# LOGOS

# PRE-PROCEEDINGS OF THE 1984 NATIONAL LOGO CONFERENCE

Massachusetts Institute of Technology
Cambridge, Massachusetts
June 26 - 29, 1984

Sponsored by the
Laboratory for Computer Science
Massachusetts Institute of Technology

EDITING AND COMPILATION OF THE
1984 NATIONAL LOGO CONFERENCE PRE-PROCEEDINGS
BY RENATA J. SORKIN
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

# TABLE OF CONTENTS

# PREFACE

The first National Logo Conference will be held from June 26 through June 29, 1984 on the campus of the Massachusetts Institute of Technology, under the sponsorship of the MIT Laboratory for Computer Science. The *ad hoc* steering committee hopes that this will be the first of a series of annual conferences that will help foster communication within the Logo community. We have planned this first conference to be a relatively small meeting for people who are already familiar with Logo and actively involved with it. The conference will be centered around four panel presentations devoted to topics of particular interest to Logo workers: What do children learn from using Logo?; What kinds of learning environments are appropriate for Logo use?; Advanced programming with Logo; Extensions and new developments with the Logo language. Following the panel presentations, there will be opportunities for all conference participants to further pursue these topics in scheduled discussion groups. In order to help stimulate discussions, papers by the panelists on these topics are included in these preproceedings.

Thursday, June 28, will be devoted to talks, exhibits, and poster sessions that illustrate the wide variety of Logo work. A schedule of talks and exhibits is given below, together with abstracts of each presentation.

There will also be informal opportunities to share programs and ideas. In particular, clusters of the more popular microcomputers that currently run Logo will be available for conference participants to reserve for impromptu discussions, demonstrations, and program sharing. (Bring disks of your programs to share -- also blank disks on which to copy programs for personal use.)

The steering committee would like to thank all of the panelists, speakers, exhibitors, discussion groups leaders, volunteer helpers, and companies who have contributed to the success of the conference. Also, as chairman of the committee, I would like to especially thank: Greg Gargarian, for devising the overall structure of the conference and supervising its organization; Renata Sorkin, for managing the Herculean task of collecting papers, scheduling talks, and assembling and editing these preproceedings; Joyce Tobias, for coordinating contacts with industrial sponsors; and Tom Lough, for compiling the bibliography of Logo publications that is included here.

Hal Abelson

# GENERAL INFORMATION

## Registration

The registration fee for the conference is $125 if paid before May 25, 1984, and $165 if paid after that date. This fee includes admission to all sessions; one copy of the Pre-proceedings; the Reception on Tuesday, June 26; lunch on Wednesday and Thursday, June 27 and 28; and the Clambake on Thursday, June 28.

Registration at the conference will be held in the MIT Athletic Center Lobby on Tuesday, June 26, from 12:00 noon until 6:00 p.m. A registration/information desk will be located in the Athletic Center Lobby during the conference and will be available to conference participants, their families and friends for information and assistance.

## On-Campus Housing

Single and double dormitory rooms on the MIT campus will be available to conference participants from Monday, June 25 through Saturday, June 30, 1984. Dormitories are located along the Charles River, within walking distance to all conference facilities. All rooms are furnished with twin-size beds, bed linens, blankets, towels and soap; rooms are serviced each day. There are no private baths or air-conditioned rooms on campus. Dormitories are equipped with elevators, ice machines, and coin-operated laundry facilities. There are telephones located in each room allowing campus and local calls. Public telephones are situated in the lobby of each building. The dormitory desks are staffed from 8:00 a.m. to 1:00 a.m. daily.

Cots are available for children between the ages of 6 and 14, with a maximum of two cots per double room. Youths over the age of 14 must be accommodated in a separate room.

The dormitory rates are $28 per night for a single room, and $32 per night for a double room. Cots are charged at $4 per child, per night.

Prepayment for the anticipated number of nights is required. A full refund will be granted if cancellation is received two weeks prior to the start of the conference (June 12). No refunds will be made after the arrival for nights the rooms are not occupied, except for early departure if 24 hours notice is given.

Dormitory rooms can be reserved by sending payment by check or money order to MIT Special Events Office, Room 7-111, Cambridge, MA 02139.

## Off-Campus Housing

Blocks of rooms have been reserved for those participants who do not wish on-campus accommodations. Attendees should contact the hotel of their choice directly (see hotels listed below). Please state that you will be attending this conference when making your reservations. Availability and rate are not guaranteed after May 24, 1984. Room rates do not include a 5.7% tax.

Hyatt Regency Hotel
575 Memorial Drive
Cambridge, MA 02139
(617)492-1234
Single $80. Double $80.
The Hyatt is located on the Charles River within a 10-minute walk from MIT. Parking is available at the hotel (per day charge)

Hotel Sonesta
5 Cambridge Parkway
Cambridge, MA 02142
(617)491-3600
Single $70. Double $80.
The Sonesta is located on the Charles River, approximately 3/4 of a mile from MIT. This hotel has a complimentary shuttle service to the campus. Parking (free) is also available.

## On-Campus Dining

Lunch on Wednesday and Thursday, June 27 and 28, will be provided in the Athletic Center. Dinner on Thursday, June 28, will be served at Calder Court (see Special Events below).

Families and guests may use the Lobdell Cafeteria or the Walker Memorial Dining Hall (lunch only in Walker) for meals at their own expense. A dining guide to the many eateries in the Cambridge/Boston area will be available at the conference information desk in the Athletic Center Lobby.

## Transportation

Logan International Airport is approximately six miles from MIT. Taxi fare to the campus is about $12 regardless of the number of passengers. There is public transportation between the airport and MIT, however this involves a bus ride and three subway lines.

If you are arriving by train at Boston's South Station, take the MBTA Red Line to Kendall Square. Subway fare is 60 cents each way.

## Sightseeing

Cambridge and Boston offer a wide variety of daytime and evening activities boasting a unique combination of old and new. Faneuil Hall and Quincy Market afford many fascinating shops and restaurants. There are an abundance of eating establishments in the area offering dozens of different foods, including Boston's famous seafood.

Attractions of particular interest include the Freedom trail, the Museum of Fine Arts, the John F. Kennedy Library, the Museum of Science, the John Hancock Tower, Copley Place, the Isabella Stewart Gardner Museum, and Boston's historic waterfront. Just to the west are the famous battle roads in Lexington and Concord, featuring the Museum of Our National Heritage. In addition to the many musical events taking place in Boston, there are a number of fine theatres.

