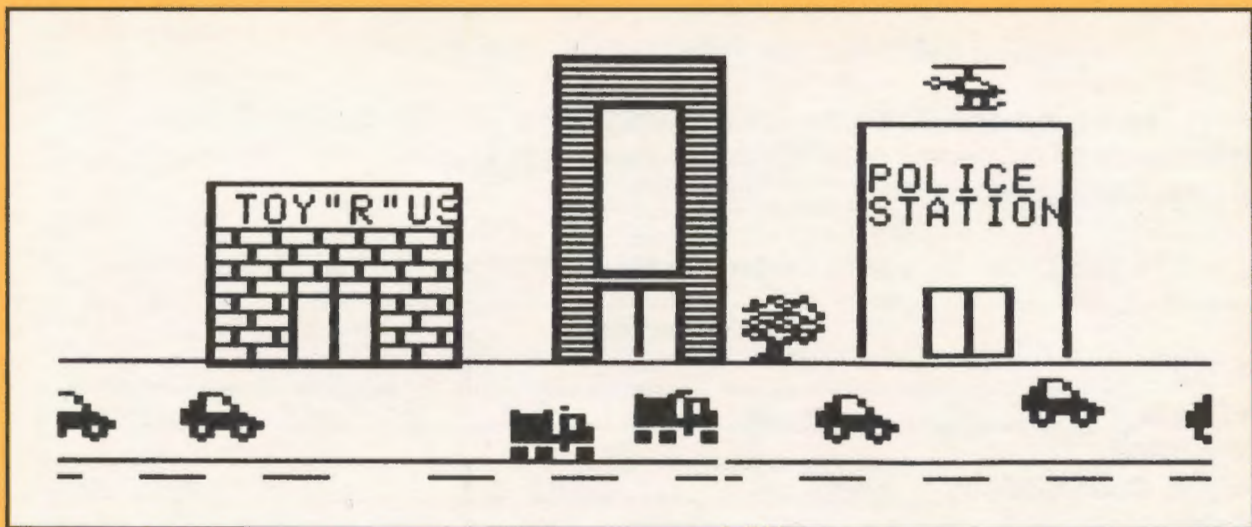

Journal of the ICCE Special Interest Group for Logo-Using Educators



LOGO EXCHANGE

FEBRUARY 1988

VOLUME 6 NUMBER 6



International Council for Computers in Education



Publications

Non-Profit
Organization
U.S. POSTAGE
PAID
Permit No. 63
Eugene, OR

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

Edited by
Theodore C. Burrowes and Sharon K. Burrowes

An ICCE Publication

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

The ECCO Logo Project
Includes:

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

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

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

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

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

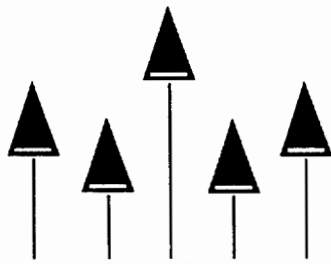
Please send me ____ copies
of *The ECCO Logo Project*
at \$20 each. Total price \$ _____

Add \$2.50 for shipping \$ _____

Add \$2.50 handling
for billed orders \$ _____

Total Amount \$ _____

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



LOGO EXCHANGE

VOLUME 6 NUMBER 6

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

FEBRUARY 1988

Founding Editor

Tom Lough

Editor-in-Chief

Sharon Burrowes Yoder

International Editor

Dennis Harper

Senior Contributing Editor

Robs Muir

Field Editors

Eduardo Cavallo
Patricia Dowling
Anne McDougall
Harry Pinxteren
Fatimata Seye Sylla
Hillel Weintraub

Contributing Editors

ASTROLUG
Eadie Adamson
Gina Bull
Glen Bull
Doug Clements
Bill Craig
Sandy Dawson
Judi Harris
Barbara Randolph
Linda Sherman
Gary Stager

Managing Editor

Anita Best

Special Interest Group Coordinator

Keith Wetzel

Advertising Director

Kathleen Geygan

SIGLogo Board of Directors

Peter Rawitsch, President
Gary Stager, Vice-President
Ted Norton, Communications

Publisher

International Council for
Computers in Education

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

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

CONTENTS

From the Editor		
	Sharon Burrowes Yoder	2
From the President		
	Peter Rawitsch	2
Monthly Musings — <i>HyperHype</i>		
	Tom Lough	3
SIGLogo Bylaws		4
Logo Ideas — <i>Creating a Kaleidoscope</i>		
	Eadie Adamson	6
<i>Natural Language Microworlds:</i>		
	<i>An answer to the Challenge</i>	
	Michael Tempel	10
Teaching Tools — <i>Sound Concepts</i>		
	Glen Bull and Gina Bull	13
LogoLinx — <i>Rhyme Paradigm</i>		
	Judi Harris	16
Stager's Stuff — <i>The Subject is the Predicate</i>		
	Gary S. Stager	18
InLXual Challenges — <i>Structuring the Structure</i>		
	Robs Muir	21
Testudinal Testimony — <i>Research on Variables, Algebra, and Logo. Part II</i>		
	Douglas H. Clements	23
PenPoints — <i>Logo Book Reviews</i>		
	Gayle Lawrence	26
LogoPals		
	Barbara Randolph	27
LXionary		
	Bill Craig	28
LX International Logo News		
	Dennis Harper	29
What Your Love of Logo Reveals about You		
	Cara Garcia	31

SIGLogo Membership (includes *The Logo Exchange*)

	U.S.	NON-U.S.
ICCE Member	24.95	29.95
Non-ICCE Member	29.95	34.95
ICCE Membership (includes <i>The Computing Teacher</i>)	28.50	31.50

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

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

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

From the Editor

by Sharon Burrowes Yoder

In this issue of *LX* you will find the bylaws of the SIGLogo. (Remember, long time *LX* subscribers that your *LX* subscription now carries membership in the ICCE SIGLogo!) Take a moment to look over these rules that govern this organization, and think about how you might like to contribute: by being an officer, by submitting articles, by sending letters to the editor...

Interwoven into this issue are a couple of, what I consider to be important threads in the ongoing Logo culture. First, we see a broadening of Logo beyond typical activities such as drawing polygons. Some of these articles barely mention Logo, at least at the outset, and yet are clearly about Logo-like thinking. Glen and Gina Bull use Logo to explore sound (largely without TONE), Tom Lough takes a look at HyperCard, Doug Clements examines algebraic concepts in some depth, while Sandy Dawson discusses the work of Gattegno, a mathematics educator influenced by Piaget.

