, 5.

To have a fair sampling of subjects for each experiment, it may be helpful for each team to state its hypothesis, set up its experiment, and have all class members act as subjects for each group in turn. Or, as a class, decide on a hypothesis to test. As the initial hypothesis is proved or disproved, each group can develop a new hypothesis.





3. Program your reaction machine. Here is one way to do it:

```
to light.react
wait random 250
talkto 1 on
resett
listento 6
waituntil [sensor?]
off
show timer/10
end
```

The procedure **light.react** does the following:

1. waits a random amount of time between 0 and 25 seconds.
2. turns the light on.
3. resets the timer.
4. listens to the touch sensor.
5. waits until the touch sensor is pressed.
6. turns the light off.
7. displays the timer amount in seconds.

Conducting the experiment

1. Divide your subjects into appropriate groups. For example, if you are testing whether girls are faster than boys, split your subjects into two groups—girls and boys.

   You may want to establish some rules for the test. For example, you may want to position subjects so that they cannot watch the operator. You may also have subjects place their hands a specified distance from the machine—perhaps on the table. *Remember: All subjects must follow the same rules or the experiment will not be accurate.*

2. Conduct tests on your subjects. Tell them to watch for the light to go on. As soon as they see the light, they must press the button.

Students should try at least two tests of response time. Data recorded must be accurate and reliable. The more trials run per subject, the more accurately and reliably data will reflect a subject's reaction time. Also, the more subjects used per experiment, the more accurately and reliably an experiment will reflect the general population.

Challenge

The following program can be used to test reaction time to sound:

```
to sound.reaction
wait random 250
tone 440 1
resett
listento 6
waituntil [sensor?]
off
show timer/10
end
```

Discussion

Is there an age when reactions are fastest? Ask the students to formulate a hypothesis for testing.

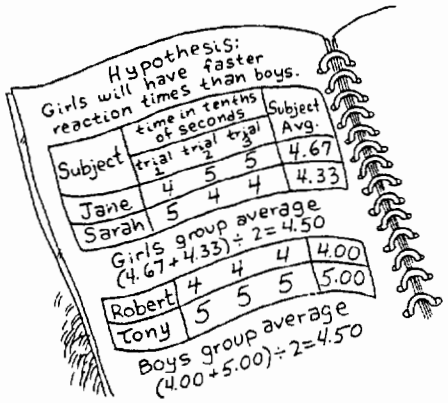
In addition to reacting to a stimulus, the brain is capable of learning. If the subjects practice, will their reaction times improve?

We depend on fast reactions to protect us from harmful stimuli in our environment. For example, fast reactions protect us from becoming badly burned when we touch a hot stove. What are some other examples?

While the subjects were involved in the experiment, their brains were sending messages to other parts of their bodies. What were some of those messages?

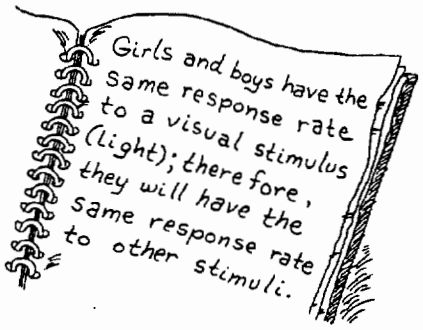
Recording the results

3. As each subject completes the test, his or her reaction time is expressed in seconds on the screen. Write the results in your Inventor's Notebook or on a sheet of paper.



Drawing a conclusion

4. Based on your data, was your hypothesis correct? In the example data, girls and boys have the same average response rates: therefore, our hypothesis seems incorrect. The fact that girls tend to mature faster than boys does not seem to affect reaction time. What other explanations could there be? Perhaps the sample size (2 girls/2 boys) was not large enough to draw an accurate conclusion. Perhaps maturity doesn't affect reaction time at the subjects' age level, or other factors (height, general health, degree of alertness, etc.) are more significant influences on reaction time.
5. Formulate a new hypothesis to test your conclusions.
6. Write your new hypothesis in your Inventor's Notebook or on a sheet of paper.



Challenge

Vision is only one of your senses. Do you think reaction time will differ in a test of hearing? Change your experiment design so that you can test an auditory stimulus rather than a visual stimulus.

Etchings: *LogoWriter* to *MicroWorlds*

by John Gough

I recently came across Web-A-Sketch, an entertaining Internet site on the World Wide Web (http://www.digitalstuff.com/web-a-sketch/) that offers simple instructions and an interactive graphics screen. By pointing and clicking with the computer mouse, you can make a drawing of connected line segments that join from point to point wherever you click.