Information on area tourist attractions will be available at the information desk in the Athletic Center Lobby.

## Climate and Dress

New England's weather is notoriously unpredictable, but during June the weather in Boston is generally warm and pleasant. The average temperature during the day is 70 degrees Fahrenheit, but this can sometimes be accompanied by high humidity. A light jacket may be needed in the evening; rainwear is usually not necessary, however it would be advisable to come prepared just in case.

## Special Events

A welcoming reception will be held on Tuesday, June 26, from 4:00 to 6:00 p.m. in the Athletic Center. This reception is open to all conference participants and their guests.

On Thursday, June 28, the conference banquet will feature a traditional New England Clambake to be held at MIT's Calder Court (rain location Walker Memorial Dining Hall). Tickets for spouses and guests are available at $25 per person and may be purchased during registration on Tuesday, June 26, at the Athletic Center Lobby. The cost to conference participants has been included in the registration fee.

## CONFERENCE ARRANGEMENTS/QUESTIONS

Questions concerning any of the conference arrangements should be directed to the MIT Special Events Office, Room 7-111, Cambridge, Massachusetts 02139, Telephone (617253-1703).

# LOGO 84 - SCHEDULE OF EVENTS

## Tuesday, June 26

| | |
|---|---|
| ✓12:00-6:00 P.M. | Registration<br>Athletic Center Lobby |
| ✓4:00-6:00 P.M. | Reception<br>Athletic Center |
| ✓6:00-8:00 P.M. | INTRODUCTORY PANEL<br>Kresge Auditorium |

## Wednesday, June 27

| | |
|---|---|
| ✓9:00-10:30 A.M. | WHAT DO CHILDREN LEARN?<br>Kresge Auditorium |
| ✓10:30-11:30 A.M. | Discussion Groups<br>Student Center |
| ✓11:30 A.M.-2:00 P.M. | Lunch<br>Athletic Center |
| ✓2:00-3:30 P.M. | THE LEARNING ENVIRONMENT:<br>FORMAL AND INFORMAL<br>Kresge Auditorium |
| ✓3:30-4:30 P.M. | Discussion Groups<br>Student Center |
| 4:30-7:30 P.M. | Dinner |
| 7:30-9:00 P.M. | ADVANCED PROGRAMMING<br>Kresge Auditorium |
| 9:00-10:00 P.M. | Discussion Groups<br>Student Center |

## Thursday, June 28

| | |
|---|---|
| 9:00 A.M.-12:00 | Poster Sessions/Talks/Exhibits<br>Various |
| 12:00-2:00 P.M. | Lunch<br>Athletic Center |
| 2:00-5:00 P.M. | Poster Sessions/Talks/Exhibits<br>Various |
| 6:30 P.M. | Clambake<br>Calder Court |

## Friday, June 29

| | |
|---|---|
| 9:00-10:30 A.M. | EXTENDING LOGO<br>Kresge Auditorium |
| 11:00 A.M.-12:00 Noon | CLOSING PANEL<br>Kresge Auditorium |

# LOGO 84 - CONFERENCE PROGRAM

## 1. TUESDAY, JUNE 26

| | |
|---|---|
| ✓Registration | 12:00-6:00 P.M.<br>Athletic Center Lobby |
| ✓Reception | 4:00-6:00 P.M.<br>Athletic Center |
| ✓INTRODUCTORY PANEL | 6:00-7:30 P.M.<br>Kresge Auditorium |

Panelists:

Seymour Papert, *Massachusetts Institute of Technology*

Alan Kay, *Atari, Inc.*

Dan Watt, *Educational Alternatives*

. . . . . . . . . . . . . . . . . . . . . . 23

## 2. WEDNESDAY, JUNE 27

### WHAT DO CHILDREN LEARN? — 9:00-10:30 A.M.
**Chaired by William Higginson — Kresge Auditorium**

Panelists:

### Discussion Session — 10:45-11:30 A.M.

1) *Little Kresgė*
Moderators: Ann Berger and Robert Lawler

2) *Student Center Room 407*
Moderators: Uri Leron and James Milojkovic

3) *Student Center Room 491*
Moderators: Jeanne Bamberger and Jose Valente

4) *Student Center Mezzanine Lounge*
Moderators: Dale Burnett and Douglas Clements

### Lunch — 11:30 A.M.-2:00 P.M.
**Athletic Center**

## 2. WEDNESDAY, JUNE 27 (CONT.)

### THE LEARNING ENVIRONMENT: FORMAL AND INFORMAL — 2:00-3:30 P.M.

Chaired by E. Paul Goldenberg — Kresge Auditorium

### Discussion Session — 3:45-4:30 P.M.

1) *Little Kresge*
Moderators: Geraldine Kozberg and Glenn Fisher

2) *Student Center Room 407*
Moderators: Susan Jo Russell and Molly Watt

3) *Student Center Room 491*
Moderators: Tessa Harvey and Tim Riordan

4) *Student Center Mezzanine Lounge*
Moderators: Beth Lowd and Steve Tipps

### FREE — 4:30-7:30 P.M.

## 2. WEDNESDAY, JUNE 27 (CONT.)

### ADVANCED PROGRAMMING — 7:30-9:00 P.M.
Chaired by Margaret Minsky — Kresge Auditorium

### Discussion Session — 9:15-10:00 P.M.

1) *Little Kresge*
Moderators: Ursula Wolz and Wallace Feurzeig

2) *Student Center Room 407*
Moderators: Gary Drescher and Michael Eisenberg

3) *Student Center Room 491*
Moderators: Mark Gross and Ed Hardebeck

4) This group will split into two sections:

a) *Student Center Mezzanine Lounge*
Moderators: Larry Davidson and James Milojkovic

b) *Kresge Auditorium*
Moderators: Jim Davis and Eric Solomon

## 3. THURSDAY, JUNE 28

### Poster Sessions and Talks 9:00 A.M. - 12:00

(SEE P.155 FOR POSTER SESSION AND TALK ABSTRACTS)