A second thread is examination of Logo in more depth. Robs Muir takes a look at some specifics of Logo grammar, while Gary Stager explores the idea of predicates. If you are a relative novice, you may find these articles a bit overwhelming. Don't be discouraged, however. Put them away for a few months; then come back and see how much more you understand!

If Logo is going to continue to grow, we need both depth and breath. We need more microworlds like Glen and Gina give us; we need more research such as Doug reports; and we need discussions of underlying principles such as Robs has given us. I hope this issue, as well as others this year, have and will help you to grow in your understanding of Logo. I certainly never cease to learn more about Logo and to be amazed as it's flexibility. Hopefully the same is true of you!

Sharon Burrowes Yoder, ICCE / SIGLogo, University of Oregon, 1787 Agate Street, Eugene, OR 97405-9905

From the President

Why Did The Chicken...

by Peter Rawitsch

Acting-President of SIGLogo

At least once a month, the cafeteria at my elementary school offers a menu that includes chicken noodle soup and an egg salad sandwich. It always reminds me of the age old question about chickens and eggs. My role in the "birth" of SIGLogo raises an equally puzzling question. Last December I met with the SIGLogo Board of Directors to create the SIGLogo Bylaws that create the SIGLogo Board of Directors. Which came first? In this timeless tradition of recursive thinking, I welcome you to the Special Interest Group for Logo Using Educators.

This has been a challenging and rewarding year for SIGLogo. The challenge has been to let the international Logo-using community know we exist. The *Logo Exchange's* publication schedule for the fall and winter was very disappointing for all of us. As it begins to get on track, memberships, in the form of *LX* subscriptions, continue to grow. The reward has been the support we have received in the form of articles from faithful columnists and from the dedicated staff at ICCE. I am especially grateful for the advice and cooperation I have been given by Sharon Burrowes Yoder, the *LX* editor, and Keith Wetzel, the Special Projects Coordinator.

The next step for us will take place at NECC '88 in Dallas, Texas. SIGLogo will hold its first general business meeting on June 15th from 7 - 9 P.M. Look for announcements as to location on general bulletin boards and at the ICCE booth in the exhibit hall. The meeting will be led by Gary Stager, Acting-Vice President, Ted Norton, Acting-Communications Officer, and myself. At that time, members will have an opportunity to discuss and ratify the Bylaws. (A majority of the members who are present at the meeting will be needed for ratification.) We will also consider the future of SIGLogo. The establishment of a SIGLogo bulletin board and the sponsorship of a Logo conference are among the possible topics.

In addition to your ideas, we will be looking for volunteers. We need someone to serve as Acting-Treasurer for the coming year. The Logo Exchange is always in need of samples of students' work and articles about interesting ideas that can be used in the classroom. A Nominations Committee will also be appointed to recommend candidates for the April, 1989 election of Vice President, Communications Officer, and Treasurer. As prescribed in the Bylaws, Gary Stager will serve as SIGLogo President beginning at next year's NECC conference.

Gary Stager, Ted Norton and I look forward to meeting you at NECC '88 and to working with you in the coming year. SIGLogo is at the crossroads. Come and help us get it to the other side.

Peter Rawitsch, P.O. Box 254, Guilderland, NY 12085.

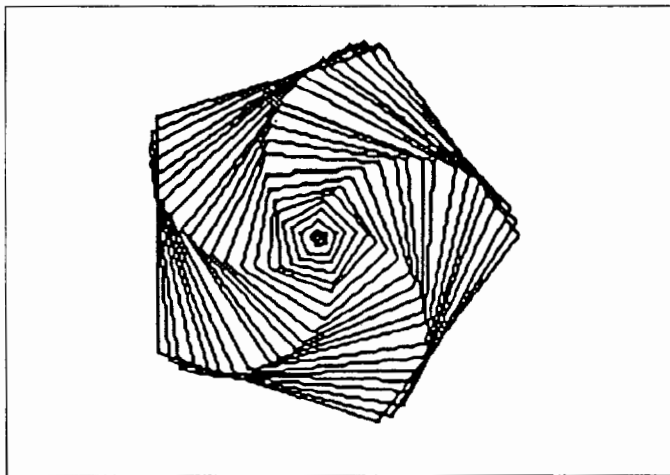
Cover: Freddy Victoria, The Allen-Stevenson School, New York, N.Y. Freddy did this sample in grade 4 upon first encountering LogoWriter. The assignment was to make a street scene, making use of stamped shapes, filled or shaded shapes, and label text. Later the class used their street scenes and programmed moving vehicles with keys to provide control over the speed.

Monthly Musings

HyperHype by Tom Lough, Founding Editor

Recently, the prefix *hyper* has taken on a couple of new meanings for me. For example, our four and one-half year old son graciously shared some of his *hyper* behavior on several occasions over the holidays. The dizzying frenzy of his antics certainly created a memorable definition, needless to say.

Another meaning is embedded in the familiar looking graphic below.



Almost instinctively, one is tempted to think:

```
TO POLYSPI :SIDE :ANGLE
etc.
```

If you examine the figure carefully, however, you might hesitate. The figure was not drawn by a turtle going forward and turning left in a changing manner. Rather, the figure consists of a number of regular pentagons nested inside each other and slightly rotated with respect to each other.

The figure was drawn with the paint tools in a new application for the Macintosh called *HyperCard*. Many of you have probably seen or heard of it by the time you read this. As in the figure above, I believe we can observe something from more than just a quick examination of *HyperCard*.

During my first encounter with *HyperCard*, I explored the main ideas. The screen became a *card* with could contain text or graphic information and buttons. The cards could be grouped in *stacks*. In fact, the stack seemed to be the main metaphor.

I enjoyed browsing through address books, to-do lists, and pages of artwork. It was fun to click the various buttons and discover what action followed.

But it was not until I acquired *The Complete HyperCard Handbook*

by Danny Goodman (Bantam 1987, ISBN 0-553-34391-2, \$29.95) that I really began to understand what *HyperCard* was all about. Just as the second look at the "spiral" pentagon above may have given you second thoughts, Goodman's book gave me a glimpse of some exciting future possibilities.

In his book, Goodman quotes Bill Atkinson, the developer of *HyperCard*, suggesting the application is an authoring tool, and comparing it with Logo and other computer languages. (p. xxviii) On the same page, Atkinson also predicts that "education will pick it up right away," and suggests several interesting sample projects.

But more than Atkinson's comments, the thing which excited me about *HyperCard* was the HyperTalk language which accompanied it. Although definitely *not* Logo, this language is modular and extensible. Moreover, it was designed to enable ordinary users to create their own specialized applications, giving rise to the term "stackware." Suddenly, it is possible for you and me to write reasonably sophisticated programs for reasonably sophisticated applications for the Macintosh.