This idea was based on the popular children's toy "Etch-A-Sketch" (from Mattel), which has two knobs. Rotating one knob moves a "graphic cursor" vertically, drawing a line; turning the other knob moves the cursor horizontally. Turning both knobs simultaneously combines the two directions of movement to produce diagonals and even curves. Web-A-Sketch on the Internet can make only straight lines, but is otherwise similar to this toy.

A project for Logo

Probably the mouse clicking was the mental trigger for me: I suddenly thought this would be fun to do with *LogoWriter*. Because I am still exploring *MicroWorlds*, I thought I would try it first with *MicroWorlds*, believing that any *MicroWorlds* procedures would run, with simple modifications, in *LogoWriter*. But I immediately ran into several interesting difficulties.

How could I program the mouse in *MicroWorlds*?

My Logo experience began a little before the introduction of *LogoWriter*. Soon I began to feel confident with *LogoWriter* on either Apple IIe or IBM—that is, versions of *LogoWriter* with no mouse. A little later I had great fun exploring the Logo-like language HyperTalk for programming in Macintosh's *HyperCard*, which uses the mouse a great deal. Next came Macintosh *LogoWriter* with a mouse, about 250 colors, a microphone, and so on. Perhaps I have been a bit slow to extend my keyboard-programming skill with *LogoWriter* to include mouse programming in the Macintosh version. But I felt confident I could make a *LogoWriter* mouse do more or less what I wanted, at least at the simple level of Web-A-Sketch's point-and-click, even though I knew *LogoWriter*'s mouse was extremely primitive compared to *HyperCard*'s ultrasensitive little mouse. But *MicroWorlds*?

Encountering a mouse problem

The only *MicroWorlds* mouse-related primitive, **mousepos**, reports the current position of the mouse on the screen. I have not (yet) found a *MicroWorlds* equivalent of the *LogoWriter* primitive, **button?**, which reports true if the mouse has been clicked and false if the mouse has not been clicked. But this is the essential ingredient in any point-and-click: We want to know *where* the mouse is currently located after moving (or "pointing") and we want to know *when* the mouse has been clicked. Contrast this with the extremely sensitive mouse of *HyperCard*, which sends signals about whether the mouse button is up, down, or still down; whether the mouse is entering, leaving, or still in a location; and so on.

The *MicroWorlds* mouse can do virtually nothing except manipulate ordinary screen objects such as the cursor and menu bars; click on buttons, text boxes, and tool palette options; and drag on slider bars. It cannot be programmed. Let me put this another way: A mouse click is not an event that *MicroWorlds* recognizes. So, no "Web-A-Sketch" in *MicroWorlds*.

Reverting to *LogoWriter*, it was easy to make a point-and-click, dot-to-dot drawing program that uses the mouse. Notice in **checkpos** that clicking anywhere below a y-coordinate of -80 stops the drawing, and clicking anywhere to the right of an x-coordinate of 200 erases the current drawing. The intermediate procedure **begin** lets the user choose any starting point, and from then on **continue.on** lets the user click anywhere, with the turtle drawing from point to point as the user moves and clicks the mouse.

```
to start.sketch
pu setpos [-100 -95]
pd label [Click here to stop]
pu setpos [200 0]
pd label [ERASE!]
pu setpos [-200 90]
pd label [Move the mouse and click to draw
  lines]
pu setpos [0 0]
begin
end

to begin
ifelse button?
  [checkpos make "pos1 mousepos
    pu setpos :pos1 continue.on]
  [wait 5 begin]
end

to continue.on
ifelse button?
  [checkpos pd setpos mousepos]
  [wait 5]
continue.on
end

to checkpos
if (last mousepos) < -80 [stopall]
if (first mousepos) > 200 [cg begin]
end
```

This *LogoWriter* program can be extended by specifying locations (a *LogoWriter* equivalent of a "button") where the mouse can be clicked to make the turtle lift or lower the pen, change pen color, make selected letters appear at the place clicked, or print the resulting picture.

An alternative solution?

