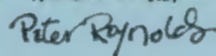


Logo Exchange

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



INSIDE

- Fred D'Ignazio
Down Under
- Prime Time
Math Challenge
- Logo in Russia
- Making MicroWorlds
Music
- StarLogo Starters
- Drawing with Arcs
- Clever Kids
and Computers
- Logo and Jazz
- Book Review,
Logo News,
Teacher Feature



Logo Exchange

Volume 16 / Number 3

Editorial Publisher

International Society for Technology in Education

Editor-in-Chief

Gary S. Stager, Pepperdine University
logoexchange@moon.pepperdine.edu

Copy Editing, Design, & Production

Ron Richmond

Founding Editor

Tom Lough, Murray State University

Design, Illustrations & Art Direction

Peter Reynolds, Fablevision Animation Studios
pete@fablevision.com

Contributing Editors

Dr. Doug Clements, SUNY Buffalo
Dr. Carolyn Dowling, Australian Catholic University
Alan Epstein, Metasoft
Dr. Brian Harvey, U.C. Berkeley
Daniel E. Kinnaman, University Affiliates
Dr. Julie Sarama, Wayne State University

International Editor

Jeff Richardson, Monash University, Australia

International Editor Emeritus

Dennis Harper, Olympia, Washington School District

SIGLogo Officers

Chuck Friesen, President
Steve Sesko, Vice-President
Hope Chaffian, Secretary/Treasurer
Gary S. Stager, Editor

Director of Advertising Services

Judy Stickney

1997-1998

ISTE BOARD OF DIRECTORS

ISTE Executive Board Members

Lynne Schrum, President *University of Georgia-Athens (GA)*
Dave Brittain, Past President *MGT of America (FL)*
Cheryl Lemke, Secretary *Milken Family Foundation (CA)*
Michael Turzanski, Treasurer *Cisco Systems, Inc. (MA)*
Chip Kimball, At Large *Lake Washington School District (WA)*
Neal Strudler, At Large *University of Nevada-Las Vegas*

ISTE Board Members

Jose Calderoni *ILCE, Mexico*
Penny Ellsworth *Western Springs School District 101 (IL)*
Cameron Gonzales *New Mexico State University*
Cathy Gunn *Northern Arizona University*
Dennis Harper *Olympia School District (WA)*
Paul O'Driscoll *Salem-Keizer Public Schools (OR)*
Jorge Ortega *FACE/Leon County SD (FL)*
Heidi Rogers *University of Idaho*
Carla Schutte *Technology Specialists (FL)*
Peter Wholihan *Sts. Paul & Peter School, Virgin Islands*

ISTE Committees

LaJeanne Thomas *Accreditation and Standards*
Dave Brittain *Awards*
John Ketelhut *Distance Learning*
Michael Turzanski *Finance*
Paul Resta and Gerald Knezek *International*
Jenelle Leonard *Minority Affairs*
Lary Smith *Policies and Procedures*
Gwen Solomon *Publications*

ISTE Executive Officer

David Moursund

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.

Submission of Manuscripts

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

Send surface mail to:

Gary S. Stager
21825 Barbara St.
Torrance, CA 90503 USA

Send electronic mail to:

logoexchange@moon.pepperdine.edu

Deadlines

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

Vol. 17, No. 1 July 15, 1998
"Back to School—Practical Logo Ideas for Teachers"

Vol. 17, No. 2 Sept. 15, 1998
"Perspectives on Papert"

Vol. 17, No. 3 Dec. 15, 1998
"Logo, Math, and Beauty"

Vol. 17, No. 4 Feb. 15, 1999
"The Turtle Parties Like It's 1999"

Logo Exchange is published quarterly by the International Society for Technology in Education (ISTE), 1787 Agate St., Eugene, OR 97403-1923, USA; 800/336-5191.

ISTE members may join SIG/Logo for \$24. Dues include a subscription to *Logo Exchange*. Non ISTE member subscription rate is \$34. 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.

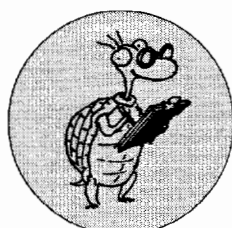
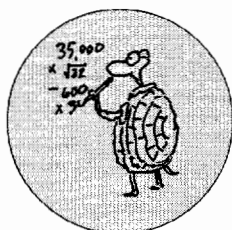
Logo Exchange solicits articles on all topics of interest to Logo-using educators. Submission guidelines can be obtained by contacting the editor. Opinions expressed in this publication are those of the authors and do not necessarily represent or reflect official ISTE policy.

© 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

This publication was produced using Aldus PageMaker®.

Contents



ARTICLES

- Fred and Catie's Excellent Adventure
Fred D'Ignazio and Catie D'Ignazio 5
- Prime Time Math Challenge: A Game of Multiplication and Division
Donald Bourdon 13
- Logo in Russia
Olga A. Tuzova 16
- Turning Manipulatives Inside Out
Gary S. Stager 19
- Making MicroWorlds Music
John Gough 20
- Drawing with Arcs: Part I, Replacing Straight Lines with Arcs
William J. Spezeski 24
- Clever Kids and Computers: The Future Down Under!
Mark Brown and Tracy Riley 26

COLUMNS

EDITORIAL

- Logo and Jazz
Gary S. Stager 2

QUARTERLY QUANTUM

- Make "Seven 5"
Tom Lough 3

LOGO NEWS

- BOOK REVIEW
The Games Children Make
Carolyn Dowling 4

TEACHER FEATURE

- Josie Hopkins
Jeff Richardson 10

STARTING WITH STARLOGO

- Birds
Alan Epstein 11

LOGO: SEARCH AND RESEARCH

- Logo in the Classroom:
The UK Story
Douglas H. Clements and Julie Sarama 18

FOR BEGINNERS

- Putting MicroWorlds
Projects on the Web
Gary S. Stager 30

THE LAST WORD:

- Teach to the Standards;
Not to the Test
Daniel E. Kinnaman 31





Logo and Jazz

Before I dedicated my professional life to teacher education and Logo evangelism, I studied to be a professional jazz musician. Although I no longer practice an instrument or arrange music my CD collection continues to expand and I am continuously inspired by what I learn about learning at the feet of great jazz artists.

Improvisation is at the heart of jazz and the "samba school" discussed in *Mindstorms*. Jazz is an art form you learn and hone in a social context, on the bandstand. There are jazz textbooks, but great musicians are not the product of such books. They are the products of experience. Watching Betty Carter prowl the stage coaxing, cajoling, inspiring and nurturing talent out of young musicians is a special treat. Miles Davis grew as an artist and reinvented the musical genre repeatedly for five decades by collaborating with young musicians. Art Blakey's Jazz Messengers was the MIT of jazz performance for 40 years. Blakey had an enormous influence on generations of musicians by creating a climate of musical excellence to which every young musician aspired. Art Blakey was fond of saying of his band members, "when these guys get too old, I'll get some new ones." Is a second or third generation of Logo leaders being nurtured?

The following is a list of similarities I've identified between Jazz and Logo:

Jazz and Logo value freedom

Wynton Marsalis argues that jazz epitomizes the highest ideals of Ameri-

can-style democracy in that it celebrates individual expression and excellence within a group context for the collective good.

Jazz and Logo are about community

Learning occurs best in a community of practice. Jazz musicians work and learn within a community of other musicians. There is no greater model

Too few music educators learn the philosophy and techniques associated with jazz.

of a learning community than what you may find in the kitchen of New York's Village Vanguard between sets. It is not uncommon to see three generations of musicians trading stories and sharing wisdom with one another.

Both require greater levels of commitment and artistry than their popular cousins

Jazz and Logo are hard fun. Their personal worth derives from being challenging and meaningful.

Much more expressive and personal than popular forms

Math Blaster vs. MicroWorlds. . . John Coltrane vs. Spice Girls. . . 'nuff said!

Easily caricatured and misunderstood

Whether we consider the goateed be-ret-wearing hipster or the elitist Logo

zealot, both groups of people are the subject of misunderstanding and trivialization.

Educators remain ignorant of both jazz and Logo

Too few music educators learn the philosophy and techniques associated with jazz. Too few computer-using educators are exposed to the philosophy and techniques associated with Logo learning.

Their popularity ebbs and flows, but the form continues to evolve

New forms of Logo and jazz continue to emerge. New practitioners are added to both fields.

Jazz and Logo are both under-represented in the media

Philosophy is consistent over time

Jazz and Logo share a clarity and continuum of purpose that is not lost with the advent of new technology. New instruments are embraced and styles assimilated without abandoning the artistic/learning objectives.

An essentially American invention achieves greater levels of respect and popularity overseas

The expatriate is a jazz tradition. For generations, jazz musicians have sought monetary rewards and respect overseas. Logo is alive and well in Latin America, Australia and Eastern Europe.

See LOGO AND JAZZ (Page 8)



Make "Seven 5

Looking at things from different perspectives is always fun for me. At the very least, I get some mental exercise. Often, I discover a delightful path to thinking that otherwise would have remained hidden.

I'm not sure why, but triangles were on my mind the other day. Then, by happenstance, I saw a television program about Pythagoras on the History Channel, showing the elegant relationship between the legs of a right triangle and the hypotenuse.

Later, I went out for some exercise. Leaving home, I jogged four blocks east along the city streets and then three blocks north to the office. I began thinking about the distance I covered—a total of seven blocks. And yet, considering the hypotenuse distance, I was only five blocks from home (the hypotenuse of 5 is the square root of 25, which is the sum of 4 squared plus 3 squared).

When I imagined myself jogging to the office along smaller "blocks," repeatedly going east and then north, I found myself still traveling the equivalent of four original blocks east and three north. This happened no matter how small I imagined the "blocks" to be. The total distance was four plus three or seven original east-north blocks regardless.

So, when does seven become five?

Hmmmmmm. Time for the turtle!

```
to jog :factor
  seth 90 fd 400 / :factor
  seth 0 fd 300 / :factor
end
```

By positioning the turtle in the lower left corner of the screen and running **jog 1**, I generated the legs of a so-called Pythagorean triangle, with one leg of 400 turtle steps and the other of 300 steps, for a total of 700 turtle steps. The hypotenuse of the triangle is 500 turtle steps.

(Note: If your Logo version has a screen width of less than 400 and a height of less than 300, then use proportionally smaller numbers such as 160 and 120 in the jog procedure.)

Then, by returning the turtle to the starting position and typing lines such as

```
repeat 2 [jog 2]
repeat 10 [jog 10]
repeat 100 [jog 100]
```

I was able to study how the path of the turtle began to approach the hypotenuse. When I used larger numbers, such as 1,000, the path of the turtle actually looked like a straight line. But the turtle was still traveling the same 700 turtle steps in going along the hypotenuse, even though Pythagoras showed this to be 500 turtle steps in length. How can this be?

I am still thinking about this question. Part of the answer is probably related to what is called "taking the limit" in calculus. In this mysterious regime of the infinitesimal, does the total east-and-north distance of 700 suddenly become the Pythagorean distance of 500?

For the moment, the actual answer doesn't really matter to me. What does matter is the wonderful process of exploration, of reflection, of musing—all facilitated by the Logo turtle. Both the Pythagorean perspective and the "east-north" perspective have something to suggest about the length of the hypotenuse.

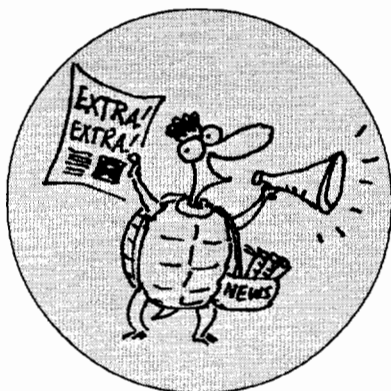
But there is an even larger process, or so it seems to me—the willingness of examining the hypotenuse from a different perspective, and then applying that willingness to other situations.

For example, in education we often use standardized test scores to measure student achievement. But authentic assessment means are gaining in popularity and use. Each of these two perspectives have something to say about student achievement.

Or, perhaps a student suggests a way of doing something completely differently—so differently that, in fact, it seems impossible. Even though it may not be aligned with convention or even possibility, the student's suggestion has something to say about the ultimate outcome. It was from such different perspectives that the Newtons, the Einsteins, the Feynmans obtained inspiration for their conceptual leaps that advanced science so dramatically.

Finally, technology is shrinking the world at an incredible rate, bringing cultures from different countries into contact with each other without the need for physical transportation. What a fantastic opportunity to work to-

See MAKE "SEVEN 5 (Page 8)



Logo News

Mark Your Calendars for Logosium!

June 21, 1998

San Diego NECC

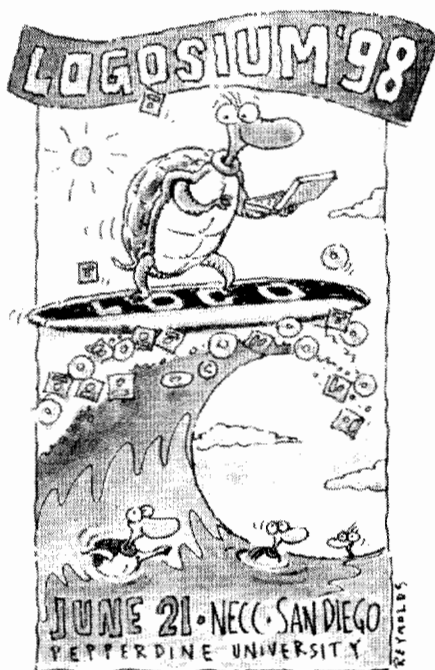
NECC would not be complete without the annual Logosium mini-Logo conference. Logosium '98 will be held June 21, 1998 at Pepperdine University's Irvine, California, campus. Logosium is *the event* for all educators interested in Logo. Beginners and Logo veterans alike will enjoy stimulating discussion, minds-on sessions, hands-on workshops and exciting classroom project ideas at this unique event. Session topics will include:

- Introductions to MicroWorlds
- LEGO and Logo
- Publishing Logo Projects on the Web
- Computer Science Logo Style
- Exploring Mathematics and Logo
- Logo Multimedia Project Ideas

Logo enthusiasts will board a morning bus at NECC headquarters in San Diego and travel to the terrific site to share ideas, collaborate on projects, enjoy hands-on workshops and seek inspiration from Logo veterans. At the end of the day, Logosium participants will be treated to a seaside dinner before returning to San Diego. For more information, check your NECC Advance Program or:

E-mail: logoexchange@moon.pepperdine.edu
 URL: <http://necc98.csusm.edu/necc98workshops.html>

Call for Participation—Logosium
 ISTE's SIGLogo is looking for educators who are willing to share their expertise,



student work or ideas with international colleagues at this year's Logosium. You can lead a hands-on workshop, present research findings, share classroom strategies, join a panel discussion or moderate a brainstorming session. Logosium is about creating a space in which we can all learn together. Please participate. Send a one-page proposal, **by April 30th**, detailing the way(s) in which you can help make this the best Logosium ever to:

Gary Stager
 Logosium '98
 Pepperdine University Graduate
 School of Education and Psychology
 400 Corporate Pointe
 Culver City, CA 90230
logoexchange@moon.pepperdine.edu
 fax: 310/316-7334

Online Sample Issue of Logo Exchange!

The complete Fall 1997 issue of *Logo Exchange* is available online at: <http://moon.pepperdine.edu/~gstager/logoexchange/>. Tell a friend or make a link to it from your Web page!