POSTER SESSIONS
*Lobby of Kresge Auditorium*

9:00 - 10:30 A.M.:

| | |
|---|---|
| *Integrating Mathematics and Computers (Logo Language) with Science Activities*<br>Lyle Andersen and Gilbert Blankespoor | *Table 1* |
| *Logo: Tricks or Topics*<br>Fred Achberger | *Table 2* |
| *Logo: A Mirror for Learning Personalities*<br>Nicole Michaud | *Table 3* |
| ✓*Training Teachers to Use Logo*<br>Glenn Fisher | *Table 4* |
| *Use of Logo in the Teaching of French*<br>Lorne Bouchard and Louisette Emirkanian | *Table 5* |
| *Color Logo Animated Film Production*<br>Chris Templar | *Table 6* |

10:30 - 12:00 Noon:

| | |
|---|---|
| *Plotting with Logo*<br>Steve Tipps | *Table 1* |
| *Logo-Based Job Training for Inner-City Adults*<br>Vicki Carver | *Table 2* |

## 3. THURSDAY, JUNE 28 (CONT.)

| | |
|---|---|
| *Logo and Physics*<br>David Briskman | *Table 3* |
| *Logo Explorations in Language and Algebra*<br>Allison Birch and Larry Davidson | *Table 4* |
| *Friends of the Turtle*<br>Sandra Crowther and Michel Eltschinger | *Table 5* |
| *Teaching Structured Logo*<br>Reinhold Wappler | *Table 6* |
| *Investigating the Effect of Age and Cognitive Style on Children's Intuitions of Motion Using Concrete and Computer Tasks*<br>Andy diSessa and Tamar Globerson | *Table 7* |

### TALKS

*Various Locations*

9:00 - 9:45 A.M.:

| | |
|---|---|
| *Math and Science Investigations Using Logo*<br>Technical Education Resource Center | *Stud.Ctr.Mezz.Lnge.* |
| *Logo Effects in Public School Classrooms*<br>Peter Fire Dog | *Stud.Ctr.Rm.491* |
| *Introduction to List Processing Through Fantasy*<br>Jim McCauley | *Bldg. 4 Rm. 270* |
| *Intervention Strategies and Collaboration in Learning Logo*<br>Celia Hoyles and Rosamund Sutherland | *Bldg. 4 Rm. 231* |

10:00 - 10:45 A.M.:

| | |
|---|---|
| *Creating a Logo Culture a la Monadnock Logo Users' Group*<br>Molly Watt, Dan Watt and Tony Stavely | *Stud.Ctr.Mezz.Lnge.* |

## 3. THURSDAY, JUNE 28 (CONT.)

*Mathematical Concepts and Programming Skills Acquired by Eight-Year-Olds in a Restricted Logo Environment*
J. Hillel — *Stud.Ctr.Rm.491*

*Languaging Through Logo*
C. Roxanne McDiarmid — *Bldg. 4 Rm. 270*

*Logo and the Reality of Elementary Classrooms: A Report on the "Creative Uses" Project at Queen's University (1982-1984)*
J. Dale Burnett and William Higginson — *Bldg. 4 Rm. 231*

11:00 - 11:45 Noon:

*Logo as a Part of an Elementary Teacher's Preparation*
Janice L. Flake — *Stud.Ctr.Mezz.Lnge.*

*Effects of Logo Programming on Cognitive Style and Cognitive Development*
Douglas H. Clements — *Stud.Ctr.Rm.491*

*Learning Language with Logo*
Wallace Feurzeig and E. Paul Goldenberg — *Bldg. 4 Rm. 270*

*Learning and Logo: Collaborative Research in the First Grade*
Judith Kull, Joyce Shea Strong and Bernard Cohen — *Bldg. 4 Rm. 231*

Lunch — 12:00 Noon - 2:00 P.M.
Athletic Center

## 3. THURSDAY, JUNE 28 (CONT.)

Poster Sessions and Talks (cont.) 2:00 - 5:00 P.M.

### POSTER SESSIONS

*Lobby of Kresge Auditorium*

2:00 - 3:30 P.M.:

*Interdisciplinary Logo* — *Table 1*
Suzanne Chapin and Susan Holden

*Where Are the Microworld Designers?* — *Table 2*
David Andrew

*Talking with Logo: Logo in Speech, Hearing and Language*
Glen Bull — *Table 3*

*Logo in Malaysia* — *Table 4*
Dennis O. Harper

✓*Logo Training: Some Experiences and Recommendations for Change* — *Table 5*
Michael Tempel, Harry Nelson and Nicole Michaud

*Logo as a Medium for Creating Special Effects* — *Table 6*
Dan Suttin

3:30 - 5:00 P.M.:

✓*Teacher Workshops: Examples of Workshop Challenges and Teacher Creations* — *Table 1*
Technical Education Resource Center

*A Logo Authoring System* — *Table 2*
Eric Brown

*List Processing Tools* — *Table 3*
Tony Stavely

## 3. THURSDAY, JUNE 28 (CONT.)

*Logo as a Tool for Studying Physics* — *Table 4*
Evelyn Dale

*Advanced Logo and Artificial Intelligence* — *Table 5*
Jeff Haas

*Logo at Punahou School* — *Table 6*
Elaine Blitman

✓*Building Bridges from Logo to School Mathematics* — *Table 7*
Temple Arey and Sylvia Weir

### TALKS
*Various Locations*

**2:00 - 2:45 P.M.:**

*Perspectives on Turtle Graphics*
Brian Silverman — *Stud.Ctr.Mezz.Lnge.*

✓*Logo and Educational Change*
Geraldine Kozberg — *Stud.Ctr.Rm.491*

*Ten Steps to Creating a Microworld*
Molly Watt and Dan Watt — *Bldg. 4 Rm. 270*

*Computer-based Environment for the Handicapped*
Jose Armando Valente — *Bldg. 4 Rm. 231*

**3:00 - 3:45 P.M.:**

*NACCIS Performance Methodology Project*
Steve Louie and Judy LeFevre — *Stud.Ctr.Mezz.Lnge.*

## 3. THURSDAY, JUNE 28 (CONT.)

| | |
|---|---|
| *The Senegalese Project: Computers in Education* | |
| Fatimata Seye Sylla | *Stud.Ctr.Rm.491* |
| *The Logo Microworlds Project at OISE* | |
| Rina Cohen | *Bldg. 4 Rm. 270* |
| *The Aesthetics of Logo and Instruction in the Arts* | |
| Pamela Sharp | *Bldg. 4 Rm. 231* |