Sure, there are some (present) limitations. The minimum memory, for example, is one megabyte, with a hard drive recommended. You cannot create a card larger than the Macintosh Plus screen. You cannot look at more than one card at a time. But what *HyperCard* and HyperTalk can do within these limitation is both impressive and exciting.

As you might suspect, I sense many parallels between *HyperCard* and Logo. I'll not go into them here, however. Instead, let me invite you to explore *HyperCard* for yourself. As for me, I'd like to take up the challenge extended by Bill Atkinson to the education community. Let's "pick it up" and see what possibilities we can discover with the new tool. As always, I would be interested in your ideas.

FD 100!

Tom Lough, P. O. Box 5341, Charlottesville, VA 22905

Answers to this month's *InLXual Challenges* article, see page 18.
(Read upside-down and in a mirror.)

```

e'  L'  L'
2'  BOZ' BEI2LVLE' XBOZ'
4'  L'  L'
3'  C2' HL' 2L' BU'
5'  IE' KEBEVL' IBEI2E'
I'  BADDGE' EIK2L' LV2L'
    
```

Bylaws for SIGLogo

Special Interest Group for Logo-Using Educators (SIGLogo) International Council for Computers in Education

Article I - NAME

This organization will be known as the Special Interest Group for Logo-Using Educators of the International Council for Computers in Education. It is hereafter referred to as SIGLogo. The International Council for Computers in Education is hereafter referred to as ICCE.

Article II - PURPOSE

SIGLogo will support and promote the use of Logo as a tool for learning and as a language for computer science. Specifically, SIGLogo will:

- a. Collect and disseminate information through publications and electronic communication networks.
- b. Sponsor all or part of meetings, conferences, and workshops.
- c. Organize working groups for research, study, and writing activities to meet the needs of its membership.
- d. Develop and distribute position papers on Logo implementation and philosophy.
- e. Serve as the Logo-using educators' liaison to the ICCE and its other Special Interest Groups.

Article III - MEMBERSHIP, DUES, and PRIVILEGES

Section 1: Membership

- a. Individual members of ICCE are eligible for membership in SIGLogo upon application and payment of ICCE Individual Member SIGLogo dues.
- b. Non-individual members of ICCE are eligible for membership in SIGLogo upon application and payment of non-ICCE Individual Member SIGLogo dues.

Section 2: Dues

The SIGLogo Board of Directors will determine the annual rate of membership dues for Individual Members and non-Members of ICCE. Upon approval of the ICCE Board, the dues rates will become effective on July 1 of each fiscal year.

Section 3: Rights and Privileges

All members of ICCE have the right to attend and to speak at SIGLogo meetings and at meetings of its Board of Directors. ICCE members also have the right to serve on committees of SIGLogo. The rights to hold office and to vote are reserved to SIGLogo members.

Section 4: Termination of Membership

A member may resign from SIGLogo by submitting a written resignation to the Communications Officer. Membership may be automatically terminated for failure by a member to pay dues. A member may also be expelled or suspended for reasonable cause after a hearing before the SIGLogo Board of Directors and by a majority vote of the SIGLogo Board.

Section 5: Membership Roster

The collection of dues and the maintenance of a membership roster will be administered by the ICCE. A copy of the membership will be provided to the ICCE Board upon request. The membership roster will be used for the ordinary business of the SIGLogo, such as mailing of meeting notices and ballots.

Article IV - OFFICERS

Section 1: Election of Officers

The Vice President, Communications Officer and Treasurer of SIGLogo are selected by a vote of the membership. The Vice President becomes the President at the end of the term.

- a. The SIGLogo President will appoint a nominations committee to nominate candidates for the offices of Vice President, Communications Officer and Treasurer. Every two years, nominations for the ballot will be submitted prior to February 1 of the election year.
- b. ICCE will oversee the election of SIGLogo officers. Vitae of the candidates, election ballots and voting instructions will be mailed to all members of SIGLogo or printed in the Logo Exchange no later than March 1 of the election year. Members will cast their votes no later than April 30 of the election year. Ballots post-marked later than April 30 will not be valid.
- c. The candidates receiving a simple majority of votes from the total votes cast will assume duties of their respective offices at the NECC conference, or by July 1, whichever comes first.
- d. All terms of office are for two years and expire upon resignation of an officer or at the NECC conference, or July 1, whichever comes first.

Section 2: Board of Directors

The four voting members of the SIGLogo Board of Directors will be the President, Vice President, Communications Officer, and the Treasurer. The Editor(s) will serve as ex-officio members of the board.

Section 3: President

The President is the principal officer. The duties of the President include:

- a. Calling and presiding at meetings of the Board of Directors and of SIGLogo.
- b. Appointing all standing and ad hoc committee chairpersons.
- c. Appointing members to fill offices that may become vacant through resignation, incapacity or ineligibility of an incumbent officer.
- d. Acting as the SIGLogo liaison to the ICCE Board and its other Special Interest Groups.

Section 4: Vice-President

The duties of the Vice-President include:

- a. Assuming the duties of the President in the event of the President's resignation or incapacity.
- b. Coordinating SIGLogo standing committees.
- c. Communicating with regional contacts and setting up conferences and workshops.
- d. Assuming the office of President after serving as Vice President.

Bylaws for SIGLogo — CONTINUED

Section 5: Communications Officer

The duties of the Communications Officer include:

- a. Keeping minutes of business meetings of SIGLogo and of the Board of Directors.
- b. Maintaining records and correspondence of SIGLogo.
- c. Notifying members of the Board of Directors of the time, place and agenda of the Board of Directors meetings.
- d. Notifying the general membership of SIGLogo of the time, place and topic of the general meetings.
- e. Sending official notifications to the ICCE Board of Directors of changes of the officers of SIGLogo.

Section 6: Treasurer

The duties of the Treasurer include:

- a. Overseeing finances for SIGLogo.
- b. Filing financial reports as are required by ICCE.

Section 7: Editor(s)

The Editor(s) will oversee all SIGLogo publications. The SIGLogo Editor(s) are appointed by the SIGLogo President and the ICCE editorial staff upon approval of the SIGLogo Board of Directors.

Article V - BOARD OF DIRECTORS

SIGLogo will be governed by the Board of Directors, which comprises the four officers. All decisions made by the Board of Directors must be approved by a majority of the board.

Article VI - BOARD OF DIRECTORS MEETINGS

The SIGLogo Board will conduct regularly scheduled business meetings each year. At least one meeting will be held in a place that is open to all members.