Jim Muller's New Hands-on Logo Book

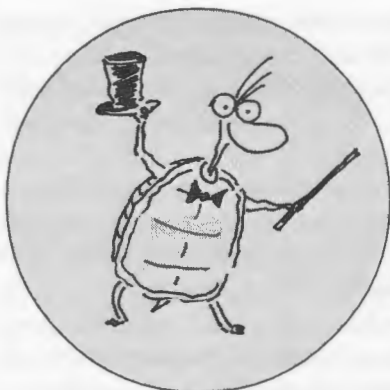
Logo veteran Jim Muller has just published a brand new book, *The Great Logo Adventure—Discovering Logo On and Off the Computer*. The book comes with a CD containing sample files and full versions of MSWLogo and UCB Logo. The activities can be completed in most versions of Logo. To purchase the book, contact:

Doone Publications
 7950 Hwy 72A, #G106
 Madison, AL 35758 USA
 ISBN 0-951934-6-2
 1-800-311-3753
 fax 205-837-0580
 e-mail asmith@doone.com
 web <http://www.doone.com>

New Anthology of Logo Thinking

Logo: A Retrospective is a new anthology of Logo research, thinking and case studies written by an impressive list of American and Australian educators. Authors include Doug Clements, Julie Sarama, Anne McDougall, Gary Stager, Yasmin Kafai and others. The book is available in hardcover and in a softcover version titled *Computers in the Schools—Logo: A Retrospective Volume 14, Num-*

See LOGO NEWS (page 32)



FEATURE ARTICLE

Fred and Catie's Excellent Adventure

by FRED D'IGNAZIO AND CATIE D'IGNAZIO

Editor's note: Multimedia guru Fred D'Ignazio and his daughter, Catie, sent me an animated MicroWorlds project for publication in *Logo Exchange*. Because print journals do not support animation, the text of their fable, complete with screen shots, appears below.

On August 6, 1997, two little turtles set off on a fantastic journey. Fred the Lumbering and Catie the Plodder left their little pond in East Lansing, Michigan, USA, bound for Melbourne, Australia. They carried only what they could strap to their shells (or bribe the airlines to check). This included laptops and tape recorders, magic cables, adapters, microphones, Zip drives, quickcams, and videos. CD-ROMs and phone cords and SCSIs, video cameras, a scanner, and a good deal more. And like good little turtles, they allotted themselves one chocolate bar each to ensure their success.

For me, getting to know a domain of knowledge is much like coming into a new community of people. Sometimes one is initially overwhelmed by a bewildering array of undifferentiated faces. Only gradually do the individuals begin to stand out.

—Seymour Papert, *Mindstorms*

old, creaky structure, rooms with organs, and stray cats. This 200-year-old building seemed like the only shelter available. Catie wanted to jump back in the ocean, but Fred declared that he was plunging ahead, by golly! In the midst of their confusion, they heard a noise. Fred jumped under a bush and Catie prepared to do kung-fu.

It was another turtle!

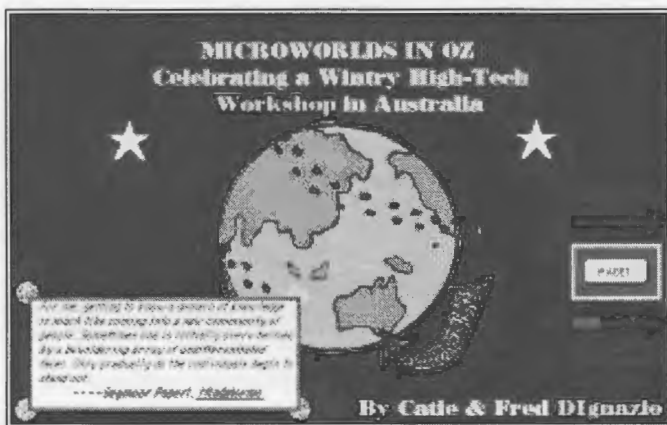


Figure 1

On August 14, 1997, after a long ocean trek involving pirates and smugglers (not wholly relevant to the current story), Fred and Catie spotted land. Avalon! Home of mythical Camelot and the court of mighty King Arthur.

The two turtles splashed and clambered ashore, only to find themselves in the middle of a gothic fairy tale. Wandering up to a nearby sign, they read Avalon Conference Center, Avalon, Australia. The thunder crashed, the lightning flashed, and suddenly, they saw it right in front of them: An

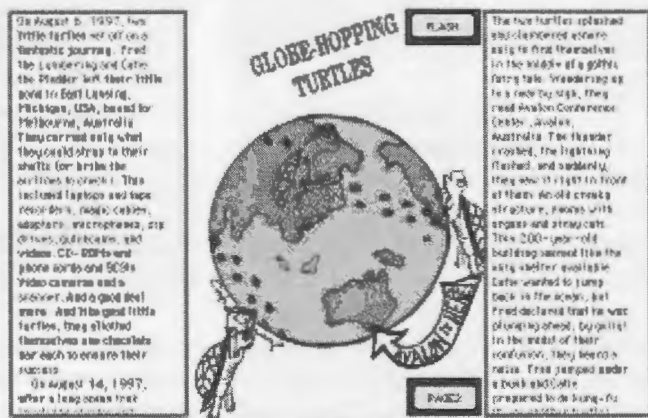


Figure 2

Hey, said Gary the Grumbling, you guys wanna do some MicroWorlds, or what? Fred and Catie breathed a sigh of relief. They had reached their destination for the week: an intensive gothic playground with multimedia for jungle gyms and MicroWorlds for merry-go-rounds. Gary ushered them into the castle-like conference center, which they be-

gan to decorate with cables and computers.

The week was to be major event in Turtle-World, which is, incidentally, the world we all live in. Turtle-teachers carrying little laptops came from all across the Australian continent. They journeyed to Avalon Castle on a pilgrimage to pool knowledge and gain insight into the bewildering world of educational technology.

It was winter in Avalon, miserable, cold and damp. Avalon Castle sat on the edge of Victoria Bay. On one side of the castle lay a vast, flat realm of water; on the other, an endless panorama of wetlands, bogs, and swamps. The castle itself was part of a farm. As they strolled the castle grounds, the turtle-teachers mixed with a noisy gaggle of chickens, horses, sheep, and pigs. Mornings, the teachers leapt out of their beds and raced to the showers. The early birds got hot showers; the slackers shivered in water that was numbingly cold.



Figure 3

To stay warm during the day, the teachers created turtle shells fashioned from layers and layers of sweaters, jackets, pants, and socks. To gain a few precious degrees of body heat, the teachers drank coffee and tea in great quantities. And they built fires—huge, roaring fires—till all the firewood in the castle was squandered. Then teachers hungrily eyed surrounding trees and wooden farm buildings, gauging the waves of cozy heat the wood might give off if it were heaved into the castle's huge fireplaces.

With frostbite and hypothermia a constant concern, the workshop leaders designed a warm-up activity as

soon as the first wave of turtle-teachers arrived. Their plan was home movies! The movies were to:

- Throw the teachers immediately into a hands-on multimedia project.
- Help the teachers reveal their personalities, backgrounds, etc.
- Encourage the teachers to bond into MicroWorlds production crews for the rest of the workshop.
- Keep the teachers warm, discourage frostbite, and stir up the blood.

Fred and Catie brought three silly two-minute biographical movies from the USA that they used as examples for the teachers and as a means to introduce themselves.

The movies worked. They propelled the diverse group of teachers into instant action. Five minutes after the movies were over, the teachers grabbed hand-held camcorders and drove off in cars, chased pigs and chickens across the castle lawn, collapsed corpse-like on the nearby beach, and climbed tall trees, all on frantic missions to shoot wacky biographical videos of their own.

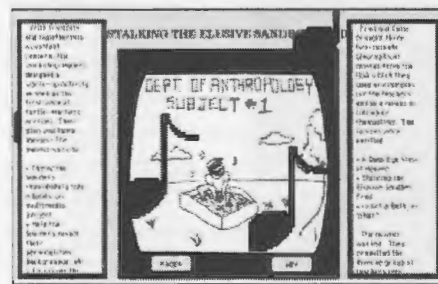


Figure 4

Avalon Castle was large enough to become its own little microworld. Gary, Catie, and Fred called the main room in the castle Mission Control. Here they connected the teacher's laptops to a network. Participants brought files from other parts of the castle, as offerings on Zip drives, to add to the centerpiece of their work, a MicroWorlds 2.0 project constructed by each turtle team. Gary acted as master chef on the MicroWorlds

projects. Fred and Catie assisted teacher multimedia machinations.

Avalon Castle's far-flung rooms became the project studios for the teams multimedia productions. A cavernous living room was converted to a video-capture theater where teachers transformed their movies into Zip files bound for MicroWorlds projects in Mission Control. A satellite prep kitchen became an audio room. At all hours of the day and night, a cornucopia of sounds emanated from this kitchen. Groups of teachers became improvisational rock bands banging on spoons, wailing on electric guitars, plunking musical keyboards, harmonizing, jamming, and crooning. Far into the night, strains of music of all sorts drifted down the hall from the audio kitchen into dormitory rooms, lulling good little turtles to sleep.

In other rooms scattered around Avalon Castle, teachers sat down on floors like high-tech squatters and tapped on keyboards, chatted feverishly, previewed videotapes, listened to boom boxes, scribbled notes, wrote MicroWorlds procedures, and hastily sketched cartoon frames in project storyboards.



Figure 5

Teachers rushing outdoors lugging video cameras on tripods passed other teachers hurrying indoors hauling armloads of worm-eaten firewood. Strong smells of hot chocolate, strong coffee, English tea, and burning logs ebbed and flowed through the castle's interior, all overlaid by the odor of undeniable tension. Every few hours like a town crier, Gary floated through the castle crying, Demo or Die! Demo or Die! And frantic teachers would

dive into action in front of roving bands of other teachers, demonstrating their ragged, unfinished projects as raw works in progress.

Editor's note: "Demo or die" was the original brainchild of *Logo Exchange's* own Jeff Richardson. Kids feel completely comfortable getting up, walking around the room and investigating the work of their peers. A great deal of learning occurs via these interactions. Adults are prone to spend all of their time focusing on their own work while sacrificing the opportunity to learn from others or find solutions to their problems in the work of colleagues. "Demo or die" is an explicit attempt to require workshop participants to share their triumphs, challenges and ideas on a regular basis.

Catie and Fred were overwhelmed by the cold, chaotic frenzy that surrounded them. There were too many skills to be learned by too many teachers in not enough time. What to do? They glanced nervously over their shoulders. At any moment they knew that Gary would come grumbling through, crying Demo or Die!

They had to do something to speed things up. Teachers were simultaneously trying to learn MicroWorlds, LOGO programming, digital video, MIDI music composition, digital photography, research on the Internet, and a smorgasbord of other arcane, high-tech arts. All the raw media gathered by the teachers had to be clipped, cut, copied, and pasted into their MicroWorlds projects. MicroWorlds projects were the glue that held all the media together. And the projects were due in a few short hours.

Suddenly Fred and Catie remembered a quote from Seymour Papert. The quote was perfect. It was about new domains of knowledge. The domains were the hodgepodge of skills these teachers were trying to learn. Papert had called the domains "a be-

wildering array of undifferentiated faces." The only way for everyone to learn these faces in time was to distribute responsibility for learning across the entire group of workshop participants. After the biographical home movies, the teachers momentum to learn had grown so high that perhaps each teacher could learn one or two pieces of the knowledge needed and then share these pieces with other teachers who needed the same pieces. No single guru could be in all rooms in Avalon Castle coaching all teachers. But individual teachers could be one minute gurus who could be called upon, at a moments notice, to share the small but critical skill they had acquired. All teachers could make a contribution. All the teachers together could share knowledge. A knowledge web could be formed. Each node in the web would be a human being.

Fred quickly sketched the major knowledge domains in his workshop notebook. Catie grabbed a stack of poster paper and magic markers and drew up the knowledge domain posters. Teachers pitched in and hung the posters around the Avalon Castle dining room. Catie and Marie the Registrar ran around the castle photographing teachers with a digital camera. Mark the techie-turtle printed out the pictures on a workshop printer. Fred, Catie, and Gary called for a turtle huddle. The teachers assembled, and the three leaders explained the new strategy. Each teacher was asked to:

- Walk around the dining room and study the domains of knowledge which had to be mastered.
- Pick one or two domains they would like to learn personally.
- Learn just enough of that domain to apply it in their teams project.
- Teach what they learned to at least one other team member.
- Paste an autographed copy of their digital photo onto the Domain of Knowledge poster matching their new skill.
- Seek help from teachers whose

pictures were on the domain poster where they needed help.

- Offer help to any teacher who matched their face with the photo on a poster.

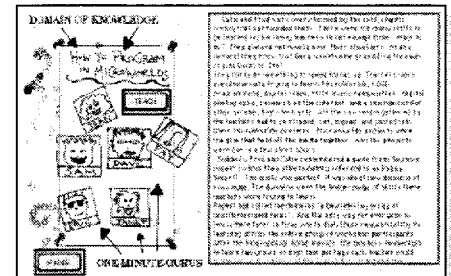


Figure 6

Ready! Set! Go! Gary cried. And the teachers raced off. By lunch time that same day the domains of knowledge posters began to fill with names and pictures of teachers who had published themselves as one-minute gurus of new knowledge. The posters functioned as oversized business cards advertising the teachers as helpers for fellow teachers. Gary, Fred, and Catie flew around Avalon Castle fighting fires, troubleshooting, and offering special-purpose advice. But all the rest of the problem-solving, teaching, coaching, and learning that day subtly shifted to the shoulders of the teachers.

**“LOOKING FOR LOGO
IN ALL THE
WRONG PLACES”**

Logo in Urban Schools

Growing Up with Logo

**Logo at the
Science Center**

**Elementary Logo
Energy Projects**

... and many surprises

Next Issue

At the end of the workshop the posters were magically full of teachers names and pictures. And the projects were miraculously completed.

Gary, Catie, and Fred turtle looked around Avalon Castle. What they saw was good.

They followed the teachers into Mission Control. The lights dimmed. Everyone relaxed in a theater-like hush and watched the final demonstration of the teachers MicroWorlds projects on the big screen. Catie, Gary, and Fred shared a common secret, a secret they shared with every teacher in the workshop: the workshop's success or failure had not rested on the shoulders (or shells) of the three leaders. It rested on the shells (or shoulders) of every participant. Because every teacher made a contribution, the teams were a success and the workshop was a success.

The demonstrations ended. The screen darkened. Mission Control's lights came on.

The workshop was over. In exhaustion and in a final gasp of energy the teachers rushed out the castle doors to their cars. Off they drove to their Australian houses, suburbs, villages, and cities.

Freedom. Free at last. Goodbye sheep. Goodbye pigs. Goodbye cold, damp swamps.

Fred and Catie were led like tiny baby turtles to cars that would whisk them away into the night. They dozed



Figure 7

as teachers drove them through the dark to their faraway accommodations.

Long live Avalon! Long live MicroWorlds! Long live Australian teachers!

(Catie and Fred would like to thank the amazing Gary Stager for inviting them to co-teach the two MicroWorlds workshops in Avalon. They would also like to thank the workshop sponsors: Margot, Bruce, Mark, Marie, and the other talented and caring people at Computelec Pty of Melbourne, Australia.)

About the author

Fred D'Ignazio is considered the father of child-created multimedia and is the father of Catie D'Ignazio. He is a contributing editor to *Learning and Leading with Technology*, a prolific author and a popular workshop leader. Fred is president of Multimedia Classrooms.

Fred D'Ignazio
dignazio@mail.msen.com

LOGO AND JAZZ / Continued from Page 2

From simple elements complexity may emerge

Primitives and procedures are the building blocks of Logo. Chords and scales are the building blocks of jazz. These simple elements are arranged to create infinite levels of complexity, function and beauty.