**FREE** **5:00 - 6:30 P.M.**

**Clambake Dinner** **6:30 P.M.**
**Calder Court**

## 4. FRIDAY, JUNE 29

### EXTENDING LOGO — 9:00-10:30 A.M.
**Chaired by Cynthia Solomon — Kresge Auditorium**

### CLOSING PANEL — 11:00 A.M.-12:00
**Kresge Auditorium**

Seymour Papert, *Massachusetts Institute of Technology*

Alan Kay, *Atari Inc.*

Marvin Minsky, *Massachusetts Institute of Technology*

# PANELIST PAPERS

# CREATING LOGO CULTURES

Dan Watt
Educational Alternatives
and
Popular Computing Magazine

Being involved with Logo in 1984 is a bit like riding a roller coaster. Moving from the hothouse environment at MIT where it was nurtured by a few people for more than a decade, it is now available to millions of people, in homes, schools, camps, libraries, even on television. Those of us who had the good fortune to be part of its formative years are extremely gratified to see Logo's widespread acceptance and use all over the country and in many other parts of the world. But like a ride on a roller coaster, things are starting to happen fast and are getting a little scary. Having ridden to the top, the downward plunge may be just a bit more exciting than we had expected.

Logo's success brings with it challenges that must be honestly faced and dealt with if we are to realize our vision of computer-based learning environments that extend and enhance our humanness. What I want to do in this talk is confront the issues that face us today as we work to create a culture that supports this kind of approach to learning.

## Logo Without Culture

I think of the major challenge we are facing today as a problem of culture. Consider the following analogy.

Suppose you are a third grade teacher who is expected to teach reading and writing without knowing how to read and write yourself. In a week-long summer workshop, you will be taught the skills that your students are supposed to master during the coming year. If you're lucky, you'll also be provided with some reading books, writing materials, and worksheets. If not, you'll have to create them all yourself during the course of the year.

If we tested your students after the first year, we'd probably be disappointed by the results. We might find that some of the kids enjoyed reading and writing, while others hated it. Skill levels probably wouldn't be very high (by current third grade standards), and it wouldn't be easy for kids to use reading and writing on their own very well. I doubt that very many kids would be reading independently, or publishing newspapers, or would have much sense of what to do in a library.

In our society, on the other hand, most teachers are remarkably successful in teaching (inspiring, helping, guiding, cajoling, coercing, etc.) most kids to read and write and to use reading and writing for their own purposes. In part, this is because we have a very well developed technology for teaching literacy skills. Textbooks, lesson plans, readers, spelling books, paperback collections with items for every taste and reading level, and lots of different kinds of writing activities and assignments undoubtedly play a role in creating successful readers and writers.

Even more important, I believe, is that teachers and children, as well as parents, are steeped in a culture of literacy. Long before they enter kindergarten, children in our society grow up in a world in which most people read and write every day. Everyone in their world values and uses literacy, and promotes its use with children. Teachers in particular, are self-selected carriers of the culture of literacy. All teachers know how to read and write fluently before they ever come to teach. They've read at least some of the classics, and they read the daily newspapers. And every day they write: notes, letters, student reports. Some of them might even write articles, poems or stories from time to time.

The bottom line is, children grow up literate because they've got a literate world to grow up in.

Now let's look at what's happening with Logo. During the past two or three years, classrooms all over the country have begun to "teach" Logo to students, primarily in grades 3-6. Teachers, whose predilections may make them excited, indifferent or terrified, are typically given from one to five days of prior training, much of which consists of learning to operate the computer, use the disk drive and editor effectively, and "experience what the children are supposed to learn," during the course of the following year.

In many situations, teachers are told that they must teach Logo in their classrooms whether they want to or not. Depending on the philosophy of their trainers, they may be encouraged to let children explore freely and invent their own problems, or they may be given a specific set of lesson plans: "First lesson, draw a box; second, a triangle; third, put them together to build a house;" etc. Either way, their condition is not much different from that of the third grade teacher I was talking about earlier. They are expected to teach without a culture.

Most teachers teaching Logo today have never used a computer before. They have had only a tiny amount of Logo experience themselves during the preparatory workshop. They have been exposed to very few ideas about what Logo can do. Often they have no one to talk to about what they are doing, and no ongoing support structure. Because "computer literacy" is the hot new program being heavily promoted within the schools, they are under intense public scrutiny. And because there is a prevailing myth that Logo is supposed to be fun, easy and natural for both teachers and students, they are expected to be "successful." ("There's no such thing as failure in Logo," is a commonly heard slogan. The heck there isn't. Just ask anyone who hasn't been able to get a stop rule to work, or ask any teacher whose students spend their entire computer sessions making the turtle wrap endlessly around the screen in a hodgepodge of colors.)

## A Culture is Growing

Fortunately, the rather gloomy picture I've just described, while more commonplace than I'd like it to be, is far from the whole truth. There are exciting things happening as well, sometimes in surprising places. Logo was designed to be embedded in a rich culture of activities, ideas, ways of describing things and thinking about them, and especially of people, collaborating to create a supportive learning environment. And although it's understandable that much of the first few years' effort has gone into mastering the mechanics of the language and the computer system, pockets of culture are emerging in ways that show that Logo can indeed serve as a catalyst and an organizing principle for the kind of human culture we are all trying to create.

Let me mention a few examples:

* Thoughtful Logo training courses that incorporate the broader Logo culture as well as teach the mechanics of the language, are beginning to be taught at a number of colleges and universities. Some of these courses are now going beyond the simplest possible uses of Logo. And their graduates are spreading what they've learned to their students and colleagues.

* At least one researcher I know (and I'm sure there must be many others) is working with teachers to observe what students are actually doing with Logo, in ways that inform and support the teaching, as well as provides honest information to the community at large. And the research is finding that with committed, thoughtful, supportive (and supported) teachers, observations of the kids provide evidence that powerful learning is going on.

* Some school districts provide real ongoing support for their

teachers, providing them with regular opportunities to learn new ideas, share with fellow teachers, voice frustrations and move further in their own thinking.

* Logo is being used in many settings other than schools. Informal learning centers in homes, store fronts, community centers and camps, are places where the culture is developing without the restrictions imposed by formal educational institutions. One problem for people in these settings is one of communication with colleagues.

* Users groups have been meeting for two years in some cases: large ones such as the one in Boston which offers a forum for relatively formal contact between users at all levels and some of the most important ideas in the Logo culture; small ones, such as the Monadnock Area Logo Users Group in southwestern New Hampshire, which has shared ideas informally and deeply among a widening circle of Logo enthusiasts.