Article VII - AMENDMENTS

Section 1:

- a. A resolution by the majority of the Board of Directors present at a meeting that has a quorum will be sufficient to cause an amendment to the bylaws to be voted upon by SIGLogo members. An amendment can be proposed to the Board of Directors by any SIGLogo member.
- b. A petition by two percent of the SIGLogo members will be sufficient to cause an amendment to the bylaws to be voted upon by SIGLogo members. The right to petition will be independent of any decisions taken in accordance with Section 1, Paragraph a, of the article.

Section 2:

The proposed amendment will be presented to the ICCE Board of Directors and SIGLogo Board of Directors prior to distribution to the membership.

Section 3:

The proposed amendment will be voted on by the following ballot procedure:

- a. The ballots will be mailed out or printed in the Logo Exchange and returned to such address as will be specified by the ICCE Board. The ballot will include (1) a copy of the proposed amendment,

including specification of the date on which it will become effective if approved, and (2) a copy of the article(s) in the existing bylaws that is (are) being proposed for amendment.

- b. Only ballots reaching the return address designed by ICCE within forty days after the last ballot was mailed out and postmarked within thirty days after the last ballot was mailed out will be valid.

Section 4:

An amendment will become effective if approved by a majority of the valid ballots cast by voting members of SIGLogo, on the effective date specified on the ballots or three months after the last ballot was mailed, whichever is later, unless specifically disapproved by action of the SIGLogo or the ICCE Board of Directors.

Article VIII - DISSOLUTION

SIGLogo can be dissolved by the consent of its members or by joint action taken by the SIGLogo and ICCE Board of Directors. Consent of the members of SIGLogo to dissolution will consist of unanimous agreement of the officers of SIGLogo together with a majority vote at a meeting which has been announced in advance to all members of SIGLogo. Upon dissolution of SIGLogo, its assets, if any, will be used to repay any outstanding debts incurred on behalf of SIGLogo. Any remaining debts will be paid by the ICCE. Any assets remaining after payment of SIGLogo debts remain the property of ICCE.

Article IX - REGIONAL CHAPTERS

Section 1: Purpose

Regional Chapters will be allowed under the operating structure of SIGLogo. Regional Chapters will participate in a national network of SIGLogo Regional Chapters.

Section 2: Formation

A group wishing to become affiliated with SIGLogo as a Regional Chapter will petition for Regional Chapter status by submitting its membership guidelines, bylaws or operating objectives to the SIGLogo Board of Directors.

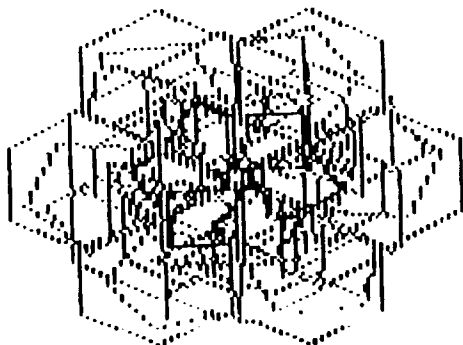
Section 3: Eligibility

Regional Logo groups will become eligible for recognition as Regional Chapters upon submission of a membership roster and verification that a majority of their members are current SIGLogo members.

Section 4: Regional Chapter Support

- a. Each Regional Chapter will receive one copy of SIGLogo publications and notice of SIGLogo meetings, conferences and workshops. Additional correspondence involving issues to be addressed by SIGLogo, committee appointments and policy statements will also be sent to Regional Chapters.
- b. Regional Chapters and SIGLogo will work together to establish groups for research, study and writing activities.
- c. Regional Chapters may include the names of ICCE and SIGLogo in communications and advertising for non-profit activities.

Creating a Kaleidoscope



by Eadie Adamson

Kaleidoscopes are fascinating objects for children of all ages. A Kaleidoscope project is a nice follow-up for a random lesson and polygon explorations (see LX December, 1987 and January, 1988). I first used this idea a number of years ago when working with Commodore Logo. I had picked up David Thornburg's *Computer Art and Animation* (Addison-Wesley, 1984)

Incidentally, this book is full of *many* wonderful ideas about working with polygons as well as some ideas on planning animation—an excellent supplement if your interests lie in this area. You might want to refer to this text for a slightly different explanation of the kaleidoscope idea, for which credit and a deep bow from me goes to David Thornburg.

My students and I had been working with polygons, had used them in geometric patterns and had done a little study of Islamic design. David's chapter about making a kaleidoscope sounded like fun for us to try. I presented the idea to my students in a series of steps which built up to a final Kaleidoscope generator. (The version below is for LogoWriter, but can be adapted for other versions of Logo.)

Before you present the project to your students, make your own KALEIDOSCOPE program so that you can use it as an illustration of what you're aiming for. It's interesting to show it to students and ask them to analyze what is happening (without, of course, showing the procedures themselves).

The first requirement of the project is to choose at least six procedures for regular polygons, all written with a variable size. Since my students had already worked on the polygon project, this task was refined to:

Choose your six favorite shapes.
PASTE them on a new page.
Name the page KALEIDOSCOPE (or IMAGE.)

I like to make this activity a cooperative problem solving session. In this case, I really do have a completed program in mind and I strongly oppose simply giving students a page of procedures to type in and run (a friend of mine calls this Fascist Logo). Here's how to turn this project into a participatory one:

Step 1: A Problem to Pose

How can you start with the turtle at HOME, make the turtle jump out a given distance, draw a polygon, and jump back to the starting position?

You might simply assign this as a problem for students to solve on their own, and then have them share solutions. Or you might prefer make it a joint problem-solving session: ask the students to help you work out the necessary commands to produce a single shape which the turtle draws by moving out from HOME, drawing the shape, and moving back to HOME. Use a question mark for *some distance* out from the center and for a shape of *some size*. If they haven't used procedure inputs much yet, this leads nicely into substituting values for the names in WINDMILL (below). Together, write a procedure to do this with a single shape.

Step 2: Drawing Shapes Around a Center

The next task will sound more difficult, although it is really simple: How can you change this procedure so that it draws six shapes spaced equally around the center?

Again, this can be an individual problem to solve and then students can share strategies and solutions. As a group project, approach it this way: Ask how to get six of these shapes (at this point you might show one generation of the completed IMAGE to illustrate what you're asking them to work out) and have them suggest how to change the procedure to do this. See if they can relate it to their HEXAGON to find the necessary turn.