Plan on joining your fellow "Logo artists" at the next Samba School, Logosium '98 at NECC in San Diego on June 21st, 1998. The sense of Logo community will be enhanced by a bus-ride to the conference site and a sea-side dinner on the way home. Check your NECC program for more information or send email to logoexchange@moon.pepperdine.edu. We would love for you to come share your ideas and experiences with us!

Keep Swingin'

Gary S. Stager

Gary

Gary Stager
logoexchange@moon.pepperdine.edu

MAKE "SEVEN 5 / Continued from Page 3

gether with international counterparts to identify and then to explore the different perspectives on education issues in general, or on Logo issues in particular!!

4 + 3 = 5? Indeed, how could this be so? Nowadays, I try to stop, listen, and think a little before offering my fixed opinion quite so quickly.

FD SQRT (80 * 80 + 60 * 60)
Or maybe FD 80 LT 90 FD 60

Tom Lough, Founding Editor
tom.lough@coe.murraystate.edu
Murray State University Dept.
of ESE, PO Box 9, Murray, KY
42071





BOOK REVIEW

The Games Children Make

by **CAROLYN DOWLING**

Minds in Play: Computer Game Design as a Context for Children's Learning
Yasmin B. Kafai

**1995, Lawrence Erlbaum Associates
(Hillsdale, New Jersey)**

**ISBN: 0-8058-1512-0; 0-8058-1513-9
(paperback)**

This scholarly but very accessible book presents an account, well supported by theory and discussion, of a project in which a group of fourth-year students spend a significant portion of their class time programming games through which third-grade students are to learn about fractions.

At one level it provides a blueprint for teachers and researchers interested in implementing and studying classroom activities that are based on constructionist principles and are of special relevance to students. For many students, when all is said and done, "computers" means "games," and attempts by well-meaning adults to convince them otherwise meet with little success. The harnessing of the deep expertise of young people in this area to an activity in which they are generating knowledge for others not only engages them in the creation of their own problems and the design of their solutions, but also calls into play the time-honored process by which teachers are believed to actually learn more about the content they impart than do conventional learners.

At a deeper level, theoretical links to the work of a number of authors and researchers, many of whom would be

quite familiar to readers of this journal, provide a dimension that not only illuminates the project itself, but also present an insightful and fresh perspective upon well known theories and practices. In particular, the work of Idit Harel, in her Instructional Software Design Project, is acknowledged as central to the subject matter of this book. In addition to suggesting a theoretical paradigm and a particular form of classroom practice, the comparison of these students to those studied by Harel is an integral part of the comparative evaluation process described in Chapter 5. This is only one example of the rich web of association that currently exists between the work of this "school" of researchers, incorporating Papert, Turkle, Harel and Kafai among numerous others, many of whom are acknowledged in this work. While such a degree of intertextuality is of course not uncommon in research communities, it is unusual and most pleasing for such a comprehensive body of interrelated research to be made available in a format and style that is accessible to the public at large.

While it might be tempting for the more casual reader to focus on the central chapters that recount in narrative form three individual and in some respects contrasting case studies, the earlier chapters and subsequent discussion, while less of a "good read," will amply repay careful attention. This is, of course, wholly in keeping with a recognition of the importance of con-

text in learning—including our own!

The book is an important contribution to the body of research and practice concerning the theory and implementation of constructionist principles. In a broader sense it relates in a quite compelling way to a wide range of issues pertaining to how we learn and live.

Since this research was undertaken, the Internet has become vastly more accessible to young people and, it might be argued, has for some students supplanted games as the prime *raison d'être* of computing. While the Internet is itself a fertile source for games, it offers a number of other options for creative activity. The application of some of the principles propounded in this book to activities such as the publication of personal "home pages" is an interesting avenue of conjecture.

From the special perspective of *Logo Exchange* readers, Kafai's project and commentary together are ample evidence of the degree to which Logo has "moved with the times," continuing to provide, in its different and still evolving manifestations, a medium through which children can bend the power of the computer to their own purposes.

LX

Carolyn Dowling
Australian Catholic University
412 Mt Alexander Rd
Ascot Vale, Victoria 3032
AUSTRALIA
c.dowling@mercy.acu.edu.au



Josie Hopkins

by JEFF RICHARDSON

Josie Hopkins would not still be a teacher if it wasn't for Logo. Josie currently works at the pioneering "Logo & Laptops" school, Methodist Ladies College, in Melbourne, Australia, where she is a physics and general science teacher. But not so long ago, she was close to leaving school teaching completely to devote her time and energy to Olympic coaching. Josie is a lifelong swimmer, representing her country at World Championships, European Opens, USA Opens, PanPacs, and the Seoul Olympics. When she finished her career as a competitor, Josie took up coaching, giving particular attention to developing the most recent addition to the Olympic Pool, Synchronized Swimming. Teaching in schools (Josie majored in both physical education and physics) was something she did around her dedication to her sport.

Josie's encounter with Logo came along at just the right time for her as a teacher. She was becoming increasingly aware of the dissonance between the creative, impassioned and focused intensity of her work as a coach with her charges in the pool, and what was achievable with a high school physics class. Josie found herself working at John Paul College (JPC) in Queensland, Australia, another "Logo & Laptops" school.



*"All the kids in the class had laptops with Logo Writer, I was *determined* to use it. JPC had not only a laptop for every child, but a logo culture where difference was encouraged. I found that Logo matched what I was doing as a coach, and what I'd always *wanted* to do as a physics teacher. Logo was a revelation of the kinship between physical and cognitive coaching, except now I saw clearly that kinesthetic creativity is relatively easy, cognitive creativity is *hard*. Teaching, really *teaching* physics is much more challenging than physical coaching, but Logo gave me a holistic, choreographic medium. It changed my view of computing and of teaching. If I hadn't discovered Logo I would have left". (Josie Hopkins)*

Well, Josie didn't leave. She applied her Olympian nature to teaching. Her classes at JPC made dynamic models of the solar system, complete with epicyclic orbits; Mendelian Genetics simulators; and a wrote and performed a version of Shakespeare's "Hamlet" as a dramatic embodiment of Quantum Mechanics. Since moving to Melbourne and MLC, Josie's students have used MicroWorlds to simulate Cellular Ecology, demonstrate Kinetic Theory, and adapt the "Tamagotchi" idea to simulate the germination and growth of plants from seed under

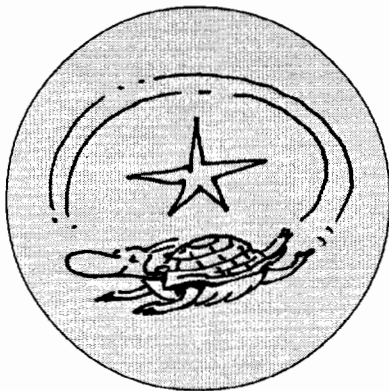
varying conditions. Josie's work at MLC also includes developing World Wide Web implementations of the curriculum to allow students to participate flexibly, from home or school camp or wherever. The class is no longer confined to the classroom. **LX**

About the author

Jeff Richardson, *Logo Exchange* International Editor, is a Senior Lecturer in Education at Monash University and on the faculty of the Royal Melbourne Institute of Technology, both in Victoria, Australia. Screenshots of programs created by Jeff's five-year-old hacker son, John, appear in Volume 16/Issue 2.

Jeff Richardson
jeff@rmit.edu.au

Josie Hopkins
hopkinjl@mlckew.edu.au



Birds

StarLogo is well suited to explore systems in which the individual players have only local knowledge and behavior, but whose overall effect is much greater in scope. Sometimes systems look like they are controlled by a leader when in fact they are not. For example, many people assume that birds flock by following a leader bird, when in fact this aggregate behavior results from the birds deciding to fly alongside only one or two others.

Each bird has acquired a simple set of rules to follow when it wants to go flying:

- 1) fly around for a while, looking for another bird (of compatible breed),
- 2) when you find a flying partner, alter speed and direction to conform with the new partner,
- 3) after finding a partner, flap a few extra times if necessary to avoid colliding, and to set up some spacing.

We will make a simulation of bird flocking behavior using StarLogo. One of our ancillary goals is to introduce and explain some commands that are unique to StarLogo and parallel programming. In this description, I will use the terms "turtles" and "birds" interchangeably, because although StarLogo recognizes them as turtles, we will also interpret them as birds.

The key elements of this program will be a setup procedure, a movement procedure and a procedure to alter course. Remember that all of the birds will follow the same single set of procedures.

The setup procedure will simply clear all, create a number of birds, and call a procedure to set them up:

```
to setup
  ca
  crt numbirds
  config-birds
end
```

The value for *numbirds* in the create-turtles command will be derived from a slider so that it can be changed before each run without having to recompile the program each time. To make a slider, select the interface window, then select the slider icon from the palette and place it somewhere in the Interface Window. Be sure to use the name *numbirds* as the Observer Variable in the *crt* command above, and give it an arbitrary range, say 2 to 100.

The procedure for configuring the birds will take care of randomly placing them in the Graphics Window, and giving them an arbitrary heading and velocity:

```
to config-birds
  setxy random screen-size
  random screen-size
  seth random 360
  setspeed (1 + (random 20)) / 10
end
```

The *setxy* command uses the random number generator with the screen width and height to distribute the birds around the window. We want to be sure that these birds don't have any headstart organizing themselves at the outset. Their headings are also ran-

domly set, and their speeds randomly range from .1 to 2, which is the distance each bird will move in one clock tick. I'd say this group is sufficiently unorganized at this point!

Since we are using the turtle variable, *speed*, in this procedure, we must be sure to declare that *speed* is owned by all turtles. We do this by adding this line at the very top of our program:

```
turtles-own [speed]
```

For cosmetic reasons, let's replace our customary turtles with bird shapes. StarLogo comes complete with a palette of customizable turtle shapes, one of which conveniently looks like a seagull. You can view and change any of the turtle pictures by clicking the right-most icon in the Interface palette. For simplicity, we will use the seagull, shape #3. Extend the *config-birds* procedure:

```
to config-birds
  setxy random screen-size
  random screen-size
  seth random 360
  setspeed (1 + (random 20)) / 10
  setshape 3
end
```

Remember that every bird runs this procedure, but each will end up with a unique position, speed and heading.

Create a Setup button in order to test these procedures. This is also a good time to save your work.

Now for the flight lessons. We will create a procedure called *fly* that will execute over and over. Whenever it runs, each bird moves forward according to its speed, and moves a little farther if it

finds itself on top of another bird. However, before moving, each bird looks for the closest bird it can see and, if that bird is close enough, alters its course and speed a little. In our program altering course and speed merely involves averaging these values from the two birds and flying the average. What this means is that when a bird encounters another, it does not just fly to conform to the other's flight pattern. Instead it goes half way, and assumes that the other bird will do the same. [Consider also the possibility that three or more birds might be very close to each other; this can produce some interesting interactions.]

Finding the closest bird involves a command that rapidly scans all the other birds and returns the turtle ID of the closest one it finds. The *who-min-of-turtles* command is used with the *distance* command to find our target, closest bird:

```
setclosest who-min-of-turtles
[distance xcor-of myself
ycor-of myself]
```

Clearly this looks very different from standard Logo!

The command *setclosest* sets the the variable *closest* to whatever value the rest of the line produces. Since each bird will have its own closest bird, each bird must have its own copy of this variable in which to store the closest ID. This variable must appear in our program as one that all turtles own, similar to the way we set up *speed*:

```
turtles-own [closest]
```

The *who-min-of-turtles* command uses the bracketted command to help it make comparisons among all the turtles it investigates. We know that each other turtle is going to execute this statement in its turn. However, when, for example, it's turtle #3's turn, *who-min-of-turtles* causes *every other* turtle to execute the bracketted command, and keeps track of the one whose result is the smallest. The ID of the turtle returning the smallest ends up stored in the variable *closest*.

The bracketted command can contain any turtle commands that report something. In this case the command checks

for distance from the original, executing turtle. But how does *distance* work?

The *distance* command reports how far away a given X & Y coordinate is from the executing turtle. For instance, a turtle sitting at home (0, 0) executes this:

```
setdist distance 10 16
```

The turtle variable *dist* would contain the direct distance of 18.868, which is the hypotenuse calculated from triangle sides 10 and 16. To find out the distance to the position of another turtle, whose ID is stored in the variable *friend*, you could use:

```
setdist distance xcor-of
friend ycor-of friend
```

xcor-of is a convenient way of reporting the X coordinate of some other turtle, in this case *friend*.

Getting back to our original statement, we find one last trick: the use of the word *myself*. Since the bracketted statement is going to be executed by each of the other turtles that are being checked for distance, the use of *myself* is used as the ID of the calling turtle (in our example, turtle #3).

Here is the *fly* procedure so far:

```
turtles-own [closest]
to fly
setclosest who-min-of-turtles
[distance xcor-of myself
ycor-of myself]
fd speed
if turtles-here > 1 [fd 1]
end
```

The command, *turtles-here*, reports how many turtles there are on this patch. Obviously if there are more than one, we would have to move a bit further away to avoid collision.

The last piece of the *fly* procedure is a statement that does something useful with the knowledge of who the *closest* turtle is. As each turtle executes the *fly* procedure, what we will do is this: check whether it is pretty close, say within 4 units, and if it is, alter our course according to our averaging plan. We use a small number like 4 to avoid having the turtles affected by a turtle all the way across the screen. You could even think

about this as the "vision thing"—in our simulation, birds are very nearsighted.

```
turtles-own [closest]
to fly
setclosest who-min-of-turtles
[distance xcor-of myself
ycor-of myself]
if (distance xcor-of closest
ycor-of closest) < 4
[alter-course closest]
fd speed
if turtles-here > 1 [fd 1]
end
```

The final procedure, *alter-course*, takes an argument, *pal*, which is the ID of the other, very close, turtle. [We could have also used the variable *closest* in *alter-course* instead of an argument.]

As you can see, each bird averages the heading and speed of itself and its *pal*.

```
to alter-course :pal
seth (heading + heading-of
:pal) / 2
setspeed (speed + speed-of
:pal) / 2
end
```

With *alter-course* finished, your birds are free to fly. Starting is easier if you create a *fly* button. Remember to set it to check the *forever* box. Also, now is an excellent time to save your program.

Some people will find that their birds fly a little too fast. Try to add a slowdown statement to your program. Hint: the *wait* command will be useful with an argument less than 1.

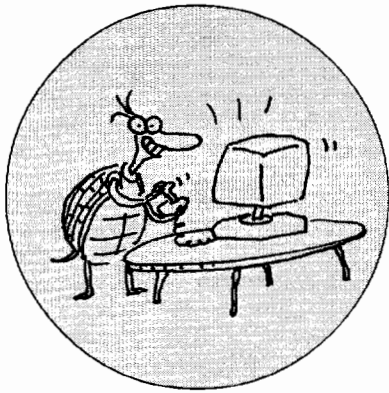
If you're really feeling energetic, try adding a second type of bird, and set it up so that the different types refuse to fly with each other.

Happy flying!

About the Author

Alan Epstein has been developing software since 1975 and most recently has been studying ways that computer technology can be effectively used in schools. He is most interested in robots, especially for children.

Alan Epstein
logoexchange@moon.pepperdine.edu



Prime Time Math Challenge

A Game of Multiplication and Division

by **DONALD BOURDON**

Students like to play games. Mix in some friendly peer competition and you get a winning combination for motivating student learning. Prime Time Math is a competitive two-player math game that makes practicing multiplication and division fun.

Prime Time Math is a logo implementation of the Juniper Green math game featured in the March 1997 issue of *Scientific American*. The program is written in Power Math Logo by New Directions Computer Math.

Program design overview

The main program subprocedures are contained in the superprocedure **START**. First, the subprocedure **GETNAMES** prompts the players to enter their names in the Command Center. **INTRO** gives the players a choice of reading the instructions or beginning a new game.