* Newsletters have become a regular vehicle of exchange for people from all parts of the country. And at least one educational magazine provides regular coverage of Logo. As a result, the idea that Logo has much more to it than just things that are "easy" for little kids, is gaining wider currency.

* Published books, curriculum guides, and activity cards of all kinds and qualities are starting to appear in response to a broad demand from educators. While most of these are still concerned with mechanics and show little connection with the broader Logo culture, there are now a few publications that are delightful, thoughtful, and provide access to different aspects of the culture. (It's our job as carriers of the culture, to support the dissemination of the quality materials, and let the others fall into oblivion. It's no longer useful--if it ever was--to take the position that any Logo book is a good Logo book.)

## Beyond "Training" and "Curriculum"

This conference is a most encouraging development. As a gathering of Logo activists, it gives us the opportunity to meet each other and build a real community of people who share a vision, to deepen our thinking, to get a clearer sense of what our next steps are, and to create an ongoing support system. Most of all, it gives us the opportunity to create a quantum leap in the quality and persistence of the Logo culture.

The criticisms of Logo also come at an advantageous time. It gives those of us who believe in its importance an opportunity to examine our commitment as carriers of the culture. We have a chance to confront our own roles in perpetuating myths and misconceptions that are beginning to interfere with the growth of the Logo movement.

Some of the myths we ought to confront: Logo is primarily for young children. Logo is identical with turtle graphics, and other aspects are not interesting or important. Logo is easy to learn. Logo is hard to learn. Logo should be learned by children interacting with computers in a "free discovery" mode, without teachers or support materials. Logo can't be evaluated because it's so "innovative and different." Logo is the ultimate computer language and environment, good for everything that might ever be useful to do in an educational setting (the "no threshold, no ceiling" slogan, taken to absurdity). I'm sure you can identify many more.

Logo is a culture and a movement. But it's not a religion or a political party. There are no deities or priests, no dogmas, creeds or party lines. Logo can encompass many styles, possibilities and directions, including ones that move beyond Logo in a number of ways. What unites us is a vision of what learning environments could be: rich with experiences, ideas and people that break down the barriers between disciplines, between teacher and learner, and between people with different learning styles. It is the excitement of this vision, and the potential embodied in Logo to create this vision that brings us all together in this conference. I look forward to the opportunity that this conference represents.

# ABOUT THAT ROSE GARDEN: REMARKS ON LOGO, LEARNING, CHILDREN AND SCHOOLS

William Higginson
Queen's University at Kingston

Education is concerned with two worlds: the world that man lives in and the world he wants to live in.[1]

Space constrains and suggests a mixture of styles. We proceed from the fabulous to the telegraphic and then to the reportial. The thesis is that claims about what children learn or do not learn from Logo should be scrutinized carefully. In school settings, which very seldom provide the sort of atmosphere envisaged in *Mindstorms*, the teacher is seen as a critical factor in determining the success of any Logo venture. Evidence of significant social and intellectual experiences of a Papertian type with children using Logo in standard classroom settings is reported from the Queen's "Creative Uses" Project.

## The Little Engine that Might: A Techno-Fable for Our Times

Once upon a time in a northern land not so far away, a learned and caring professor became concerned about the physical health of children. Disturbed by the elitism, expense and competition of organized sports in schools and the general lack of support for physical activity in the culture he wrote a book. In this book, called *Bodysqualls*, he decried the existing situation and argued passionately for the virtues of an activity which he had come to love, cross-country skiing. This activity, he claimed, was good for lungs, legs, arms and attitudes. It made it easy to commune with nature; in short, a balm for body and mind. Although immediately accessible to beginners, it could be among the most demanding of sports for the experienced. And *Bodysqualls* became, as often happened in that society, an overnight success. In many a smoky staffroom its praises were sung. The Orson Welles School of Ballet made cross-country skiing a compulsory part of its curriculum. Raffles were run to buy schools a pair of cross-country skis. Sometimes even several pairs; almost always the no-wax type because the other kind was complicated and messy. And whenever it snowed (which was fairly often in that part of the world) teachers,

[1]Northrup Frye (1967, p.76)

sometimes experienced skiers--usually of the alpine or water variety--let their pupils ski around the playground all day.

And then, as often happened in that society, disillusionment set in. In most schools after a whole term of cross-country skiing average scores on the national fitness achievement tests, as always in paper and pencil form, were just as low as before. Researchers showed, with only the least shadow of a doubt, that children who had had cross-country skiing were no better at riding unicycles than children who had not. And in the whole land never was there found a single child who was able to do a telemark turn.

And in the smoky staffroom some teachers said that they knew all along that it wouldn't work. Others said that since they had the skis anyway they were going to use them as a reward since the children seemed to like cross-country skiing. At the Orson Welles School of Ballet a raffle was organized to raise money for motorized golf carts --otherwise all that walking makes one so sweaty. And at one of the most famous universities of the land, experts gathered to discuss what children learned from cross-country skiing. And everafter happily anyone hardly lived.

* * * * * * *

**Question:** What do children learn from Logo?
**Answer:** [1] It depends.

> It was a long while before it was recognized, even by Dewey himself, that the form of progressive education seized upon by the emerging profession was a bastard version, and in important ways a bastard version, and in important ways, a betrayal, of the new education he had called for.[1]

The response begs the question, "depends on what?". To which the equivocal answer has to be, "on a number of things". Foremost among these is the atmosphere or culture in which children encounter Logo. Here we meet the first of a number of difficulties. The mechanics of research together with the pattern of access to Logo have together determined that the great majority of research studies have been school-based. The fact that few schools subscribe to the epistemological and pedagogical

[1]Diane Ravitch (1983, pp.46-47)

principles on which Logo, which is first and foremost a philosophy of education, is based makes this exercise a bit like asking for reactions to *Das Capital* from chapters of the John Birch Society. Nor is it the case that philosophical congruence by itself will guarantee positive results. Logo is not a serum, nor a magic bullet. It is, in all cases, mediated by some "experienced" individual who acts as the "yeast" for the culture. In most all cases so far this role has been played by a teacher. Teaching is a complex and demanding enterprise which, if it is to be done successfully, demands commitment, sensitivity, stamina and a wide range of intellectual, social and organizational skills. Very few elementary-school teachers have the scientific/mathematical background to appreciate fully the potential of Logo. (There has been considerable overemphasis on the "no floor" side of Logo at the expense of its "no ceiling" characteristic which in the long term is much more significant). Nor, without considerable personal sacrifice, do they have the time to learn about the higher levels of the language. At the next stage of instruction one often finds an isomorphic situation where courses for teachers show many of the same weaknesses as courses for children.