Step 3: Writing a Windmill Procedure

Next, having worked out the mechanics of the beginning process, suggest choosing a favorite shape and write a WINDMILL procedure which looks like this, using any shape desired where TRI is included below:

```
TO WINDMILL :DIST :SIZE
  REPEAT 6 [PU FORWARD :DIST PD TRI :SIZE
    PU BACK :DIST]
  END
```

Now the students are ready to experiment with sizes, distances, and colors. Some may add color as another input for WINDMILL; others may simply elect the strategy of changing pen color before drawing. Encourage them to create procedures (with new names) of the designs they like. (This gets at the idea that a procedure can be used within another procedure, something that is sometimes hard to make clear.) If your class has been studying Islamic design they will find some nice relationships here. Allow plenty of time, one full class period at the very least, for experimentation with this part of the process. Be sure to save and print often. You'll find you will have quite a collection of beautiful images just from this simple beginning.

Step 4: The PICK Tool

Give this tool procedure to your students before continuing the

Creating a Kaleidoscope — CONTINUED

project:

```
TO PICK :LIST
  OUTPUT ITEM (1 + RANDOM COUNT :LIST) :LIST
END
```

Explain just enough to satisfy your students about how PICK works. A good explanation is picking numbers or notes out of a hat. You might want to use the PICK procedure as an opportunity to explain how OUTPUT works.

```
PICK [JIM MARY JACK JANE]
```

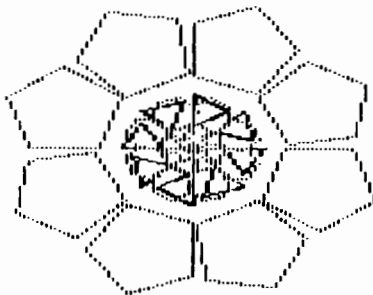
will generate an error message: I DON'T KNOW WHAT TO DO WITH JIM IN PICK. Here, if you want to digress, you can talk about the various ways you can display text in LogoWriter: PRINT, TYPE, INSERT, and SHOW. Encourage some experimenting with PICK and lists of names or numbers. Be sure to point out that PICK is a procedure used often in Logo when one needs the computer to make a random choice of items from a list. It's a useful procedure which might well have been included as a Logo primitive.

Step 5: Using the PICK tool

Now make a slight adjustment to WINDMILL to allow WINDMILL to use a shape chosen randomly by PICK (changes are in lower case):

```
TO WINDMILL :DIST :NAME
  REPEAT 6 [FORWARD :DIST PD RUN :NAME PU
           BACK :DIST RIGHT 60]
END
```

Note: Some students may want to preserve the "old" WINDMILL. They may do this by copying the original version first, then renaming the old unchanged version. The new WINDMILL will not work until IMAGE (below) has been written. WINDMILL is incorporated into IMAGE and fed information chosen randomly by IMAGE before it is generated.



Step 6: Creating the IMAGE procedure

The final step in creating IMAGE needs some explanation and does, unfortunately, require giving the students something to type in. I prefer to do this one line at a time, giving the explanation as we go along (see below). I've added line numbers here to connect with the explana-

tions. Don't include these in your procedures!

```
TO IMAGE
  MAKE "LIST [TRI SQ PENT HEX OCT DEC] (1)
  MAKE "NAME (SENTENCE
             (PICK :LIST) (5 + RANDOM 30)) (2)
  MAKE "DIST RANDOM 45 (3)
  SETC 1 + RANDOM 5 (4)
  PU
  WINDMILL :DIST :NAME (5)
  MAKE "TEMP READCHAR (6)
  IMAGE (7)
END
```

(1) This line makes a LIST of the names of the shapes procedures that you plan to use (you'd change that line if you added new shapes).

(2) Next, PICK is used to choose a shape name randomly from the list and assign the shape a random size that is at least 5 (5 + RANDOM 30). It makes a sentence of the two items: the shape and its size. It might choose TRI 30, for example, then later choose OCT 20. (Later you could experiment with changing the minimum size.)

(3) Next, DIST is given a value generated by RANDOM 45 (0 to 44, remember?). You might experiment with changing these values to see how a different distance affects the design.

(4) Next, a random pen color is picked, but black is omitted (how?)

(5) Finally, WINDMILL is run using this information.

(6) Notice something in the line MAKE "TEMP READCHAR. READCHAR is the short way of saying Read a Character. READCHAR tells the computer to wait for a key to be pressed before continuing. Once a key is pressed, it goes to the next command, IMAGE. This starts a new version of IMAGE, randomly choosing a shape, a size, a distance, and a pen color. Point out that this is how you can make many things interactive by asking a player to press a key when ready, for instance.

There are a couple of programming issues worth mentioning here. In both cases, you make the code cleaner, but then they are probably harder for beginners to understand and may just add confusion.

- You could substitute IGNORE READCHAR for MAKE "TEMP READCHAR, but then you'd need to write another short procedure:

```
TO IGNORE :KEY
  END
```

- You can "collapse" the code of IMAGE, causing the MAKE statements to be unnecessary.

```
TO IMAGE
  SETC 1 + RANDOM 5
  PU
```

Creating a Kaleidoscope — CONTINUED

```

WINDMILL
RANDOM 45
(SENTENCE PICK [TRI SQ PENT HEX OCT DEC]
 5 + RANDOM 30)
IGNORE READCHAR
IMAGE
END

```

(7) Recursion is one of the most powerful processes one can use in Logo. Can you see that the recursive call to IMAGE creates an indefinite loop? If it were a FOR loop such as found in the BASIC language, it would repeat the process a definite number of times. This example of tail-end recursion shows one way that the power of recursion can be used to build programming structures to meet your needs.

Here's IMAGE again, with the additions to make Logo give you the information (in lower case):

```

TO IMAGE
MAKE "LIST [your shapes]
MAKE "NAME SENTENCE :LIST 5 + RANDOM 30
TYPE SENTENCE "SHAPE :NAME
TYPE CHAR 13
MAKE "DIST RANDOM 45
TYPE SENTENCE "DISTANCE :DIST
TYPE CHAR 13
SETC 1 + RANDOM 5
TYPE SENTENCE "SETC COLOR
TYPE CHAR 13
PU
WINDMILL :DIST :NAME
MAKE "TEMP RC
IMAGE
END

```

TYPE makes the message appear in LogoWriter's command center. Since it does not give a carriage return, generate one by TYPE CHAR 13 (CHAR 13 is ASCII) for the RETURN key. You could also write (TYPE SENTENCE "SETC COLOR CHAR 13). Note the parentheses at each end of this statement.