Game housekeeping is performed by the **SETUP** procedure. I used Charischak's (1996) Hundred Board procedure to draw a 10 x 10 game grid. The **NUMROW** procedure labels the game numbers 1 through 100 onto the grid squares. Finally, **PLAYLOG** prints the players names on the left side of the graphics window.

The first game number, **:NUM**, is entered in **FIRSTPLAY**. The procedure **CHECK** insures that **:NUM** is even to conform to the game rules. If not even, **BADMOVE** prompts the player to reenter an even number. Once an even number is entered, **FILLSQ** positions the turtle to stamp a red square over the number.

In **FILLSQ**, the x-axis position control is accomplished by:

```
setx (-110 + 20 * last :num)
if endzero = 0 [setx 90]
```

The y-axis position control is accomplished by:

```
sety (90 - 20 * first :num)
if :num < 10 [sety 90]
if endzero = 0 [sety 110 - 20
  * first :num]
if :num = 100 [sety -90]
```

The **SETX** and **SETY** commands work fine for all numbers except one through nine and the numbers ending in zero. For these numbers, the **IF** statements move the turtle to the correct xy position to stamp over the number.

The procedure **NEXTPLAY** prompts the second player for the next number, **:NEXTNUM**. **CHECK1** tests **:NEXTNUM** to see if it is an exact multiple while **CHECK2** tests it to see if it is an exact divisor of **:NUM** entered in **FIRSTPLAY**. Both checks are done with the **REMAINDER** command in:

```
to checkmul
output remainder :num :nextnum
end
```

```
to checkdiv
output remainder :nextnum :num
end
```

since exact multiples or divisors leave no remainders in the above operations.

The procedures, **BADMOVE1** and **BADMOVE2**, prompt the player to reenter an exact multiple or divisor, respectively, for any number failing either **CHECK1** or **CHECK2**. Once the player's number passes the check procedure

:NUM is replaced by **:NEXTNUM** and the number is stamped on the game grid. The number is then printed under the player's name on the left side of the graphics window.

PLAYLOGA and **PLAYLOGB** position the text cursor, print the player's number and toggle the counter, **:P**, between one and two to position the text cursor to print the number under the appropriate player's name.

For the third and subsequent plays, **NEXTPLAY** recursively calls itself until a player cannot pick a multiple or divisor. At this point, the players end the game by clicking on the "stop sign" at the right of the Command Center. Players can begin a new game by typing **START** once again in the Command Center.

A logo primitive work around

Power Math Logo lacks a **READNUMCC** command. Since **READLISTCC** returns a list and not a word, a logo number is a special form of a logo word, I needed a way for players to enter multi-character numbers in the Command Center. I used Sharon Yoder's (1994) simple **READWORDCC** procedure to solve the problem:

```
to readwordcc
output first readlistcc
end
```

The complete Prime Time Math program can be found at the end of this article.

Playing the game

The rules for playing Prime Time Math are few and straightforward. The first

player picks an even number between 1 and 100. The next player must then pick an exact multiple or divisor of the first number. Players alternate picking multiples or divisors until one player cannot pick an exact multiple or divisor. The player who cannot pick a multiple or divisor loses. Once selected a number cannot be reused.

Game play begins by typing **START** in the Command Center. Except for the first play which must be an even number, players alternate picking numbers that are exact multiples or divisors of their opponents number. Players try to pick numbers which preclude their opponents from making a play.

Game strategies may take time for students to develop. This is fine since they get plenty of multiplication and division practice trying to refine their game winning tactics. For a detailed discussion on the subtle aspects of game's strategy see Stewart (1997).

Conversion tips

This program should be compatible with most versions of LogoWriter. With slight modifications for clearing graphics or text as required, Prime Time Math should work in other versions of logo as well.

Power Math Logo allows you to design your own shapes. I used a solid square shape to fit over exactly one grid square and defined it as shape number one. If your version of logo does not have a **STAMP** command you will need to replace:

```
setshape 1
setcolor 5
pu stamp pd
```

in **FILLSQ** with your logo version's equivalent of the **STAMP** or **FILL** command.

If your version of logo has a **READNUMCC** command, you can eliminate the **READWORDCC** procedure.

Program enhancements

Readers with a version of logo that supports importing graphics and sound, an eye-catching "splash screen" and or

musical introduction would help to "grab" the players attention.

There is no record kept of the numbers used during the game. Players could "beat the game" by selecting a previously played number. Letting the computer track the numbers played to prevent their reuse would be another useful addition to the game.

At the end of game, when no exact multiple or divisor number can be selected, players must click on the "stop sign" to quit the game. It would be a nice feature to have the program track :NUM and the subsequent possible plays. Where none exist, the program could so state, announce the winner and the end of the game, and prompt to either quit or play again.

In conclusion

Even lacking the above refinements, the students at St. Bernard's School seem to like the game just fine. Most are still working on refining their game tactics. Using logo to practice multiplication and division will be a prime fun time for your students too.

The Prime Time Math computer program

```
to start
getnames
setup
firstplay
nextplay
end

to GetNames
ct ;clear text
cg ;clear graphics
cc ;clear command center
setbg 10 ;sets background color
to blue
print [Who will make the
first play?]
make "playerb first readlistcc
print[]
print [Who is your
challenger?]
make "playera first readlistcc
end

to setup
cc
cg ct
setc 1 ;sets text color to black
setbg 10 ;sets background color
to blue
intro
```

```
grid
numrow 1 0 0
playlog
end

to intro
repeat 4 [print []]
print [If you need instructions
on how to play, press Y.]
print []
print [Otherwise, press P to
start the game.]
if readchar = "y [instructions]
repeat 4 [print []]
print [Let the game begin!]
wait 120
cc
end

to instructions
ct
setbg 17 ;sets background color
to yellow
print [Player A begins by picking
an even number between 1-100,
typing the number in the command
center and pressing RETURN.]
print []
print [Player B then types in an
even multiple or divisor of that
number.]
print []
print [Player A must now type in
a multiple or divisor of Player
B's number.]
print []
print [Play continues until a
player cannot select a multiple
or divisor.]
print []
print [The player who cannot
select an exact multiple or
divisor is the loser.]
print []
print [As numbers are selected
they are filled in on the number
grid.]
print []
print [Once used, a number cannot
be returned to the grid or used
again.]
print []
print [To end the game or if a
player cannot make a move, press
the STOP button.]
print[]
print [Press P to begin.]
if readchar <> "P [instructions]
;waits for a "P" to be typed
cc
end

to grid
ht
ct
setbg 10
setc 1
```

```

pu
setpos [-100 -100]
pd
repeat 4 [fd 200 rt 90]
repeat 10 [fd 200 bk 200 rt 90 fd
20 lt 90]
pu
setpos [-100 -100]
pd
rt 90
repeat 10 [fd 200 bk 200 lt 90 fd
20 rt 90]
pu
setpos [-100 -100]
pd
lt 90
pu
setpos [-100 80]
pd
rt 90
end

to numrow :n :step :c
if :step > 90 [stop]
number :n :step :c
pu
bk 200
rt 90
fd 20
lt 90
pd
numrow :n :step+10 0
end

to number :n :step :c
if :c > 9 [stop]
label :n + :step
fd 20
number :n + 1 :step :c+1
end

to playlog
print [Play Log]
print []
print :playerb
cu ;moves the text cursor up a
line
eol ;moves text cursor to the end
of the line
repeat 3 [print char 32 cu eol]
;prints three spaces
print :playera
print []
end

to firstplay
cc
type [Chose an even number:]
type char 32
make "num readwordcc
if not equal? check 0 [badmove]
fillsq
playloga
end

to readwordcc
output first readlistcc
end

```

```

to check
output remainder :num 2
end

to badmove
cc
type [That number was not even!]
wait 120
firstplay
end

to fillsq
pu
setx -110 + 20 * last :num
if endzero = 0 [setx 90]
sety 90 - 20 * first :num
if :num < 10 [sety 90]
if endzero = 0 [sety 110 - 20 *
first :num]
if :num = 100 [sety -90]
setsh 1 ;sets turtle shape to a
solid square
setc 5 ;sets turtle color to red
pd
stamp ;stamps the turtle shape
onto the game grid
pu
end

to endzero
output last :num
end

to nextplay
cc
type [Chose an exact multiple or
divisor of ]
type char 32
type :num
type [:]
type char 32
make "nextnum readwordcc
ifelse (:nextnum > :num) [check1]
[check2]
make "num :nextnum
fillsq
ifelse :p = 1 [playloga]
[playlogb]
nextplay
end

to check1
if checkmul > 0 [badmove1]
end

to check2
if checkdiv > 0 [badmove2]
end

to badmove1
cc
type [This number is not an exact
multiple.]
show " ;generates a blank line in
the command center
type [Pick another number.]
wait 120

```

```

cc
nextplay
end

to badmove2
cc
type [This number is not an exact
divisor.]
show "
type [Pick another number.]
wait 120
cc
nextplay
end

to playloga
print :num
make "p 2
end

to playlogb
cu
eol
repeat 4 [print char 32 cu eol]
print :num
make "p 1
end

to checkdiv
output remainder :num :nextnum
end

to checkmul
output remainder :nextnum :num
end

```

LX

References

- Charischak, I. (1996). Hundred Board Patterns. *Logo Exchange*, 14 (3), 7-10.
- Stewart, I. (1997). Mathematical Recreations. *Scientific American*, 276 (3), 118-120.
- Yoder, Sharon (1994). *Introduction to Programming in Logo Using LogoWriter*. Third Edition. Eugene, Oregon: ISTE.

About the author

Donald J. Bourdon is a K-5 computer teacher at St. Bernard's School, 32 River St., Saranac Lake, NY 12983. His distinguished career includes: Lt. Col., US Air Force, Retired, 1993; Assistant Professor of Defense Analysis, Naval War College, Newport, RI; Adjunct Faculty, Management Department, Salve Regina University, Newport, RI.

Donald J. Bourdon
dbourdon@northnet.org



Logo in Russia

by **OLGA A. TUZOVA**

Edited by Jeff Richardson & Gary Stager

In Russia there are many interesting and gifted teachers working differently, moving in their own ways. Each of them has many things to share with us. Just in St. Petersburg one can find Lego-Logo, Theatre, Psychology, Ecology, Landscape design and Video (Evgeny Patarakin, <http://dll.botik.ru/~pat>), the creation of animated GIFs based on Logo drawings, solving serious logical puzzles using logical variables and operations, introduction to dynamic programming (Sofia Gortitskaya), and solving linguistic problems (Irina Kuznetsova). Colleagues work on beautiful graphical projects: Logo-ballet "Swan Lake" (Tatyana Gurina), "A Walk With Pushkin Along The Nevsky Prospect" (Irina Dribinskaya). Or, making multimedia Math and Science text-books (Galina Touzova), the creation and exploitation of Logo-databases (Alexandra Yudina).

On the other hand, Russian educators only recently have discovered Logo environments as a powerful tool for discovery, constructive learning and a vehicle of self expression.

So, I invite you to a Round Table discussion, where participants are limited in time and, here on the paper,—in place, and give you just a glimpse into what is happening in most of the St. Petersburg's schools which are using Logo. Welcome to our Round Table!

Organizer: Dear colleagues, the goal of our meeting is to share the experiences we've had in teaching Logo. Since the computer science and information technology entered our classrooms we have been offered a great variety of software, developed especially for school teaching, including software for learning about computer programming. Why do we choose to use Logo? What goals do we pursue? Have we come up to expectations? What problems have we encountered in our practice and how do we overcome them? I'd like these questions to be the stem of our discussion. Who would like to get us started with their comments?

First teacher: As it is in many schools, I begin to teach Logo in the 5th grade and my choice is LogoWriter. First of all, being a programming language, Logo develops such important human experiences as analysis and synthesis, planning and experimentation, composing of the algorithm and its description in a formal language. Besides, the process of debugging fosters confidence in students. They learn to not fear making mistakes, just to bravely find and revise them, and this is crucial to their growth. Most importantly, the LogoWriter environment conceptually is very clear, children feel comfortable there and although commands both for turtle and for cursor

are very simple and few, serious and really creative projects can be done using this software. Second, I use Logo as a special tool that helps kids in their cognitive activity. The simple fact that they can not only look at something, hear or read about something, but reproduce this, make a model of the object or event on their own, is most effective. Thus, working on the project, devoted to St. Petersburg's crude iron fences, children inevitably are being involved in their own investigation of rhythm and repetition in ornaments, symmetry and proportion. What is most important, these investigations are not forced on them by a teacher, but inspired by their desire to make their own model, to make it beautiful.

Second teacher: Supporting this point of view, I'd like to add that these activities would be more effective if specialists in different knowledge domains were involved. Therefore, I work in close contact with the math teacher and we have developed a special collaborative curriculum which helps children greatly in their efforts to master many geometrical concepts, such as angles, different kinds of polygons, coordinates, circles and so on. Before my students learn to write programs on their own, I introduce them to these concepts via games, which have been developed by my senior students who

also use the Logo environment. For example, let's take the game Target. In this game, the child has to estimate the direction (angle) and the distance from his turtle, which is placed randomly on the screen, to the target and send the turtle directly into the center of it. Or, I'm sure a child will have fewer problems with the coordinates after he learns to play the game where he has to convert 10 monsters (that is, 10 turtles wearing masks of monsters) into 10 beauties after correctly pointing to the screen coordinates of each turtle. We also make special investigations about the properties of circles and polygons, about the number "pi"...

Third teacher (Grumbler): I think you are very lucky to have an opportunity to establish good contacts with your colleagues. Classes where I work, as most of yours, are divided into two parts for computer science lessons. These two parts have their lessons simultaneously in two different computer labs ruled by two different teachers. Forget about the math teacher, I can't find common language with my closest colleague!

Organizer: It's surely not a typical situation! Though, I should confess, we intended for many teachers of different specialities would use the Logo environment for their purposes, that children would do special projects for Math and Science, Biology and Russian . . . but unfortunately it hasn't happened and it couldn't have happened without special mechanisms involving teachers in using new technologies. I think we all feel this "effect of scissors" — the more a teacher is experienced in his subject, the less he wants to change something in his practice. It's a real problem.

Fourth teacher: I'd like to comment on the division of a class into two parts for computer science lessons. I'm sure more or less serious computer programming need not be forced on kids who are not interested in it and the fact is, in conventional class there are many of them. The division of the class helps a student to find his place according to his own interests as we have different

curriculum for these two groups of students. For one group we offer a special course of advanced programming based on MSW Logo. And the other course is offered to those, whose brains "are made differently." I've got very good results with this part as students work in LogoMiry, the Russian version of MicroWorlds. The students effectively use the multimedia opportunity of this environment making wonderful animated comics, multi-paged magazines, TV news or simple games with button control. It's a real joy to see how some of them who have been more used to learning failures now come to light through this work.

Russian educators only recently have discovered Logo environments as a powerful tool for discovery, constructive learning and a vehicle of self expression.

Grumbler: But this work with a built-in painting program and just changing shapes has nothing to do with Logo or at least with real programming! And I have no answer for a student who can't understand why he should write something like REPEAT 4[FD 100 RT 90] instead of taking an appropriate tool in a menu of painting program and just drawing the square, which is much more understandable for him.