In short, a situation where a powerful and sophisticated instrument has been put into the hands of people who are not, in most cases, for a number of reasons, well situated to use it effectively. One can use a Porsche to pull a plough or to deliver milk but one shouldn't be surprised if it isn't particularly efficient for those tasks. The current situation with Logo is akin in many ways to the remark Gandhi is once reputed to have made about "western civilization;" "It sounds like a fine idea, perhaps someone should try it some time."

**Question:** What do children learn from Logo?
**Answer:** [2] It's not easy to say.

I know that you, ladies and gentlemen, have a philosophy, each and all of you, and that the most interesting and important thing about you is the way it determines the perspective in your several worlds.[1]

[1]William James (1978, p. 9)

This second response also requires explanation. The root of the problem is that classical research methods in education are predicated on a neo-behavioristic model of knowledge and intelligence which simply is not appropriate for the study of Logo. Nor is it the case that Logo is somehow an exception here since the quantitatively-driven, positivistic methodology is equally limited in most other areas as well. (Robert Kennedy once observed in connection with measures of the Gross National Product that we could measure everything except those things which are worth measuring). No one would dispute the fact that, "What happens when you inject 'x' units of substance 'y' into the blood stream of individual 'z'?" is a very different sort of question from, "What happens when individual 'p' is exposed to language 'q'?" Despite this, much of the discussion of what children learn from Logo is carried out as if the question were of the first type rather than the second. (The situation has elements akin to the old story of the drunk looking for his keys underneath the lamp post; not because that was where he lost them, but because the light was better there). Logo, both because of its strong connections to Piaget's constructivist theories and the relative paucity of other exemplars, particularly in education, has become one of the major contemporary battle-grounds for an old philosophical dispute. (William James' distinction[1] between "tender-minded" and "tough-minded" temperaments, made almost eighty years ago, fits many aspects of today's situation rather well). Workers in other fields traveling along parallel paths include some members of the women's movement, most notably Carol Gilligan (1982), the "Aquarian Conspirators," to use Marilyn Ferguson's (1980) phrase and, perhaps most surprisingly, the managers of America's best-run companies. The case for the last contention is made quite forcefully by Peters and Waterman who state at one point, "the old rationality...has ceased to be a useful discipline."[2]

Northrup Frye once observed that, "knowledge is not something one has, it is something one is."[3] This is a view which is consistent with the philosophical underpinnings of Logo. "Itemizing possessions" is a much simpler task than "comprehending being". This makes studies of Logo learning experiences which are consistent with the philosophy of Logo much more difficult than ones which are not. But they are not impossible to do. To the extent that they are naturalistic, long-term, broadly-focused,

---

[1]William James (1978, p. 13)

[2]Peters and Waterman (1982, p. 42)

[3]Northrup Frye (1967, p. 59)

qualitative and carried out by researchers familiar with the settings, such studies are more anthropological than analytic in character.

> Children only learn well of their own accord. ... When children are only dragged, they learn to cope with a curriculum of intimidation. To teach, then, is to have the authority of the guide, of one who shows the way. To teach is not to drag, though some element of the imperative mood is always latent in teaching; the guide who shows the way is never merely permissive. ... Good teaching is above all a preparation for the unforeseen, for the lovely things that can happen when one has faith that they will happen.[1]

From the perspective of an educator and philosopher like Hawkins, the intent behind the question, "What do children learn from Logo?" might better be expressed in the form, "I is it the case that Logo can be used to aid the intellectual and social growth of children in the manner described by Papert (1980)?". On the basis of a nearly-completed two-year study (Burnett et al., 1984) of a number of elementary classrooms in Eastern Ontario, the answer in some of the settings is clearly positive. (A more detailed report on that project will be given later in this meeting; Burnett, Higginson, 1984). It seems that the factors which most influence the benefits that children get from working with Logo are largely related to the teacher. Especially important are her educational philosophy, pedagogic style and level of understanding of Logo. In those cases where the teacher's views about the purpose and preferred procedures of education are congruent with those of Papert, Logo has proven to be a very powerful tool for the development of significant learning. (It is, however, the case that, unfortunate as it may be, in the general teacher population these positions are quite rare.) At a macro level there appears to be two different types of learning occurring. The first, and perhaps the more obvious, is social. In those classrooms where sharing and discussing ideas is given considerable emphasis, we observe children learning to cooperate, to listen, to be critical in a constructive fashion, to appreciate the work of others and to see themselves as capable and responsible intellectual agents. On the academic side at a specific level they are learning to use implicitly concepts such as variable, coordinates and angle. (This is not to say that they would either be able to articulate these ideas in a rigorous form, or that they would be able to recognize them in another context. The neo-Piagetian position of researchers like Donaldson (1978) seems appropriate here.) More generally they are learning about ideas like

[1]David Hawkins (1983, pp. 65, 74)

modularity and debugging and the challenge, frustration and satisfaction that can come from the creation of an intellectual artifact.

At this point it seems appropriate to close with a remark made by a pair of ten-year old girls from the Queen's study who struggled with and finally successfully completed a one-variable procedure entitled "Windmill." The comment serves both to capture the spirit of Logo when used by a sensitive and knowledgeable teacher and, with the necessary change of tense, as a statement of the challenge to those of us who are committed to bettering the educational experience of children.

This was hard to do but it had to be done.

## References

Burnett, J. Dale et al. *A Multisite Evaluation of the Creative Use of Microcomputers Elementary School Children: A Research Report*. Toronto: Ministry of Education, 1984 (forthcoming).

Burnett, J. Dale and William Higginson "Logo and the Reality of the Elementary School Classroom." A paper presented at Logo '84, MIT, June, 1984.

Donaldson, Margaret *Children's Minds*. London: Fontana, 1978.

Ferguson, Marilyn *The Aquarian Conspiracy: Personal and Social Transformation in the 1980's*. Los Angeles: Tarcher, 1980.

Frye, Northrop "The Instruments of Mental Production," 59 - 83 in Wayne C. Booth, Ed., *The Knowledge Most Worth Having*. Chicago: University of Chicago Press, 1967.