The output from your IMAGE procedure should read something like this:

```

SHAPE TRI 30
DISTANCE 20
SETC 5

```

Now if you want to save a design by making a procedure, you need just a small change to the WINDMILL procedure to allow you to use it with whatever shape you wish, BUT be sure to change the procedure name also, so you can have this procedure to use at the same time as WINDMILL:

```

TO MILL :DIST :NAME :COLOR
SETC :COLOR

```

```

REPEAT 6 [PU FORWARD :DIST PD RUN :NAME PU
BACK :DIST RIGHT 60]
END

```

Here's how to use MILL:

```
MILL 40 [TRI 25] 4
```

In this example 40 is the value for DIST and TRI 25 is the NAME; (the shape with its input must be enclosed in a list of its own). The COLOR selected here is 4. You can use MILL in a procedure now to reproduce a design produced by IMAGE. Give your design a special name, then incorporate the information for each generation of IMAGE into a single line for MILL, like this:

```

TO PIC
MILL 40 [TRI 25] 4
MILL 30 [OCT 15] 3
MILL 45 [PENT 20] 2
END

```

Other Ideas for Image

If you're using LogoWriter, try creating the design with three or four turtles instead of one. A SETUP procedure will be necessary to get all the turtles to start in the same place. SETUP will also need to set each turtle to a different heading to begin. Three turtles work well if you generate six shapes, four turtles if you use eight. Each will draw a shape, then turn and draw the shape again. (See solution below.)

One idea that is only marginally satisfactory with LogoWriter is to FILL the shapes as they are drawn. This works at first, until shapes overlap. Then sometimes the results are not pleasing. Still, it's interesting to try!

Another idea might be to go to the SHAPES page and design a series of geometric shapes. Then change IMAGE so that it selects the shape, now no longer a procedure but a shape number, and STAMPS rather than draws the shape. You might even adjust the minimum distance from the center, change the turn, and allow the turtles to draw colored lines from the center each time (try something like a turn of 15) before STAMPing the SHAPE.

You might put a collection of printouts together to form a quilt or cut them out, color them, and paste them together to make a stained glass window or add sound as the turtle(s) draw, or -- Any other brilliant ideas out there?

Eadie Adamson, Allen - Stevenson School, 132 East 78th St., New York, NY 10021

Some Polygons to use with IMAGE:

```

TO TRI :SIZE
REPEAT 3 [FORWARD :SIZE RIGHT 120]

```

Creating a Kaleidoscope — CONTINUED

```

END

TO SQ :SIZE
  REPEAT 4 [FORWARD :SIZE RIGHT 90]
END

TO PENT :SIZE
  REPEAT 5 [FORWARD :SIZE RIGHT 72]
END

TO HEX :SIZE
  REPEAT 6 [FORWARD :SIZE RIGHT 60]
END

TO OCT :SIZE
  REPEAT 6 [FORWARD :SIZE RIGHT 60]
END

```

The PICK procedure:

```

TO PICK :LIST
  OUTPUT ITEM (1 + RANDOM COUNT :LIST) :LIST
END

```

The IMAGE procedure:

```

TO IMAGE
  MAKE "LIST [TRI SQ HEX PENT OCT]
  MAKE "NAME SE PICK :LIST 5 + RANDOM 30
  MAKE "DIST RANDOM 45
  SETC 1 + RANDOM 5
  PU
  WINDMILL :DIST :NAME
  MAKE "NAME READCHAR
  IMAGE
END

TO WINDMILL :DIST :NAME
  REPEAT 8 [FORWARD :DIST PD RUN :NAME PU
            BACK :DIST RIGHT 45]
END

```

For four turtles, eight shapes each time:

```

TO SETUP
  TELL [1 2 3]
  PU SETPOS [0 0]
  TELL 1
  SETH 90
  TELL 2
  SETH 180
  TELL 3
  SETH 270
END

```

```

TO I
  MAKE "LIST [TRI SQ HEX PENT OCT]
  MAKE "NAME SENTENCE PICK :LIST 5 + RANDOM 30
  MAKE "DIST RANDOM 45
  SETC 1 + RANDOM 5
  PU
  WINDMILL2 :DIST :NAME
  MAKE "NAME READCHAR
  I
END

TO WINDMILL2 :DIST :NAME
  TELL [0 1 2 3]
  HT
  PU
  REPEAT 2 [FORWARD :DIST PD RUN :NAME PU
            BACK :DIST RIGHT 45]
END

```

Expanding Kaleidoscope: Reporting on the Process

Suppose you'd like to be able to recreate some of the designs your IMAGE program generated. To do this, you can add a few lines to IMAGE which ask the computer to report its random choices as it runs.

What would you need to know? The shape—in this case, the NAME—chosen by IMAGE, and its size. You'd also need the distance and pencolor. It's not hard to make the computer give you this information at the bottom of the screen while the procedure runs. Since IMAGE waits for you to press a key before it continues, you have all the time you want to write down what's going on before generating another shape. With LogoWriter you can also scroll back up to read what's in the command center if you prefer to generate a number of shapes first.

Correction to Judi Harris' Picture Tools article:

```

For Terrapin Logo:
TO POS
  OUTPUT SENTENCE XCOR YCOR
END

TO LOAD .PIC : NAME
  READ :NAME
  READPICT :NAME
  PU
  SETX FIRST THING WORD : NAME "POS
  SETY LAST THING WORD : NAME "POS
  PD
  SETHEADING THING WORD : NAME "HEADING
  ST
  CLEARTEXT
  SPLITSCREEN
  END

```

With thanks to Mel Levin.

Natural Language Microworlds: An answer to the challenge

by Michael Tempel

In the October issue of *LX*, Robs Muir invites readers to explore the development of natural language microworlds. In that column, he raises a number of interesting ideas which this article will address and extend.

Initially, he draws what I think is a false dichotomy by separating list processing from turtle graphics. How about this pair of procedures, written in LogoWriter:

```
TO GO.AWAY
  TELL [0 1 2 3]
  ST
  PU
  CRAWL [0 1 2 3]
END
```

```
TO CRAWL :WHO
  IF EMPTY? :WHO [STOP]
  TELL FIRST :WHO
  RIGHT 45
  REPEAT 100 [FORWARD 1]
  HT
  CRAWL BUTFIRST :WHO
END
```

The structure of `crawl` might look familiar to you if you've seen:

```
TO TRIANGLE :LIST
  IF EMPTY? :LIST [STOP]
  PRINT :LIST
  TRIANGLE BUTFIRST :LIST
END
```

```
TRIANGLE [Welcome to Logo]
Welcome to Logo
to Logo
Logo
```

Another approach which integrates graphics and list processing has been created by Peter Rawitch. Peter created a Logo Zoo which puts together pieces of animals to form new creatures by processing lists of turtles to create the animals and lists of words to create their names. For example, each syllable of *croc-o-dile* is represented by a shape. By creating a number of these "subdivided" animals, "hybrid" animals can be created. For example, a *croc-o-ake* would be two parts crocodile and one part snake. (Can you expand on this idea as Peter has?)