Fifth teacher: The choice of Logo environment depends on the purposes you pursue and on the abilities and age of the students. And the effectiveness of the work is defined in a major part by the teacher's skill in formulating tasks and how students are directed and encouraged, isn't it? In our school we try to introduce students to as many possibilities as possible, and we work with LogoMiry, MSW Logo and Lego Dacta Control Lab. At the end of the school year, students have to perform a project on their own in any of these

environments. The project may be for individuals, but it also may be done by a group of students, which is preferable. Students are offered the list of about 20 themes, followed by some explanation as to what is expected as a result of each project, but children are free to develop their own proposals and to coordinate them with the teacher. They are encouraged to use Internet resources and different modern technologies and tools during their work and for project presentation, such as HTML editors, PowerPoint and word processor from Microsoft Office, Adobe PhotoShop for processing images after scanning, and others. In two months of intense work, a great school presentation of Lego-Logo projects takes place and it's really worth seeing! There you can see computer games, graphical and fractal patterns, education programs for younger students, animated GIFs—based on Logo patterns, computer tests and quiz programs, computer controlled Lego models, and many other very different things including WWW pages. Every group of students present their works by themselves. The main part of this is that you hardly could meet there a student who is not proud of his work. Though, for teachers this kind of work with students is of course is a most challenging one.

Organizer: Thank you all very much. Though we have heard many interesting and useful things in our brief discussion, I'm sure we are putting not a full stop here, but just a comma. **LX**

Acknowledgement

The author wishes to thank Donald A. Paulson, who made this article readable in English.

About the Author

Olga Tuzova, Ph.D., is a computer science teacher at the International School of General Education, St. Petersburg, Russia.

Olga Tuzova

olgatu@ort.spb.ru

http://www.ort.spb.rutuzova_e.htm



Logo in the Classroom: The UK Story

by **DOUGLAS H. CLEMENTS AND JULIE SARAMA**

Our last few columns discussed Logo innovation in the classroom: How and why does it happen... or not? Our colleagues in the UK also conducted research on these questions.

Insights Gained Trying to Realize the Logo Dream

Celia Hoyles (1992) describes four processes that occur as Logo moves into classrooms: Pedagogizing, compartmentalizing, incorporating, and neutralizing.

Pedagogizing

In looking back at their experiences, Celia and her colleagues noticed the strong influence of their interventions. They tried to follow the students' leads, but the interventions still largely dictated what the students did. They did well in areas in which they intervened, but not so well in other areas.

Also, when any of us look over students' shoulders, it's easy to see the wonderful things going on and to jump to lots of conclusions about all the mathematics they are doing. However, they might really be doing something different. We have to keep asking specifically what is going on inside the students' heads.

The point is, the role of the teacher is crucial. You have to start thinking about teaching, teachers' beliefs and values.

Another irony is that many teachers who do well with computers get promoted. In one of their projects, a follow-up study a year later found that 99% of the teachers who went through

the courses had been promoted out of the classroom!

Why don't other teachers get involved and take their place? They have found several reasons: computer access difficulties or limitations; management difficulties; and the fear of new ideas, change, not knowing more than the students. "We have to accept though that whatever your view of mathematics education, if it is a matter of authentic mathematics education or classroom control, there is absolutely no doubt which must come first!" (p. 15).

So, we have to consider, more fully, the whole issue of teaching.

Compartmentalizing

This is the "adding on" phenomenon we mentioned in previous columns. Computers are often kept separate from the mathematics curriculum... insulated from the "main work" of the classroom. This "allows" schools to "innovate" without upsetting the power balance in the curriculum. Students pay a psychological price: Fractured knowledge is often useless knowledge.

So, we have to sort out Logo-based ideas that are mathematically useful. Then we have to find ways to build bridges between Logo and the curriculum.

Incorporating

Sometimes, Logo is "incorporated" into the curriculum. But again, Logo is changed. Classroom mathematics is linear, atomized, and often trivial. Certainly, Celia noticed how their own in-

terventions changed when they knew the bell was coming in five minutes! Teachers feel a need to bring things together at the end, even if this cuts off mathematical exploration.

Whatever is "officially" curriculum is at risk of becoming "reading the teacher's mind." Some boys had constructed a suite of cars and planes. Celia suggested a park of different kinds of cars of different sizes. The boys just shrunk theirs non-mathematically. "When I have an agenda that is not negotiated with the pupil, the more powerful the medium, the more chance they have to escape from my agenda!" (p. 21)

Similarly, Celia and her colleagues believed they won a major victory when Logo was designated as part of the UK National Curriculum. Until they saw an item on the next National exam: "rewrite SQUARE to have a repeat statement."

Neutralizing

"Oh, well, computers are just tools." Celia hates that common comment. It neutralizes her intentions. "Just a tool" carries a lot of messages, from non-change to a top-down transmission metaphor of learning. These de-professionalize teachers. In a way, "the move to interactive video and CD-ROM, with their singing and dancing software, is the next step in curriculum delivery via computers. You can sit and watch all this happening, the teachers are just

See THE UK STORY (Page 23)



Turning Manipulatives Inside Out

by **GARY S. STAGER**

Dust Off Your Old Math Manipulatives

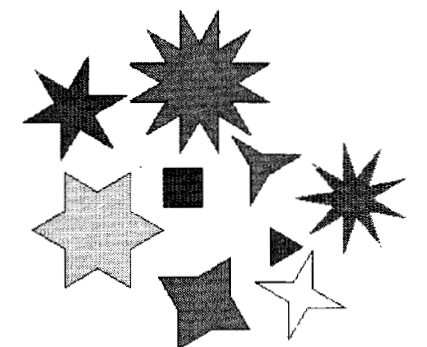
Tactile math manipulatives are a great tool for counting, measuring, representing numerical relationships and exploring geometric patterns. Logo provides an opportunity for manipulatives to be used in new ways. While the following classroom activity appears simple, it can yield quite complex mathematical thinking and problem solving. Any manipulatives may be used including pattern blocks, Cuisenaire Rods, Infix Cubes, tangrams and pentominoes. Manipulatives that tessellate provide some of the richest opportunities for problem solving. I am rather fond of the "Puzzelations," foam-rubber tessellating manipulatives, available from: <http://www.tessellations.com/>. These manipulatives have a finite number of shape pieces which may be combined in infinite ways.

The activity described below is appropriate for 4th grade through adult learners. Past experience and mathematical knowledge may be brought to the problem solving process, regardless of age.

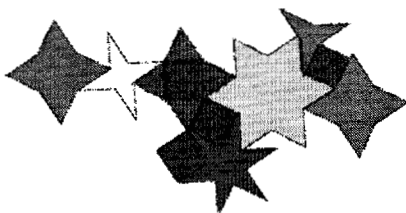
Off-Computer Preparatory Activity

- Each group of 3-4 students should be given a representative set of manipulatives with a few copies of each piece.
- The same type of manipulatives may be given to each student team or different sets may be used, depending on the educational goals of the project.

- Students should spend some time moving the pieces around and exploring their physical relationships.
- Students decide on names for each shape (invented names are OK for strange pieces or with young learners)



Logo drawn examples of pieces from the Puzzelations Tessle-gram set



A tessellation using several of the pieces

Logo-based Activity

Teams of students will design "manipulative" software to be used by others. The user should have a satisfying experience manipulating the shapes on the screen.

Each team of students should pursue the following three goals:

- Create a Logo procedure to draw each shape individually (polygons should all have equal sides)
- Write procedures to combine/connect two or more of the pieces.
- Design an interface for a user to manipulate the shapes on the screen and create interesting patterns. Commands, buttons, sliders and text instructions may be used to assist the user.

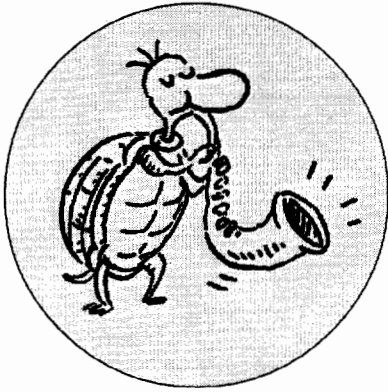
Extension Activities

- Use variables to control the size of each piece.
- Use sliders to control the variable size of each piece or even to zoom in/out of a tessellation.
- Use Find/Replace to replace constant *fd* and *bk* distances with variables, such as: Replace *FD* with *FD Scale **. Be sure to do the same for *BK* and use *:SCALE* if global variables are used instead of sliders.
- Write procedures to randomly tessellate the pieces!

By using language to describe the attributes and measurements of each piece students enhance their ability to understand and communicate geometric relationships to the computer. Issues of variable, measurement, angle and orientation are made concrete. Adults who did well in high school math classes are often hamstrung by their recollections of half-memorized algorithms while their colleagues (or students) use their bodies, common sense, and turtle geometry to solve the same problems.

At the end of the activity, kids can trade software and explore the work of their peers.

LX



Making MicroWorlds Music

by **JOHN GOUGH**

MicroWorlds offers several ways of working with music, one note at a time. In fact, the primitive term that plays a note is, quite simply, **note**, followed by two numbers, one for pitch of the note, one for duration. MicroWorlds has a Melody gadget (accessed through its pull-down menu bar item *Gadgets*), a sound-writing equivalent to the pop-up graphics palette of tools accessible through the paint-button. This lets us click with the mouse on a palette of sound tools that includes a built-in piano keyboard to create a melody with no programming at all. Click on one of the on-screen piano keys and you place a note in the new melody you are making. Hunt and peck, and presto!—you can write a smash-hit Top 40 tune. Fine, unless you happen to believe, as I do, (and I think Seymour Papert did in *Mindstorms*) that programming is the thinking soul of Logo languages. Immediate-mode point-and-click computer work is not the same as either programming or thinking.

It is terrific to be able to control volume and tempo using built-in sliders, to be able to name any melody that you make, and to be able to assign a particular instrument. On a Macintosh the list of possible instruments includes violin, clarinet, lute (actually the standard sound resource, “sosumi,”) and orchestra (the standard resource, “indigo”). On PCs there are six different sound resources named piano, bottle, harp, xylophone, violin, and organ. Other sounds for instruments can be used, especially if you have a plug-in microphone to make new sound resources. But you need to use the primitive command `setinstrument` to access the non-standard instruments or sound resources.

The command `show soundlist` will display all the sound resources available to be used as MicroWorlds instruments on your particular computer. Using the recording capacity of your computer (if you have this), you can even add sound resources, although it is best to keep these short, like a yap, snap, or grunt. It can be quite amusing to hear MicroWorlds playing a tune by singing the words “Uh oh!” at the appropriate pitch, or to have a choir of ducks, or droplets. These cute sound resources work nicely on my Macintosh where they are the standard sound resources “Oops”, “quack,”

and “droplet”. I have also used a microphone to record and add an “arfing” dog sound, just for fun. Incidentally, a sound resource whose name consists of two or more words is not easy to use in MicroWorlds. But it is possible if you use commands that collect together a string of separate words, and space-characters, and treat them all as one long word, for example:

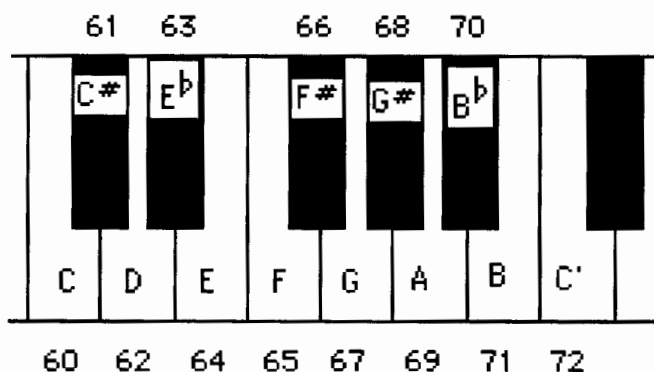
```
setinstrument (word "simple char 32 "beep).
```

You assign the instrument, either by clicking on the appropriate button in the Melody tools palette, or by using the MicroWorlds primitive command `setinstrument` *inputname* or `setinstrument` *input number* (although number inputs only work on a PC). This lets us program the way that any subsequent note commands will be played. You can “orchestrate” our melody by assigning different instruments to different notes in the melody. Try this Macintosh example: You may need to specify other sound resources to suit the resources on your particular computer.

```
to Mary.orchestrated
setinstrument "violin
note 64 3 note 62 1 note 60 2 note 62 2
setinstrument "sosumi
note 64 2 note 64 2 note 64 4
setinstrument "indigo
note 62 2 note 62 2 note 62 4
setinstrument "droplet
note 64 2 note 64 2 note 64 4
end
```

Incidentally, the speed of adjustment from one sound resource to another varies according to the processing speed of your computer. This can be annoying if your computer runs slowly, even though it would not influence the ordinary speed with which a sequence of note commands are played using only one sound resource.

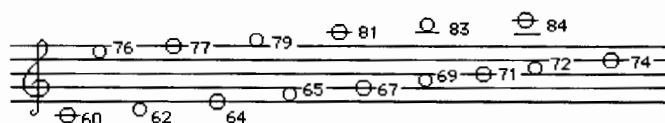
Here is how MicroWorlds really uses the primitive command `note`. The pitch-input can be seen in the diagram of a piano keyboard, showing part of the range, from one to 127.



You start at the musical note called Middle C (near the middle of a piano), whose pitch-input for **note** is 60, and proceed upwards (# or sharp and *b* or flat refers to the black keys on the keyboard) increasing the pitch-input by 1 for each half-step or piano key as you move up the piano's chromatic scale. In table or chart form you have this:

C	C#	D	D#	E	F	F#	G
	D <i>b</i>		E <i>b</i>		F#	G <i>b</i>	
60	61	62	63	64	65	66	67, and so on.

So that you can turn a printed tune into a MicroWorlds' procedure it is also necessary to see how this piano keyboard arrangement of pitch-inputs relates to the ordinary Treble clef in written music. (Bass clef notes are easy to work out from the Treble clef example.)



But it is important to remember the fact that this way of writing music does not show black notes, except by using sharp or flat signs, usually at the left-hand-end where the music starts being written. For example, a sharp sign # on the top line means that all F notes are treated as F sharp, the black note immediately to the right of any F note: so instead of 77 you use pitch-input 78, instead of 65 you use 66, and so on. Similarly a flat sign *b* on the middle line means that all Bs are treated as flattened, so that instead of using the white note to the left of C you use the black note immediately to the left of B: for example, instead of 71 you use 70.

I am glossing over many significant technical music aspects. But these are not essential if you want to start making MicroWorld's music. It is not important that students already know how to read music. The diagrams above can be treated as a kind of technical diagram or even as a map, with contour lines and distinguishing features that need to be interpreted. After that it's just straightforward programming with a logical computer language—the Logo language underlying MicroWorlds.

For example, can you complete a *MicroWorlds procedure* which will play the following tune? I have given the first three note commands—beware of the F sharps.



```
to play.tune
note 72 8
note 67 8
note 71 4
... and so on ...
end
```

The MicroWorlds reference manual and online help describes its music command **note** this way:

note number duration

Plays a note using the current instrument. The first input is the MIDI note number and the second is the duration in tenths of a second. Middle C is 60. The maximum number for note is 127; the maximum duration is 255. Note acts like a small tape recorder playing recorded sounds. Because of this, notes of long duration will play in a short period of time followed by silence. Examples:

```
note 60 2 Plays middle C.
note 64 5 Plays E above middle C, lasting
one-and-a-half times longer.
```

What is this technical reference to MIDI note numbers? MIDI is the standardized computer system that deals with music-related data, such as sounds. It is comparable to the standardized ASCII coding of alphanumeric characters (letters, numbers and symbols) on a computer keyboard (see my earlier article "My Turtle Parles Français" *COM3* vol. 19 no. 1 May 1992 pp 19–20).

What about the comment that notes of long duration will play in a short period of time followed by silence? Try this to hear what it means: **note 72 2**, **note 72 4**, **note 72 8**, **note 72 16**, and **note 72 32**. The only practical solution for the limitation on long durations in the MicroWorlds' command **note** is to avoid using inputs for duration that go much beyond 10. If you take a sixteenth note or semi-quaver as the minimum duration possible, 1, this leads to an eighth note (or quaver) having duration 2, a quarter note (or crotchet) having duration 4, and a half-note (or minim) having duration 8. It is helpful to show



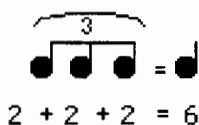
these durations as they appear in written musical form.