Gilligan, Carol *In a Different Voice: Psychological Theory and Women's Development*. Cambridge: Harvard University Press, 1982.

Hawkins, David "Nature Closely Observed," 65 - 89 in Daedalus 112 (2) Spring, 1983. (Special Issue on "Scientific Literacy.")

James, William *Pragmatism and The Meaning of Truth.* Cambridge: Harvard University Press, 1978.

Papert, Seymour *Mindstorms: Children, Computers and Powerful Ideas.* New York: Basic, 1980.

Peters, Thomas, J. and Robert H. Waterman Jr. *In Search of Excellence: Lessons from America's Best-Run Companies.* New York: Harper and Row, 1982.

Ravitch, Diane *The Troubled Crusade: American Education 1945-1980.* New York: Basic, 1983.

# SO, WHAT DO CHILDREN LEARN WITH LOGO?

Rina S. Cohen
Ontario Institute for Studies in Education

The following dialogue has two participants: **E** - an educator, and **M** - a mathematician.

**E:** So, what do children learn with Logo?

**M:** Uhm.. I would say that this depends on a number of factors.

**E:** Such as?

**M:** Such as the learning environment, the extent to which it is structured, the type of guidance provided to the students, who provides the guidance, the attitudes and individual characteristics of both the teachers and the students. Also it depends on the kind of activities they are engaged in...

**E:** Wait a minute. I thought we are discussing Logo activities.

**M:** Yes, but Logo can take on many forms. Turtle Geometry seems to be the most common type of application today. But there are so many other microworlds, such as Multiple turtles and Sprites, Dynaturtles, List processing, Language labs based on word processors plus utilities, or combinations of the above. There are still many other microworlds as yet to be invented.

**E:** So which microworlds are we going to discuss?

**M:** Let's assume for now that we only deal here with Turtle Geometry and word processing applications, which are the most prevalent uses today.

**E:** So, where do we start?

**M:** We could start by discussing the learning environments and the teachers. I suggest that we include in our discussion only those typical Logo environments in the regular school setting that are mushrooming all over the country. Particularly, let's refer to those elementary school teachers who had caught the "Turtle Fever".

**E:** The typical Logo teacher inflicted with the Turtle fever has chosen to teach Logo because she believes in the educational philosophy. She has probably practiced this philosophy within the limits of the regular classroom also in the pre-Logo years, using a variety of concrete materials. She is enthusiastic about Logo and carries this enthusiasm with her into the classroom. She is caring, supportive and non-judgmental toward her students and knows when to step in for help and how to provide the appropriate kind of guidance. The general atmosphere in the class is that of sharing, collaboration and mutual help. Don't you think this description fits beautifully most of the Logo teachers we have met so far?

**M:** Most of them, yes, but certainly not all of them. In fact, we personally know two cases of teachers on whom the use of Logo was imposed by the board. They were each given a quick "marathon" Logo workshop so they could start teaching Logo shortly afterwards in order for their classes to participate in some Logo study conducted by the board. Since both teachers came from the structured, teacher centered classroom tradition, the resulting Logo environments didn't do much good for the Logo studies, to say the least.

**E:** In fact, they were rather sad to watch, as I recall.

**M:** So, obviously the students of these two teachers will not benefit from their Logo experience nearly as much as the students of the other more "authentic" Logo teachers. For instance, we cannot expect these students to learn the "debugging philosophy" and acquire a relaxed and constructive attitude toward errors, while their teacher finds it hard to accept her own mistakes as well as theirs.

**E:** Unfortunately you are right. Why not just leave out such teachers from our discussion on learning outcomes of Logo and consider only those fully committed Logo teachers who have chosen to teach Logo in the first place?

**M:** This would be fine with me, so long as we don't try to carry out a systematic evaluation of the effects of Logo on the students.

E: I don't quite get it.

M: Well, let's consider all these educational claims made for Logo, e.g. that extended Logo experience enhances thinking and problem solving skills, creativity, self concept, love of learning, etc. Suppose we want to validate some of these claims empirically. We would then need to design a controlled experiment...

E: But wait a minute. Who do you think would want to validate such claims empirically?

M: Ministry of Education, school boards and their research departments. They need such information.

E: But why? Isn't there a fair amount of anecdotal evidence supporting these claims? Besides, Logo IS JUST TOO GOOD TO BE EVALUATED. I'd rather leave Logo to grow and develop on its own. We don't want these bureaucrats to interfere.

M: I disagree with you on that. Do you remember the meeting we had last fall with some large school board administrators and research officers regarding our Logo project work?

E: Yes certainly. In fact, one of those people was in charge of the whole computer budget of the board.

M: Exactly. And they told us in that meeting that they were planning to conduct a quantitative research study on the effects of Logo on children. They were wondering if we would be interested in participating.

E: Of course we weren't.

M: Well, actually, at that time I wasn't quite so sure. You see, it took me some time to understand, at least partially, how such large school boards usually operate and make decisions. They really needed some "hard evidence" in order to justify increased expenditure on Logo programs. They explained to us that such hard evidence should preferably refer to some actual learning outcomes associated with Logo.

E: What kind of learning outcomes did they have in mind?

M: They said they were looking specifically for some outcomes related to the board's curriculum objectives, and were wondering if, by any chance, such information had been produced by any of the previous research studies. Of course, I told them I don't know any.

It just so happened that the same question came up again in two other meetings with educational administrators later that week. In both meetings, I was delicately reminded that actually, there still wasn't any concrete evidence verifying the learning outcomes of Logo that would justify a substantial increase in the budget for Logo-based facilities. I was finally convinced that such a study is necessary

E: Actually, I am getting convinced too. Just think of all those teachers out there who are so excited about Logo that they can hardly wait to use it in their classes, if only they could get hold of the necessary equipment.

M: Though, this "turtle fever" frightens me (as it frightens other people). Caution should be exercised in giving out such equipment to inexperienced teachers. They might set their hopes too high and then get frustrated when things don't work out the way they expected.

E: Maybe we should just make sure they get extensive training from experienced Logo teachers. At any rate, tell me what happened with respect to that study on the effects of Logo. Did you decide to collaborate with board researchers?

M: In a way, yes. Actually, they were planning to evaluate the effects of Logo on self concept in a controlled experiment, using the board's self concept instruments, and our seven Logo sites as experimental sites against seven corresponding control sites. As a matter of fact, such a study is already underway for the current academic year.