It's also true that working with natural language in Logo does not necessarily require list processing. This is most true in LogoWriter where text may be manipulated through word processing. The perceived dichotomy between turtle graphics and list processing has never been as great as implied in most of the Logo literature. However, LogoWriter provides more things needing lists (such as multiple turtles) and more ways to process language. In fact, the natural language microworlds that Robs refers to require no more list processing knowledge

than drawing a square does.

But, as Arlo Guthrie once said, "that's not what I came here to tell you about." I came to talk about the classroom applications that, according to Robs, the natural language microworlds are crying for. They do, in fact exist, and were an integral part of Bob Lawler's "Beach World" back in 1980.

The term *application* may be misleading when referring to this kind of natural language microworld. What is being built is a learning environment and a set of activities that go with it. In the case of Beach World the goal was to create a "play pen" in which beginning readers could attach written words to familiar objects and actions. Bob wrote the program for his three year old daughter, Peggy, using the familiar setting of the Connecticut shore near their home. Objects such as boy, girl, boat, car, sun and dog were created with the shape editor that was part of the versions of Logo he used (TI Logo and Sprite Logo). Logo procedures made the objects appear and other procedures moved them around. For example

```
TO GIRL
  NEXT
  SETSHAPE :GIRL
END
```

```
TO DOG
  NEXT
  SETSHAPE :DOG
END
```

```
TO NEXT
  TELL WHO + 1
  ST
  PU
END
```

```
TO UP
  SETY YCOR + 5
END
```

```
TO DOWN
  SETY YCOR - 5
END
```

```
TO FAST
  SETSPEED 50
END
```

```
TO SLOWLY
  SETSPEED 10
END
```

The versions of Logo used for the Beach World had 30 or 32 turtles. The procedure `NEXT` was used to get the attention of the next turtle and made it visible on the screen. Each time a noun like `GIRL` or `CAR` was used, a new turtle appeared with that shape. It could then be

Natural Language Microworlds: An answer to the challenge — CONTINUED

moved around and placed anywhere on the background which consisted of a sky, beach, and water with some buildings and trees.

The vocabulary of this microworld was written on index cards. Children using the program could refer to the cards to remind themselves of the available words. Even if they did not know a word, typing it would most likely produce a result that would reveal its meaning.

An important aspect of the Beach World was that its content was familiar and personally meaningful to the user. It was also easily modifiable so that Bob or his older children could make changes for his youngest daughter, and they, in turn, used parts of the Beach World for their own Logo projects. In a school setting, older students, as well as adults, could create or modify these language microworlds for use by younger students.

The Beach World worked best when single words were typed and entered. Each word was a Logo command that produced a result on the screen. Having Logo understand an English phrase or sentence is a bit more tricky. Some people say that Logo is like English. This is mostly false. (That's why Robs' challenge is challenging!) What is true is that a Logo instruction is made up of words, usually separated by spaces. This is like English. Also, Logo words may be English words and these words may be the names of procedures. Thus, you can have Logo appear to understand English with things like:

THE BLUE TURTLE GOES SLOWLY using the following procedures:

```
TO THE
END
```

```
TO BLUE
  SETC 3
END
```

```
TO TURTLE
  SETSH 0
END
```

```
TO GOES
END
```

```
TO SLOWLY
  REPEAT 100 [FORWARD 1]
END
```

This sleight of hand is great for demonstrations, but inquisitive second graders will quickly discover that

```
TURTLE THE BLUE SLOWLY GOES
```

works just as well! Logo syntax is not like English. For example,

```
IF IT'S RAINING AND I'M GOING OUT, I'LL TAKE
AN UMBRELLA
```

would look like this with Logo syntax:

```
IF AND IT'S RAINING, I'M GOING OUT, I'LL TAKE
AN UMBRELLA.
```

Natural language also allows for much ambiguity that programming languages do not. "Turtle the blue slowly goes" may be peculiar English, but we still get the drift. It is as interesting to see what gibberish a program accepts as it is to see what proper sentences it "understands." How can we create a microworld that accepts reasonable English phrases and rejects nonsense?

When all the words in the language microworld are Logo commands that take no inputs, they may be strung together in any order. The result may make little sense in English. English phrases are made up of words linked together in specific ways. By using reporters along with commands that take inputs we can impose some structure which limits the phrases that are acceptable.

Harry Nelson uses this approach in the "Logo Grammar" section of the documentation accompanying the *On Logo* videotapes. The sentence,

```
DRAW A RED SQUARE
```

requires the procedures

```
TO DRAW :INPUT
  RUN :INPUT
END
```

```
TO A :INPUT
  OUTPUT :INPUT
END
```

```
TO SQUARE
  OUTPUT [REPEAT 4 [FORWARD 50 RIGHT 90]]
END
```

```
TO RED :INPUT
  OUTPUT SENTENCE [SETC 5] :INPUT
END
```

DRAW is the only command in this line. The other words are the names of reporters that feed information on down the line. The procedure A just passes its input on. RED adds the command SETC 5 to its input and sends the longer input on. Finally, DRAW runs the accumulated list.

DRAW must be the first word of the line and RED must precede SQUARE. In contrast, one could write a commands only version like this:

```
TO DRAW
END
```

Natural Language Microworlds: An answer to the challenge — CONTINUED

```
TO A
END
```

```
TO RED
  SETC 5
END
```

```
TO SQUARE
  REPEAT 4 [FORWARD 50 RIGHT 90]
END
```

Using these procedures the line

```
DRAW A RED SQUARE,
```

would produce the same result as with Harry's procedures. But lines like

```
RED SQUARE
```

and

```
A RED SQUARE
```

would also work.

A language microworld based on commands without inputs works well for single words, but not so well for phrases. On the other hand, linking words by using reporters and commands that require inputs may impose a structure that is more restrictive than we might want it to be. The challenge is to create a reasonable mesh between Logo and English.