You might like to try the following versions of "Mary Had a Little Lamb" or "Merrily we Roll Along", where the durations are controlled easily by a local variable `:d`.

```
to MaryMW :d
note 64 (3 * :d)
note 62 :d
note 60 (2 * :d)
note 62 (2 * :d)
note 64 (2 * :d)
note 64 (2 * :d)
note 64 (4 * :d)
note 62 (2 * :d)
note 62 (2 * :d)
note 62 (4 * :d)
note 64 (2 * :d)
note 64 (2 * :d)
note 64 (4 * :d)
end
```

Try, for example, **MaryMW 1**, **MaryMW 2**, **MaryMW 3**, **MaryMW 5**, **MaryMW 10**, **MaryMW 20**. In MicroWorlds the tune actually becomes more and more disjointed and unmusical as the durations specified by the variable `:d` become longer and longer and there is more silence at the end of a note.

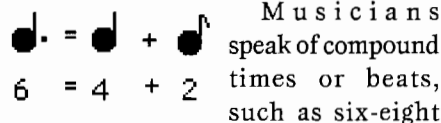
Musical triplets are a little difficult for MicroWorlds. The idea is that where you might have had two notes (for example, eighth notes) that last as long as one other (in this case, a quarter note), you have three notes, each equally long, lasting as long as the single note. This occurs occasionally in music, and can apply to triplets of shortened-quarter notes that last as long as a half-note, or triplets of shortened sixteenth notes that last as long as an eighth note.



For example, the first notes of "Here We Go Round the Mulberry Bush" can be treated as triplets made of shortened quavers, fitting the marching beat given by crotchets in each line of the song.

Triplet-like groups of three notes at a time can also be treated as simple eighth notes, where the standard beat is

not a quarter note but a dotted quarter note. A dot written after a note extends its duration by half of its length again.



time, namely, six notes in a bar of two beats. This is different from simple times such as four-four, with four quarter note beats in each bar. These differences are unimportant in this discussion.

Here is how you teach MicroWorlds to handle triplets. If you let a quarter note be represented by a note command whose duration-input is 12, then an eighth note has a duration of six, and two eighth notes add up to a total duration of 12. A sixteenth note has a duration of three. Fine. Now let's consider three notes in the space of 12—they are triplets. Clearly they must each have an input-duration of four. You will hear that they fit together differently from, say two sixteenth notes, or three sixteenth notes, each lasting three. Musically it makes all the difference. MicroWorlds can just manage this, as long as there are not too many half notes, which would require input-durations of 24 (far too long, too much silence between notes), or too many dotted-quarter notes whose input-duration is half as long again as that for a quarter note, namely, 18—also too long for MicroWorlds.

I should also mention a completely new MicroWorlds primitive command, **rest**, which is used to create a musical rest, that is, a silence which lasts for a specified amount of time. While this can be done by using a note command whose pitch-input is too low to be audible, such as **note 1 d**, or a standard *LogoWriter*-type **wait d** command, the new primitive **rest d** does exactly the same job, where the input `d` is the duration, measured in fractions of a second.

Using MIDI-coded notes, MicroWorlds essentially digitizes the succession of semitones, starting with a low C

of pitch input 1, and working upwards through 10 successive octave sequences (12 semitones at a time), with a C recurring at pitch-input values of 12, 24, 36, and so on, up to 120. If you start at Middle C, with pitch-input 60, then C# is 61, D is 62, D# or E flat is 63, E is 64, F is 65, F# is 66, and so on. Certainly neat. (Incidentally, on my Macintosh, pitch-inputs below about 26 are inaudible! This means that there are only about 100 musical pitches available. With 12 semitones per octave, this means you have about eight effective octaves.)

What if you wanted to use programming as a link to acoustics and the physics of music? MicroWorlds' MIDI-code means that instead of using the Physics rule of doubling a frequency to jump an octave, you create an octave by adding 12 to the pitch-input of two successive note commands (there are twelve semi-tone steps in an octave). The following MicroWorlds procedure plays a randomly selected octave. It should sound like a piano tuner.

```
to random.octave
make "x 40 + random 30
note :x 2
note (:x + 12) 2
end
```

Or you could try this to run through a succession of octaves, for any specified starting point. Try octaves 1, or octaves 5, for example.

```
to octaves :n
repeat 10 [show :n note (:n * 12) 2 make "n :n + 1 cc]
end
```

crotchet or quarter rest	4	minim or half note rest	8
quaver or eighth rest	2	semi-quaver or sixteenth rest	1

You can actually use such stepping through a succession of notes as an entry point or modeling resource in music classes. The following procedures allow us to create a random walk around

a given string of notes. Incidentally, the idea of a random walk is quite different from randomly choosing any item you like from a given list, as you shall see.

The procedure **rand.pick** :input.List picks, at random, one of the items in the specified input list, and hands it over for further work with another procedure. The procedure **walk.pick.12** randomly changes a variable :n so it ranges from a minimum value of 1 to a maximum value of 12, returning to the central value of 6 if it is changed too much at either extreme. (Can you devise a way of remembering the successive values of :n that are generated when you run these procedures? It can be interesting also to plot them as successive points along a number line.)

```
to rand.pick :input.List
output item (1 + random count
:input.List) :input.List
end

to walk.pick.start.12
make "n 6
end

to walk.pick.12
make "m rand.pick [-3 -2 -1 0
1 2 3]
make "n :n + :m
if :n < 1 [make "n 6]
if :n > 12 [make "n 6]
output :n
end

to diatonic :k
walk.pick.start.12
repeat :k [note
item walk.pick.12 [47 48 50
52 53 55 57 59 60 62 64 65 ]
rand.pick [1 2 4 8] ]
end
```

These procedures are written for MicroWorlds. You might notice the way the sixth item Middle C works as a kind of musical home to return to whenever our value of :n has become too large or too small. This basic idea of key is used in discussing tunes and pieces of music.

Can you change this so it walks through a minor key, using these inputs? [51 53 55 56 59 60 62 63 65 67 68 71]

Can you change these so they "walk" around the whole tone scale, sometimes used by Claude Debussy: C D E F# G#

A# C? Or the mystic chord used by Skryabin (also known as Scriabin, depending on the particular way you transliterate his Russian name from the Cyrillic alphabet): C D E F# A B flat? Of course you will need to provide two sets (like octaves) of these two six-note examples.

Compare these random walk tunes with the following procedure that picks any one of 12 notes out of the computer's hat. Composers such as Arnold Schoenberg, Anton Webern and Alban Berg suggested such an idea as an alternative to using the usual western scales. Perhaps surprisingly their music can often sound quite tuneful, although chromatic. The procedure **atonal** :k is a joke machine that shows a fully random alternative to the walk tunes generated by the preceding procedures.

```
to atonal.MW :k
repeat :k [note rand.pick [60
61 62 63 64 65 66 67 68 69
70 71 72]
rand.pick [1 2 4 8] ]
end
```

In this case there is no musical home or sense of original key to which the walking note occasionally returns, and the probability of going to any of the other possible notes is the same while the procedure runs.

You might like to consider how you can use variables to create a list that remembers the sequence of pitch and duration inputs for your random walk tunes. Then you can compose as well as improvise, letting improvisation feed your inspiration!

Within the mentioned limits, MicroWorlds offers us a sound way to enter directly into the turtle-world of programming. This musical connection with another curriculum area is extremely valuable and deserves to be widely exploited by all teachers. **LX**

About the Author

John Gough is a lecturer in Education at Deakin University, Deakin University SDS, 221 Burwood Hwy., Burwood, Victoria 3125, Australia.

John Gough
jugh@deakin.edu.au

there to switch the machines on and off and make encouraging noises." (p. 22).

Logo is an innovative approach to education. Innovation is difficult.

Teaching Teachers

Celia Hoyles' work clearly shows the need for teacher education. Her colleagues, Cândida Moreira and Richard Noss, did just that (Moreira & Noss, 1992). They started with the belief that teachers don't need "treatments" to cure their lack of enthusiasm for innovation. Instead, they need to be offered new ways of thinking about their teaching. Their new ways should fit with their own ideas and beliefs and allow the teachers to play a major role in developing educational alternatives. **LX**

References

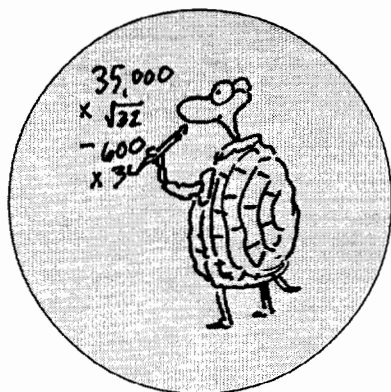
- Hoyles, C. (1992). School interface and curriculum change. In S. Dawson & R. Zazkis (Ed.), *Proceedings of the Six International Conference for Logo and Mathematics Education* (pp. 3-25). Vancouver, B.C., Canada: Simon Fraser University.
- Moreira, C., & Noss, R. (1992). The teacher's view of Logomaths. In S. Dawson & R. Zazkis (Ed.), *Proceedings of the Six International Conference for Logo and Mathematics Education* (pp. 27-50). Vancouver, B.C., Canada: Simon Fraser University.

About the Authors

Time to prepare this material was partially provided by a National Science Foundation Research Grant, "An Investigation of the Development of Elementary Children's Geometric Thinking in Computer and Noncomputer Environments," NSF MDR-8954664. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Doug Clements and Julie Sarama are long-time contributors to *Logo Exchange*.

Doug Clements
clements@acsu.buffalo.edu
Julie Sarama
Sarama@coe.wayne.edu



Drawing with Arcs

Part I: Replacing Straight Lines with Arcs

by **WILLIAM J. SPEZESKI**

The theme of this article is to explore what happens if you replace straight lines with arcs when you draw some common constructs such as polygons, stars and spirals using Logo. Some of the results are very striking. Before we get to the drawing stage, however, we need to make some observations and establish some general guidelines.

Preliminaries

For the purposes of this article, the phrase "draw with an arc" shall mean getting a comparable result with an arc as with a line segment. In one case this refers to connecting two points; in the other, it refers to moving in a specified direction a specific distance. Basically, we would like our arc to begin where a given line segment begins and end where the line segment ends. In going from point A to point B, we have two choices:

We can either draw an arc that moves left of our intended (straight-ahead) direction or one that moves right of our intended direction. Moreover, each such arc, independent of its initial direction, can have one of many curvatures, so there are many paths that can be traversed.

Arc Fundamentals

In Logo, arcs can be drawn with the traditional procedures ARCR and ARCL. For convenience, procedure ARCR is given below:

```
TO ARCR :RADIUS :DEGREES
  LOCAL "STEP LOCAL "REM
```

```
MAKE "STEP 2 * :RADIUS *
3.1416 / 36
MAKE "REM REMAINDER :DEGREES 10
REPEAT :DEGREES / 10 [RT 5 FD
:STEP RT 5]
IF :REM > 0 [FD :STEP * :REM
/ 10 RT :REM]
END
```

Procedure ARCR draws an arc as it traverses a circle in a clockwise direction. Procedure ARCL is similar to ARCR with occurrences of RT being replaced with LT; it draws an arc in a counter-clockwise direction. In both cases, the parameter RADIUS controls the radius of the underlying circle and parameter DEGREES controls the number of degrees in the central angle spanned by the arc.

In the case of replacing line segment AB shown, ARCR will produce a con-

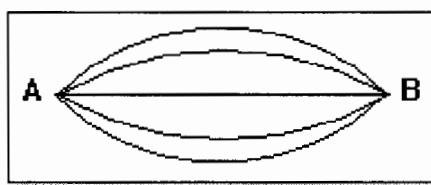


Figure 1

vex arc while ARCL will produce a concave arc. The commands that are needed to replace the line segment AB (produced by FD n) with a convex arc are

```
LT d/2 ARCR r d LT d/2
```

where :RADIUS = r and :DEGREES = d. The two occurrences LT

d/2 adjust the direction of the turtle before and after each arc is drawn. Similarly, the commands that are needed to replace the line segment AB with a concave arc are

```
RT d/2 ARCL r d RT d/2
```

With repeated combinations of these commands, note that the expression

```
REPEAT :K [ LT n/2 ARCR m n
LT n/2 ]
```

can be rewritten as

```
LT n/2
REPEAT :K [ ARCR m n LT n ]
RT n/2
```

The second form is much more efficient. Another point worth noting is that if it is necessary to make the linear distance between the two end points of the arc equal to a specific value L, then assign parameter RADIUS the value $L / (2 * \sin :DEGREES / 2)$. This provides an easy means of controlling the distance between the two end points of an arc, should that be necessary.

Modifying Logo-drawn Objects with Arcs

With the preliminaries now complete, let's proceed to some interesting applications of these ideas. One interesting example is the transformation of a very rigid sailboat to one with billowing sails.

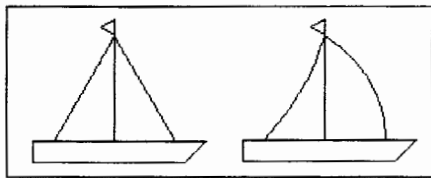


Figure 2

To accomplish this, one FD 80 command in the original drawing procedure was replaced with

```
RT 15 ACRL 80/(2*SIN 15) 30 RT 15
```

and a second FD 80 command was replaced with

```
LT 30 ARCR 80/(2*sin 30) 60 LT 30
```

The possibilities are limitless.

Drawing Polygons with Arcs

Let's try drawing polygons, replacing each side (line) of the polygon with an arc. For each polygon that we consider, though, there are actually two cases. A polygon can be drawn using convex arcs or it can be drawn using concave arcs. For the purposes of this article, all polygons and other constructs will be drawn by moving in a clockwise direction. Relative to each figure then, the procedure ARCR produces convex arcs and the procedure ARCL produces concave arcs.

Convex polygons can be drawn using

```
to convex.poly :n :d :m
lt :d/2
repeat :n [arcr 50 :d lt :d
rt :m*360/:n]
end
```

This procedure is similar to a typical procedure for drawing polygons with the FORWARD command replaced by ARCR and the related turtle direction adjustments. It draws each side of the polygon with a convex arc rather than a straight line. The parameter *n* determines the number of sides. The parameter *d* determines the span (central angle) of the arc in degrees. The procedure produces regular polygons when *m* = 1 and star-shaped polygons when *m* > 1 (and *m* and *n*

are relatively prime). The radius of the arc has been arbitrarily set to 50, but it can be parametrized as well.