Having been blocked in a mouse-driven point-and-click drawing program for *MicroWorlds*, I wondered if I could draw in Etch-A-Sketch style using arrow keys instead of the mouse. But again there were difficulties. Easily written in *LogoWriter*, the following procedures allow Etch-A-Sketch drawing with keyboard input using arrow keys.

```
to etch
cg cc
show [Use arrow keys, U for up, D for down,
  P to print, Q to quit]
sketch readchar
end

to sketch :input
if 28 = ascii :input [setx xcor - 2]
if 29 = ascii :input [setx xcor + 2]
if 31 = ascii :input [sety ycor - 2]
if 30 = ascii :input [sety ycor + 2]
if :input = "U [pu]
if :input = "D [pd]
if :input = "Q [stopall]
if :input = "P [printscreen stopall]
sketch readchar
end
```

Incidentally, the turtle begins drawing from the centre of the screen. It would be possible to incorporate a version of **begin** that would let the user shift the turtle "cursor" to a starting point, then begin drawing from that point. Alternatively, pressing U, then using arrow keys to move, then pressing D, moves the turtle to any desired starting point without drawing.

The following short procedure **check.a** helped me find the ASCII codes I needed for the arrow keys.

```
to check.a
make "which readchar
print ascii :which
end
```

In *LogoWriter*, virtually all the keys are individually accessible through either **key?** or **readchar**. The use of ASCII codes allows access to keys that do not normally type text. The mouse, as a programmable object, sends messages to the command **button?** in the same way that the keyboard sends messages to the command **key?** or **readchar**.

Reading characters in *MicroWorlds*

Despite the similarities between *LogoWriter* and *MicroWorlds*, things were not immediately so easy in *MicroWorlds*. I had deliberately made the arrow-key-driven *LogoWriter* program using primitives I knew existed in *MicroWorlds*, replacing **rg** and **print**, which I would normally have used, with **cg** and **show**. After careful cutting and pasting from *LogoWriter* to

MicroWorlds flipsides, the program, quite simply, refused to run in *MicroWorlds*. These *LogoWriter–MicroWorlds* procedures try to do Etch-A-Sketch drawing with keyboard input using arrow keys. But the program stops when it gets to the **sketch readchar** line.

Why? It seemed that **readchar** was refusing to work. I looked at the Vocabulary entry for **readchar** in the menu bar option Help, but at first was none the wiser. I looked for alternatives, such as using the built-in *MicroWorlds*' dialog box commands **question** and **answer**, which allow a user to type any text input. I discovered that in *MicroWorlds*, the command **question** is not sensitive to nonalphanumerics, such as arrow keys. When **check.alphanumerics** runs, it is impossible to get any response when pressing nonalphanumeric keys, such as arrow keys, the Tab key, the Return key, and the Space bar.

```
to check.alphanumerics
question [Type a character, and press
  Return]
make "which first answer
show :which
show ascii (first :which)
end
```

The procedure **check.ascii**, a *MicroWorlds* version of **check.a**, was no more successful.

```
to check.ascii
show [Press any key]
make "which readchar
show :which
show ascii :which
end
```

I even pasted **getinfo** from the Help Vocabulary entry for **readchar** and it didn't work. The gnashing of teeth was very audible.

```
to getinfo
show [Type your choice]
make "answer readchar
show :answer
end
```

In desperation I tried to make sense of the fine print in the online Vocabulary Help for **readchar**:

> You must click on the background of the page (outside of a text box, the Command Center, or Procedures page) for **readchar** to recognize the character typed.

So I tried running **getinfo** again, reached the point where **show** told me to "Type your answer," moved the mouse to the upper part of the screen and clicked, and then typed my choice—BINGO!

An awkward solution

The command **readchar** *does* work in *MicroWorlds*, but only if you click first with the mouse in the Front of the page. That is, you can't just run the command **readchar**. You have to shift the current location of activity from the Flipside, the Command Center, a textbox, or anything else to the Front of the page. By contrast, in *LogoWriter*, **readchar** works as soon as a key is pressed, regardless of where things were most recently active. So the instruction provided by the line

```
show [Type your choice]
```

should really read

```
show [Click near the turtle, then type your
  choice].
```

Requiring a user to click before typing, seriously interferes with smooth, simple user input. It is at least as potentially disruptive as requiring a user to press the Return key after typing any input, as *LogoWriter* requires for the commands **readlist** and **readlistcc**. It also makes the Command Center unavailable for user interaction, and, similarly, textboxes and other objects that should be simply keyboard reactive.

```
to draw.keyboard
cg show [Click near the turtle. Use arrow
  keys to move]
pd draw
end
```

LOGO LAUNCHES LEARNING

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 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

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 Time Designs—Working With Patterns in LogoWriter

Wacky Words for Traveling Tales—Logo Activities for Language Arts

As of November 1, 1995, ISTE's Area Code changed from 503 to 541