A different approach to creating natural language microworlds is to write an interpreter. This puts a layer between the user and Logo where a Logo program reads lines that are typed and acts on them. Here's an example:

```
TO LISTEN
  TYPE "?"
  INTERPRET READLISTCC
  LISTEN
END
```

```
TO INTERPRET :THISLINE
  IF EMPTY? :THISLINE [STOP]
  IF MEMBER? FIRST :THISLINE
    [A THE AN DRAW MAKE DO]
    [INTERPRET BF :THISLINE STOP]
  IF MEMBER? FIRST :THISLINE
    [RED BLUE WHITE GREEN PURPLE]
    [SETC RUN (LIST FIRST :THISLINE)
     INTERPRET BF :THISLINE
     STOP]
  IF MEMBER? FIRST :THISLINE
    [SQUARE TRIANGLE CIRCLE]
```

```
[RUN (LIST FIRST :THISLINE)
 INTERPRET BF :THISLINE STOP]
(TYPE [I DON'T KNOW THE WORD]
 FIRST :THISLINE CHAR 13)
```

```
END
```

```
TO RED
  OUTPUT 2
END
```

```
TO WHITE
  OUTPUT 1
END
```

etc.

```
TO SQUARE
  REPEAT 4 [FORWARD 50 RIGHT 90]
END
```

```
TO TRIANGLE
  REPEAT 3 [FORWARD 50 RIGHT 120]
END
```

```
TO CIRCLE
  REPEAT 36 [FORWARD 10 RIGHT 10]
END
```

Writing your own interpreter overrides Logo. You can have your program accept words like MAKE which already have a meaning to Logo and would not be useable in the top level approach. LISTEN will print an "I don't know the word ..." message when an unknown word is encountered. The rest of the line will not be evaluated. In the top level approach, the regular Logo error messages will be printed and could be misleading to beginning readers.

On the other hand, the interpreter excludes normal Logo vocabulary. The top level approach does not prohibit using a FORWARD 100 whenever one wants. The interpreter doesn't know Logo unless you build in such knowledge.

The interpreter approach puts the programmer in touch with an area of computer science that is fundamental to Logo, writing an interpreter, a program that reads lines that you type and interprets their meaning. Logo itself is an interpreter. In fact, can you write Logo in Logo? Can you write BASIC in Logo? (Would you want to?)

Natural language microworlds provide a valuable learning environment both for the beginning reader using them and the programmer creating them. They have educational value at many levels: word and phrase recognition for the beginner, more sophisticated programming for the older student, and exploration of the nature of natural language for the student of any age. Why not present some aspect of this idea to your students of whatever age, and see what results!

Michael Tempel, Logo Computer Systems, Inc., 330 West 58th Street, Suite 5M, New York, NY 10019

Teaching Tools

Sound Concepts

by Glen Bull and Gina Bull

1. A group of children sit around a collection of bottles, creating different musical tones with an impromptu jug band.
2. Several children on the playground dash ten yards, touch an oak tree, and then run back to the starting line. A second group repeats the process with a starting line that is twenty yards away from the tree.
3. Children gather around a computer monitor, using Logo tools to interpret their first two experiences.

At first glance it might appear as though events (1) and (2) have nothing to do with one another, or with Logo. Yet there is a common thread that links all three. It is desirable to bring a variety of both computer-related and non-computer-related experiences to bear on a concept. In this column, a specific illustration of how this might be done will be examined.

Teaching begins with a concept or objective to be learned. In Virginia these goals are listed as Standard of Learning objectives. Here is a Standard of Learning objective for the fifth grade:

"The student will demonstrate the ability of sound to travel through solids, liquids, and gases and describe the wave nature of sound."

By the eighth grade the Standard of Learning objective has metamorphosed into the following:

"The student will investigate the basic characteristics and technological applications of mechanical waves."

Characteristics listed include sound wave motion, water wave motion, reflection, refraction, and interference, while technologic applications suggested for this objective include music, communication, and hearing.

Both the jug band and the foot race are related to the topic of sound and motion, and the objectives listed above. The reason for inclusion of the jug band may be evident, while the foot race may require some explanation.

Sound Waves

As a starting point, consider why some bottles produce a low tone while others produce a high note. Generally tall bottles will produce a deep tone, while short bottles will produce a high pitch. A tube provides a good laboratory for experimenting with some of the variables underlying this physical phenomenon. Sound is a back and forth motion. It takes a longer time for a sound to ripple down to the end of a long tube and back again. Therefore fewer back and forth cycles occur in a given period of time, in comparison with a short tube.

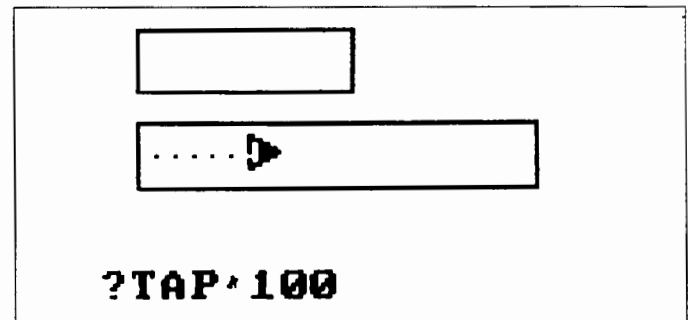
Several Logo tools will be used as building blocks to construct tools for exploration of sound and motion. The Logo TUBE procedure draws a rectangle which will serve as a tube, and places the turtle in the tube.

```
TO TUBE :LENGTH
  SETH 90
  PD
  RECTANGLE :LENGTH + 8 20
  OVER 10
  PU
  END
```

```
TO RECTANGLE :LENGTH :WIDTH
  REPEAT 2 [FORWARD :LENGTH RIGHT 90
    FORWARD :WIDTH RIGHT 90]
  END
```

```
TO OVER :DISTANCE
  PU
  RIGHT 90
  FORWARD :DISTANCE
  PD
  LEFT 90
  END
```

The tube drawn is the length specified, plus the length of the turtle (about 8 steps). Thus, the command TUBE 50 draws a tube 58 turtle steps long, while the command TUBE 100 draws a tube 108 turtle steps long.



Now that the turtle is in the tube, the procedure TAP can be used to tap the end of the tube, and send the turtle shuttling back and forth. The subprocedures STEP.FD and STEP.BK are used to ensure the turtle travels at a constant speed. In other words, the time required to travel 100 steps should be twice as long as the time required to travel 50 steps.

```
TO TAP :LENGTH
  STEP.FD :LENGTH
  STEP.BK :LENGTH
  END

TO STEP.FD :TIMES
  REPEAT :TIMES [FORWARD 1]
  END

TO STEP.BK :TIMES
  REPEAT :TIMES [BACK 1]
  END
```

When the procedure TAP is run, it produces a tap on the end of the tube, causing the turtle to slide down to one end of the tube, and then bounce back to the other end. The TAP procedure must include the length of the tube as an input.

```
TUBE 100
```