Some convex variations of a hexagon (*n* = 6, *m* = 1, *d* = 90, 180, 270) are shown below:

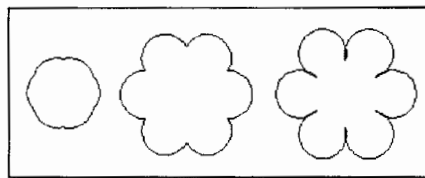


Figure 3

Concave polygons can be drawn using a procedure similar to CONVEX.POLY; ARCR should be replaced with ARCL and both occurrences of LT should be replaced with RT. Some concave variations of a hexagon (*n* = 6, *m* = 1, *d* = 90, 180, 270) are shown below:

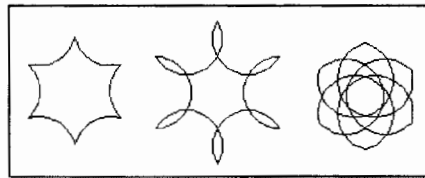


Figure 4

Drawing Stars with Arcs

Some convex variations of a 5-pointed star (*n* = 5, *m* = 2, *d* = 90, 180, 270) appear below:

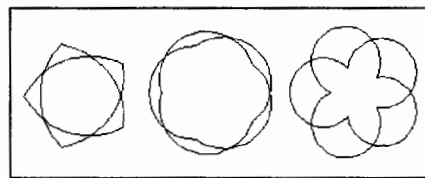


Figure 5

Some concave variations of a 5-pointed star (*n* = 5, *m* = 2, *d* = 90, 180, 270) appear below:

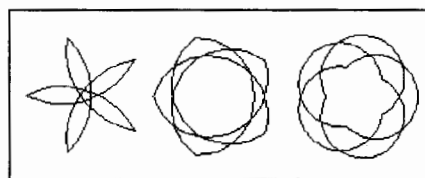


Figure 6

Drawing Spirals with Arcs

Polygonal spirals can also have their lines replaced by arcs to produce some very elegant designs. In each of the pro-

cedures above (to draw either a polygon or star), the figure's basic shape is determined by the parameter *n* representing the number of sides or vertices. In the procedures below that draw spirals, the basic shape of the spiral is controlled by the parameter *ang*, the exterior angle that the turtle turns through to create the spiraling figure.

Convex spirals can be drawn using

```
to convex.spiral :ang
:radius :d
repeat 80 [ arcr :radius :d lt :d
rt :ang make "radius :radius+1]
end
```

Some convex spirals (*ang* = 120, *d* = 90, 180, 270) are shown below:

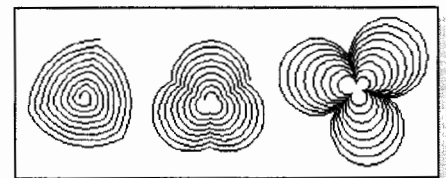


Figure 7

Concave spirals can be drawn using a procedure similar to CONVEX.SPIRAL; ARCR should be replaced with ARCL and LT :d with RT :d. Some concave spirals are shown below. For the first two figures, *ang* = 90; for the third, *ang* = 103. Can you guess the corresponding values of *d*?

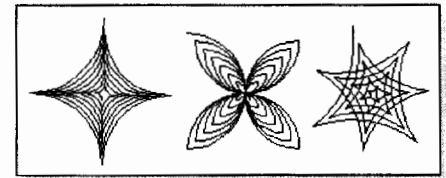


Figure 8

Conclusion

Many interesting things can happen when you begin replacing straight lines in objects with arcs. In Part II, we will explore replacing lines with some arc variations.

About the author

Bill Spezeski teaches at Massachusetts College of Liberal Arts and is a long-time contributor to *Logo Exchange*. Look for other articles by him in future issues.

Bill Spezeski

wspezesk@mcla.mass.edu



Clever Kids and Computers: The Future Down Under!

by **MARK BROWN AND TRACY RILEY**

The day started with 10 designers, seated around a conference table, contemplating serious issues related to the future. As they brainstormed, the two Operation Managers facilitated the discussion and recorded a range of ideas on a mind map.

"Evolution," chirped one.

"Intelligence," piped in another.

"What about the Internet?" one pondered.

"What sorts of historical findings do you think will be unveiled?" questioned a fourth.

"Hey, the world is definitely going to be a different place!" surmised another.

These comments were not, however, from a group of Massey University academics or even university students. Rather the Design Team was made up of 10 young people participating in a one-day workshop, *Computing for Clever Kids*.

The workshop was conceived as a way in which potential talent could be unleashed through technology. In New Zealand, few schools formally recognize the educational needs of gifted children, including those with strengths and interests in computers. At the same time, the lack of literature and research on the use of computers for teaching the gifted (Bulls, 1996) indicates that today's educational technologies are largely untapped by those seeking to carry forth the principles of gifted education. These principles set the stage for innovation, creativity and problem-solving, and individualization in learning. Thus, these were the aims

of *Computing for Clever Kids*.

A number of schools in the geographical area surrounding Massey University were targeted. Teachers were asked to distribute brochures describing the workshop to those children ages nine to 11 who had been identified as having special abilities. The identification process most commonly used in New Zealand schools is teacher observation and nomination (McAlpine, 1996). This is the manner in which the 10 participating children were identified.

The characteristics displayed by the children who participated clearly matched those supported by the literature and included such traits as task-commitment, above average ability, and creativity (Renzulli & Reis, 1985). Another notable characteristic was interest and ability in the area of computing. This type of specific academic ability is characterized by "persistence... a preference for independent learning... and pursuit of the subject outside school time" (Moltzen, 1996, p. 51).

The workshop was held all day on a Saturday utilizing a computer suite within the University's facilities. The suite is equipped with 12 Macintosh Power PC computers which are all capable of supporting the latest educational software. Each child had his or her own computer terminal from which to work, thus allowing for individualization and encouraging autonomy. Again, the goals of the workshop were to foster innovative ideas and provide opportunities for creativ-

ity and problem solving through independent and group learning experiences. The intent of this article is to provide readers with an overview of the workshop, serving as a model for further exploration of the ways in which Logo like experiences can be used to develop and enhance the special abilities of gifted kids.

The Morning Orientation

At the start of the day, the participants were transformed from children to microworld designers. The adult facilitators were no longer teachers, but Operations Managers. As a part of the *Massey University Microworld Design Team*, each designer made an application, serving as a means of both introduction and orientation. The children were asked to describe their qualifications, experiences, and special interests, as well as their reasons for being a part of the team.

Some of their qualifications included experience working on a team, good computer skills, and expertise in problem solving. Fun, experience, and the desire to learn more were the reasons indicated for wanting to be a part of the team. Their intrinsic motivation and interest shone through, with one student stating "I have always wanted to know how to program modern computers." This particular child had previously read a book on programming even though he did not have access to a computer.

Following the completion of the applications, the members introduced them-

selves stating what strength they brought to the team. In this simulated environment the children and teachers were forced from a passive to an active role, as described by Marks (1992) who further stated "they are thinking, discussing, researching, and making decisions" (p. 26). Furthermore, the simulated design team changed the learning environment from a "teacher-student information exchange into a problem solving situation" (p. 26). This was a most desirable outcome for the teachers who clearly saw themselves as facilitators.

The teaching team consisted of a specialist in gifted education and a specialist in educational technology. The opportunity to teach in a partnership which relied upon each individual's strengths and interests simply reinforced what we know about teaching gifted children. Gallagher and Gallagher (1994) advocate for this approach stating that a team can challenge students as no one teacher can. The teaching partnership between a content-area specialist and gifted educator may be one way in which technology can be further used with gifted kids.

Why Use MicroWorlds?

The software chosen for the workshop was *MicroWorlds Project Builder™* (Logo Computer Systems Inc, 1993). MicroWorlds is a multimedia version of Logo. It supports the traditional features of Logo but provides children with many additional tools with which to think. In this regard, the workshop was informed by the original philosophical and psychological foundations of Logo. The intention was to create a rich learning culture where in teaching the computer how to think, children embark on an explanation about how they themselves think (Papert, 1980).

With many software packages there is a tendency to provide so much sophistication that the task is not cognitively demanding—it is simply a case of pointing and clicking the mouse. This does not happen with Logo. There is a high level of challenge and the software requires a way of

thinking about problems that is not available in regular classrooms.

On the other hand, a major criticism of Logo is that the process of manipulating an imaginary turtle around the computer screen is not an authentic problem solving activity. Experiences in Logo do not always transfer to other problem solving situations (Lockard, Abrams & Many, 1997). The potential in this multimedia version of the software to create an engaging and meaningful context for the development and transfer of higher order thinking skills was recognized. The partnership of sound instructional strategies for gifted children and the additional features of *MicroWorlds Project Builder™* was considered to overcome many of the concerns about Logo. There was every indication that, if used in the right way, MicroWorlds was ideally suited to the education of gifted children.

Starting with Baby Talk

The Operations Managers started by introducing the kids to the features of the programming language itself. The children gained hands-on experience with the language by manipulating their own turtle around the screen. The designer team progressed beyond this 'baby talk' through experimentation with more sophisticated commands and learning how to write procedures. Beginning with quite simple procedures, such as using the 'repeat' command to create a 'square', children very quickly were placing procedures within procedures to design a variety of shapes. An interesting outcome at this early stage was how children were learning from their mistakes! These mistakes were often the trigger for a new idea and even more creative design. In just a short period the children had moved from the simple to the complex.

The kids had become articulate in their use of the language and it was clear that they were ready for a more demanding challenge.

Putting On Your Thinking Caps

Having been oriented to MicroWorlds and given some opportunity to experiment, it was time to move on to the real purpose behind the design team. Using the theme *future*, the children discussed and debated three generalizations. Following the model suggested by Riley (1997), these generalizations, once finalized, served as the central focus of the remainder of the day's work. Much discussion took place as students used their prior knowledge and experiences to develop the following:

- The future will be different.
- The future will be both predictable and unpredictable.
- The future will be what you make it.

Keeping these generalizations in mind, the students brainstormed the many topics under the umbrella of *future*. Some of their ideas are illustrated in Figure 1. During the brainstorming fluency, flexibility, elaboration, and originality, the elements of creativity described by Torrance (1966), were evident.

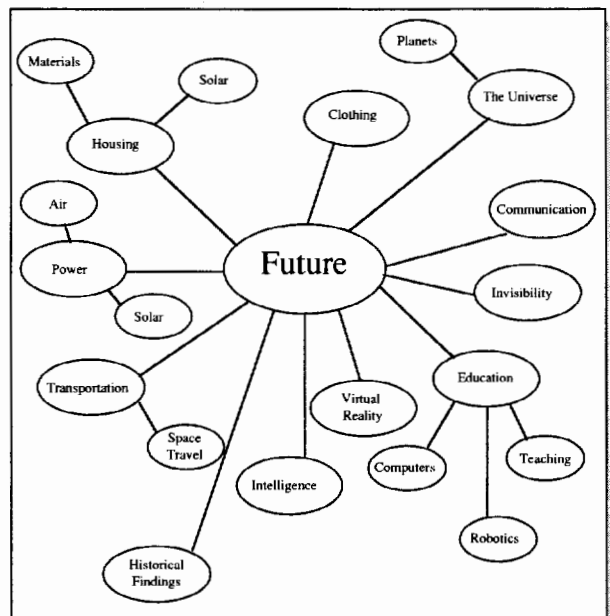


Figure 1

The use of conceptual themes, such as the future, increases the complexity of the content being offered gifted children. As students begin to examine the interrelationships between and among facts, details, and rules, they also see the inter-relatedness of areas of study (Riley, 1997). Within the Web and discussion generated by the children participating in the workshop, these connections became quite obvious. Furthermore, the use of conceptual themes lends itself to the characteristics of clever children.

Following the brainstorming session students were given the problem: to design an original Microworld using the future theme. The students were also encouraged to depict within their microworld the three generalizations which had been established by the team. Each child then completed a contract of agreement which gave their proposed future topic and a brief description of their main ideas related to that topic. Some of the topics selected included intelligent life in space, transportation, cities, bicycles, cybernetics, invisibility, and computers.

Upon signing the agreement each child was also appointed as a consultant for another child. As consultants the children were expected to "demonstrate positive communication, constructive criticism, and respect to other team members." The contract was also signed by the Operations Managers who pledged to "recognize and delight in the creative genius and problem solver of the designer." This component of the simulation was fun, but also gave the children some direction for the remainder of the day.

Getting Underway

Following the signing of contractual agreements the designers went to work, with hourly appointments with their consultants. This idea of independent work coupled with team work is similar to that espoused by Melton (1985). In writing workshops teams of authors, editors, and art directors work together to create an original book.

Melton states that altering the dynamics of the group is the key to success. Moreover, the purpose in having children serve in an advisory role is to improve or enhance the final product. Indeed this was the case in *Computing for Clever Kids*.

The consultants shared ideas with the designers and visa-versa, but at the same time something different began to happen in the computer suite. As progress was made toward the design of original MicroWorlds, "experts" began to surface. One student discovered the microphone and how to record messages, while another ventured into a clip art program transporting the art to his microworld. In another case a child asked for the manual to experiment with some advanced features of the language. The Operations Managers suddenly did not have adequate knowledge or expertise in how to create these MicroWorlds. . . but the children did! The dynamics of the group were completely transformed.

One of the keys to this transformation was the role of the teachers. As stated previously, the teachers served as facilitators of learning, particularly at this stage in the workshop. Davis and Rimm (1994) list several characteristics of teachers of the gifted which were emulated by the teachers in this program, including the ability to "create a warm, safe, accepting, and democratic environment" and "ease in new and unknown settings" (p. 41). In addition to this, the teachers intentionally went about the business of sharing with students. For example, the educational technology specialist stated at one point "If you want to know how to put a button in your microworld, see Michael. He's our expert on buttons!" By being explicit in this sharing of ideas and doing so in an encouraging manner, an atmosphere of cooperation and team work was further enhanced.

In addition to the atmospheric changes, the students' skills were also quickly developing. Not only were their computing skills beginning to

sharpen, but their skills of communication, collaboration, self-management, and problem solving were being put to the test. It would also seem fair to say that the motivational levels of the children were quite high during this component of the workshop. All of the children were highly engaged and on task in the design of their microworld.

Designing activities which are motivational for gifted children is an art in itself. However, factors such as perceived fun in, and control over, learning may influence motivation (Middleton, Littlefield, & Lehrer, 1992). Several questions regarding academic fun and student motivation were affirmed by this workshop. These questions are:

- "Does the activity engage an interest?"
- Is the activity sufficiently challenging (stimulating)? and
- What control of the learning process can be given to the students?" (Middleton, et al., 1992, p. 43)

These questions are further supported by Troxclair, Stephens, Bennett, and Karnes (1996) in relation to a multimedia course for gifted students. In planning workshops for gifted students, these questions should be kept at the forefront.

Show and Tell

It was difficult to get the children to stop as there seemed to always be just "one more thing" they wanted to include in their Microworld. The Operations Managers were successful, nonetheless, in tearing the designers away so that they could share the range of future simulations. Although Figure 2 gives a sample of one child's final product, it does not show the true extent of inter-activity between objects nor the complexity of the programming that underscores the work. The MicroWorlds were indeed impressive given that none of the children had prior experience with Logo.



Figure 2

During the final discussion, focus was directed upon the ways in which the student-created generalizations about the concept *future* had been depicted in the MicroWorlds. Each generalization was addressed, with students giving examples from their own MicroWorlds. The connections between seemingly different areas of interest were now seen in a concrete manner. For example, based upon the generalization "the future will be different" a variety of scenarios were given. One Microworld depicted new forms of transportation which included flying bicycles, while another showed futuristic buildings. Still another consisted of a number of objects which seemingly disappeared and later reappeared, including trees! The students unanimously agreed that regardless of the content - transportation, buildings, the environment - the future would be different. By reflecting upon these generalizations, the workshop was wrapped up in a purposeful and meaningful way.

Final Evaluation

The children were asked to formally evaluate the workshop in terms of each of the day's activities and on their own performance, as well as that of the Op-

erations Managers. Feedback was very positive as demonstrated by the fact that all of the children would recommend the workshop to a friend. The children also reported on the skills they had developed from the MicroWorlds experience. Results of the written evaluations confirm observations that this was a rewarding and successful experience for all concerned.

Self-report data on the development of Logo skills, based on categories proposed by Nolan and Ryba (1986), indicate that the children perceived themselves to be competent programmers. Table 1 shows a high level of creativity and a good range of reported coding, exploration and debugging skills.

Furthermore, the children reported on the development of what was deemed their professional skills. These were general skills relating to both computer and non-computer activities. Data were consistent with previous observations and confirm a high level of reported communication, self-management and problem solving skills.

At the end of the workshop parents and other adults were given an overview of the day's activities and then invited to explore their child's micro-world. This last part of the day took longer than anticipated because children were keen to show off their new found skills. Moreover, the kids were extremely proud of their creations. It was encouraging to see so many children and adults celebrating together in their learning achievements. The children relished the opportunity to teach the adults! This helped the adults to appreciate the level of challenge and overall complexity of the *future* simulations. It was apparent that both adults and children alike were highly impressed and satisfied with the MicroWorlds workshop.

What Next?

Computing for Clever Kids is just one avenue that can be taken in the provision of adequate educational technology experiences for gifted children. At Massey University it is the first step. Future workshops are being developed

See CLEVER KIDS (Page 32)

How students perceive their Logo skills

	%	%	%	%	%
	Poor	OK	Good	Very Good	Great
Coding	0	10	10	40	40
Exploration	0	0	20	40	40
Prediction	0	20	40	30	10
Creativity	0	0	40	10	50
Analysis and Planning	0	0	60	10	30
Debugging	0	0	30	40	30

Table 1



Putting MicroWorlds Projects on the Web

by **GARY S. STAGER**

The World Wide Web provides a universal vehicle for publishing student work. LCSi, the makers of MicroWorlds, have added functionality to MicroWorlds 2.X allowing users to share their Logo projects via the Web. While the process of Web publishing a MicroWorlds project is pretty simple, Mac and Windows 95 users need to be aware of dramatically different results. Hopefully, LCSi will provide Macintosh users with the same power available to their Windows 95-using colleagues.

What Does MicroWorlds Offer the WWW?

Kids should always have the opportunity to publish their work and the web offers an enormous potential audience for their creativity. MicroWorlds allows users to create truly interactive Web pages without the need to learn Java or program complicated CGIs. The Windows 95 plug-in allows MicroWorlds users to put their video games, science simulations, animated reports and mathematical explorations online. Neither Web-authoring packages nor HyperStudio can experiment with a number theory problem while changing variables, collecting data and creating graphs from any web browser (see <http://moon.pepperdine.edu/~gstager/3n.html>).

What Can You Do on a Mac?

1. **SAVEHTML.** MicroWorlds for the Macintosh users can use the **SAVEHTML** command to save an open project for publication on the Web. If you do not have the **SAVEHTML** command in your

version, you may download it free of charge from <http://www.lcsi.ca/>. **SAVEHTML** saves your MicroWorlds project as a series of JPEG files and an HTML file containing the page layout information and any text. While animations, melodies and multimedia elements will not work on the Web, any turtles or buttons designed to turn pages will. This means that Mac users will be able to use MicroWorlds to create projects such as colorful reports and stories. Use a Web authoring tool, like Claris Home Page, to open the HTML file created by MicroWorlds and add links to your pages.

2. **Apple Internet Address Detectors.** Mac users (running OS 8 on PowerMacs) have at least one advantage over Windows users because of the new free Apple Internet Address Detectors. This system extension allows Web URL links to be embedded in MicroWorlds projects (and any other application). After you install the Address Detectors just select text containing a URL or email address and Control-Click on the text. A menu will pop-up asking you if you would like to go to a link in the browser of your choice, bookmark the link, join a news-group or send e-mail to the address selected. Now your MicroWorlds projects can active Hypertext links to the World Wide Web! Download the Apple Internet Address Detectors at

http://applescript.apple.com/data_detectors/detectors.01.html.

What Can Win 95 Users Do?

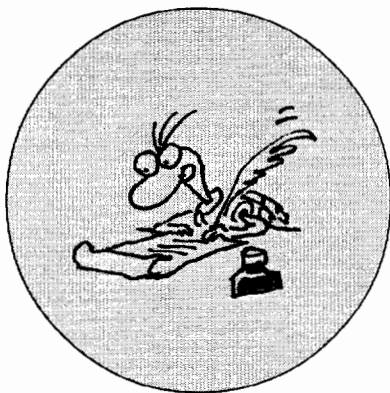
1. **SAVEHTML.** This is built into the Windows 95 version of MicroWorlds. Users can do the same things in Windows 95 as with the Mac.
2. **The Super Cool Amazingly Powerful MicroWorlds Web Plug-in.** Go to <http://www.lcsi.ca/> to download the Web MicroWorlds plug-in and follow the instructions for putting an interactive MicroWorlds project in a Web page.

To create a terrific Web-ready project follow the following guidelines:

Use **NEWPROJECTSIZE** to set your projectsize to 400X400 pixels or smaller. You want the project to fit within your browser window. Do not expect video, CD, or long audio clips to play on the Web. Melodies, MIDI sounds and short recorded audio clips will work. With no command center for typing commands, use buttons and turtles to execute procedures and **QUESTION** to request information from the user. Sliders work too.

Follow the simple instructions found at <http://www.lcsi.ca/web-player/tips.html> for setting up your MicroWorlds-energized Web page.

Be sure to let us know about your new MicroWorlds-based Web pages and we'll share them in future issues! **LX**



Teach to the Standards; Not to the Test

by **DANIEL E. KINNAMAN**

I once asked math education guru Judah Schwartz to tell me in one sentence what was wrong with the standard K-8 math curriculum. His response: *It's immoral to spend 200 hours to teach a student to be a poor imitation of a five-dollar calculator.* Sometime later, I repeated this admonition against blindly over-emphasizing computation skills as opposed to mathematical understanding while leading a LEGO-Logo workshop, and got the following retort from one of the participants: *Fine, but first they'll have to stop testing my students to be poor imitations of five dollar calculators.*

The teacher who said this is a good teacher, but in this instance, she fell victim to the misunderstanding that she should teach to the test and to specific test question formats rather than to the standards underlying them. In the discussion inspired by her comment, we used the Connecticut State Mastery Test as a case in point. (Connecticut is where this workshop took place.)

In Connecticut, students in Grades 4, 8, and 10 take a State Department of Education developed criterion-referenced mastery test widely recognized as one of the nation's best. Our discussion focused on one of the objectives on the fourth grade math test dealing with estimation. A sample question from this area of the test presented students with a picture of a paper clip with a long dark line under the paper clip. Test takers were asked to estimate the number of paper clips laid end to end needed to equal the length of the line. The common "teach to the test" response from seeing such a sample question, is to present students

with many similar questions for practice. This response, however, focuses more on helping students to develop familiarity with the test question format than with helping them to develop an understanding of the why, when, and how of estimation (which first requires that teachers understand the theory underlying the standard).

Having just played with LEGO-Logo, I shared how another teacher and I had used these materials to take an alternative approach to helping a group of students learn estimation. Our initial discussions with these students revealed that they had little understanding of the concept of estimation, and they did not consciously use it in their daily lives. Together, we built a couple of LEGO-Logo robots that could move forward and backward, and rotate left and right. My colleague and I wrote a couple of procedures to enable the robots to accept rotate commands in degrees (rather than time increments), and then together with the students we created a series of estimation games. We spent the better part of three weeks on this project and combined the LEGO-Logo activities with lots of explicit language about estimation, and using group and individual discussions to connect the classroom activities with real life situations in which estimation skills were required and used.

When we then showed the students problems like the paper clip test question, not only could they answer it correctly, they could engage in an intellectual discussion about the merits of various types of problems and question formats to test someone's understanding of estimation.

In a recent keynote address at the annual conference of the American Association of Education Service Agencies (AAESA), Linda Darling-Hammond reminded me of these experiences as she emphasized the importance of teachers and administrators knowing not just the standard, but the theory behind it, so that the activities through which students learn and apply the standard are contextualized in ways that are appropriate and effective (i.e., provide utility to the learner beyond the classroom or test). Teaching to the standards is much harder than teaching to the test. But it's worth it if we value student achievement. As Darling-Hammond pointed out in her keynote, student achievement is a function not just of student ability, but of teacher ability and student effort. Good teachers, she said, get students to try harder, and students who try harder, achieve more. It is toward that end, and in that spirit, that we should use Logo. **LX**

About the Author

Daniel E. Kinnaman, a contributing editor to *Logo Exchange*, has more than 15 years' experience as a classroom teacher and district-wide director of education technology. He also has taught graduate teacher education courses in conjunction with colleges and universities nationwide. He is executive editor of *Curriculum Administrator* magazine and serves as a consultant to Compaq Computer Corp., Safari Technologies, and school districts and education service centers throughout the United States.

Daniel Kinnaman
kinnaman@uconnvm.uconn.edu

bers 1/2/ 1997 ISSN: 0738-0569. For more information, contact Haworth Press, Inc. <http://www.haworth.com>.

A Computer in a LEGO Brick

The folks at LEGO recently announced the forthcoming availability of the long-awaited programmable brick developed by the Epistemology and Learning Group at the MIT Media Lab. The new line of products is being called *LEGO Mindstorms*, although the product won't come with a version of Logo. Visit <http://www.legomindstorms.com>.

Constructionist Books Online

A complete list of books reviewed this year by *Logo Exchange* will be maintained at <http://moon.pepperdine.edu/~gstager/logoexchangebooks.html>. You can even purchase them from this Web site.

Logo Foundation Summer Workshops

The 1998 Logo Summer Institutes take some time this summer to focus on your own learning.

Summer at Spence

New York City, June 29 - July 2

Located in the heart of New York City just steps from Manhattan's Museum Mile, the Spence School, an independent school for girls, is a leader in educational technology. Join the Spence staff for four days of Logo exploration and creation.

Logo Immersion, July 20-24

Choose from among several strands including multimedia, turtle geometry, simulations, and game design. This year's theme is Logo on the Internet.

Logo Robotics, July 27-31

Design and build cybernetic devices of all kinds and control them using Logo programs. Use the popular Control Lab® from LEGO Dacta, the Programmable Brick, and PC Logo® Robotics.

For more information contact:

Logo Foundation, 250 West 85th Street New York, NY 10024, telephone: 212/579-8028, fax: 212/579-8013, e-mail: michaelt@media.mit.edu, URL: <http://el.www.media.mit.edu/logofoundation/>

and will focus on further Logo experiences along with other multimedia authoring programs. It is hoped that models of learning and teaching for gifted children with technology will be generated and dispersed through the literature. From future workshops it is anticipated that some teacher training can be developed, so that these methods and techniques can reach beyond the walls of the university. As one nine year old stated so eloquently in discussing the future ... "The future is what you make it." For professional educators down under, the future of educational technology for gifted children is what we make it!



References

- Bulls, M. (1996). *Computers in gifted education*. Unpublished annotated bibliography, Massey University.
- Davis, G.A. & Rimm, S.B. (1994). *Education of the gifted and talented (3rd ed)*. Needham Heights: Allyn & Bacon.
- Gallagher, J. & Gallagher, S. (1994). *Teaching the gifted child (4th ed)*. Needham Heights: Allyn & Bacon.
- Lockard, J., Abrams, P., & Many, W. (1997). *Microcomputers for twenty-first century educators (4th ed)*. New York: Harper Collins.
- Logo Computer Systems Incorporated, (1993). *MicroWorlds project builder™*. Highgate Springs, VT: LCSi.
- Marks, D.F. (1992). Training teachers of the gifted to use simulations. *Gifted Child Today*, 15(6), 25-27.
- McAlpine, D. (1996). The identification of children with special abilities. In D. McAlpine & R. Moltzen (eds), *Gifted and talented: New Zealand perspectives* (pp. 63-90). Palmerston North: ERDC Press.
- Melton, D. (1985). *Written and illustrated by....* Kansas City: Landmark Editions.
- Middleton, J.A., Littlefield, J. & Lehrer, R. (1992). Gifted students' conceptions of academic fun: An examination of critical constructs for gifted education. *Gifted Child Quarterly*, 36(1), 38-44.
- Moltzen, R. (1996). Characteristics of gifted children. In D. McAlpine & R. Moltzen (eds), *Gifted and talented: New*

Zealand perspectives (pp.43-62). Palmerston North: ERDC Press.

- Nolan, P. & Ryba, K. (1986). *Assessing learning with Logo*. Eugene, Oregon: International Society for Technology in Education.
- Papert, S. (1992). *The children's machine: Rethinking school in the age of the computer*. New York: Basic Books.
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York: Basic Books.
- Renzulli, J.S. & Reis, S.M. (1985). *The schoolwide enrichment model: A comprehensive plan for educational excellence*. Mansfield Center, CT: Creative Learning Press.
- Riley, T.L. (1997). Tools for discovery: Conceptual themes in the classroom. *Gifted Child Today*, 20(1), 30-33, 50.
- Torrance, E.P. (1966). *Torrance tests of creative thinking*. Bensenville, IL: Scholastic Testing Service.
- Troxclair, D., Stephens, K., Bennett, T., & Karnes, F. (1996). Teaching technology: Multimedia presentations in the classroom. *Gifted Child Today*, 19(5) 34-36, 47.

About the authors

Mark Brown is a lecturer in educational technology in the Department of Learning and Teaching, Massey University, New Zealand. He specializes in trying to link contemporary learning theory to the application of new educational technologies in schools.

Tracy Riley is a lecturer in gifted education in the Department of Learning and Teaching, Massey University, New Zealand. She specializes in developing programs and policies appropriate for the education of children with special abilities.

Mark Brown

M.E.Brown@massey.ac.nz

Tracy Riley

T.L.Riley@massey.ac.nz

Web Player



New from LCSi!

The MicroWorlds Web Player

The MicroWorlds Web Player allows teachers and students around the world to view MicroWorlds 2.0 projects via the World Wide Web.

Share MicroWorlds projects that have interactive and powerful animation capabilities, or produce your own dynamic web content.

For a free download of our new Web Player, please visit our web site at <http://www.lcsi.ca>

(800) 321-5646



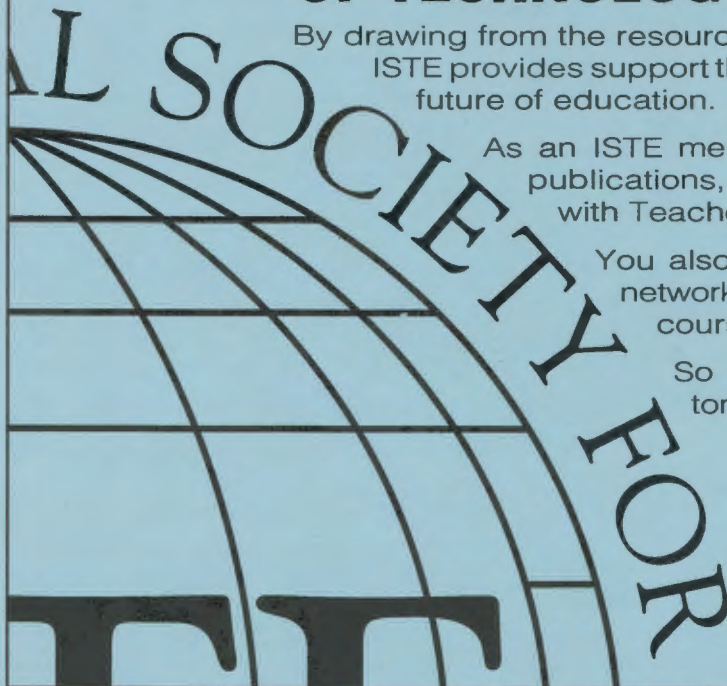
ISTE BRINGS THE WORLD OF TECHNOLOGY CLOSER TO YOU.

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.



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
America Online: ISTE CompuServe: 70014,2117
Internet: cust_svc@ccmail.uoregon.edu
World Wide Web: <http://isteonline.uoregon.edu>

WE'LL PUT YOU IN TOUCH WITH THE WORLD.