E: This pleases me a lot, because I have met all these Logo teachers from the experimental sites and believe that they are all superb, fully committed teachers. Their individualized approach and warm attitudes towards their students will surely enhance their students' self concept more than the average classroom teacher. So the experiment is very likely to produce positive results.

M: This is exactly the point. It would seem that the results of this experiment would have been positive even without computers and Logo teachers, like most Logo teachers across the country, do not in any way represent a typical cross section of the teacher population. This is why I chose to concentrate my efforts on trying to measure certain cognitive benefits that are less likely to be dependent on the teacher's personal characteristics or approach to teaching.

E: So what exactly did you end up doing?

M: I have collaborated in developing instruments for measuring specific cognitive skills such as map reading, directionality plus some math concepts at the primary level, that seem most likely to be affected by the initial stages of Turtle programming. The instruments are still at the testing and revision stage and would have to be validated before they can be used in any systematic evaluation study.

E: Somehow, this leaves me with an uneasy feeling. For me, all these "cognitive benefits", as you (and Pea & Kurland) call them, are only side benefits to the more important learning that Logo has to offer. The real benefits, in my eyes, are Logo's potential to foster in the child, through deeply meaningful syntonic learning experiences, a love for learning, a sense of mastery and power, a constructive attitude toward errors and willingness to take risks and experiment, a sense of sharing and cooperation with peers and with the teacher, and generally, a positive attitude towards school. All other learning outcomes, including intellectual development, will be a natural result of the above.

M: It makes a lot of sense, because a child who enjoys all these educational benefits is very likely to develop cognitively to his fullest potential.

The only problem with these benefits is that they represent only the *potential* of Logo, but not necessarily the *reality*.

E: But remember we have agreed to consider in this discussion only the elite group of Logo will be fully realized.

M: Not necessarily. Even with the best of Logo teachers, problems and undesirable effects will arise. Children don't learn nearly as much as was expected.

E: What makes you so sure?

M: In fact, such results have been reported by several recent studies, including the Bank Street College, Queen's University and University of Haifa studies. But we have also noted them in our own observations.

E: Do you mean, in that grade two classroom?

M: In that classroom and also in others. One major difficulty I see is inadequate support for the individual student. A single teacher in a class with thirty students simply cannot provide the amount of help needed for the children to be successful and make progress in their Logo work. This is particularly true for primary grade children who need a certain amount of hand holding, especially at the beginning stages of learning Logo. But often older children also get "stuck" in a corner unable to proceed unless somebody is available to get them "unstuck".

E: True, and if nobody is available at the time then they will end up feeling frustrated and helpless. They will have lost control.

M: Another common source of frustration is the tendency of some children to undertake projects that are beyond their ability level. Again, the absence of a knowledgeable adult to look over their shoulders and provide guidance at the right time might be crucial.

E: Yes, we have seen this happen on numerous occasions. And as the child repeatedly experiences this sense of frustration, helplessness and lack of control, he will gradually give up on any further attempts to understand what he did or to master the skills involved. Such a child has learned that he cannot succeed in the Logo environment.

M: I am not so sure about that. Actually, many such children soon learn that mini-course called "How to be Successful in a Logo Environment Without Really Understanding What You Are Doing". Some of the "golden rules for instant programming" include:

* If in doubt, produce a long REPEAT command.

* Write a "random procedure".

* Copy a procedure from a friend or from your own diskette, and add or delete a few commands. You may also wish to use any of the well known "idioms" that you have previously memorized.

* Copy an impressive graphics procedure out of a book or Logo magazine...

E: Can you please stop this right away? You are so cynical. Actually, I recall that most children did try, at least for part of the time, to do something more creative than that.

M: I am not saying they never did. But the above kinds of activities recurred quite often and certain children stayed with them over long periods of time. In fact, such phenomena have also been reported by Leron, Hillel and others.

E: Well, come to think of it, I recall that some of these activities actually created a lot of fun because of the surprise element, and probably led to some sense of accomplishment and even success.

M: Yes, but such "success" is really more likely a mystery that has been left unresolved, and that might be beyond the child's current level of ability.

E: This kind of experience is likely to leave the child with a sense of inferiority rather than mastery.

M: Furthermore, the child will learn to act spontaneously at the computer terminal, without any attempt at pre-planning or setting up goals.

E: True. So, in view of all these unexpected negative learning outcomes, what do you think we should do?

M: First of all, we have to guarantee an adequate amount of knowledgeable adult or peer support in all Logo environments. We cannot afford leaving the child on his own to stumble.

Second, if we really want to try to optimize the child's learning, we should do our best to prevent cognitive overload on the one side, and boredom or repetitive meaningless activities on the other. Most importantly, we should guide the child to engage in challenging tasks or projects appropriate for his own level.

E: Then no more leaving the child on his own to discover?

M: On the contrary. There will be a lot of discovery and exploration left for the child, only it will be usually more narrowly focused and possibly somewhat guided at times. This can be accomplished by developing a rich repertoire of Logo microworlds and other software packages catering to different levels, needs and tastes. For instance, some of these microworlds could be simplified, narrowly focused versions of Turtle Geometry. Others could provide links with various curriculum areas (e.g. Turtle Geometry with music). Structured fun-like activity packages such as games could also be included to enable the child to practice specific skills. The teacher will.help the child select the appropriate package depending on his level of knowledge, particular needs and personal preferences.

E: You seem to be describing our Logo Microworlds Project at OISE.

M: True, I am describing the philosophy behind it. Because I believe that this kind of approach can help maximize the children's' learning in the Logo environment without taking away any of the fun. At the same time, it also somewhat reduces the amount of adult support required.

E: And what about the other research project involving measurement of cognitive skills?

M: We will continue with that project too.

E: Though I enjoy the microworlds project more.

M: So do I, as enriching Logo is more rewarding than evaluating it.

## References

*Notes: The following papers have been referred to implicitly in the dialogue:*

Billstein, R. "Turtle Fever". *The Computing Teacher.* 11(2), 1983, pp. 34-36.

Clements, D. H. "Supporting Your Children's Logo Programming". *The Computing Teacher*, 11(5), pp. 24-29.

Cohen, Rina Logo in the Grade Two Classroom: A one year study". OISE, 1983.

Cohen, R., Rubincam, I., and Barker, G. Final Report of the OISE-Funded Logo Microworlds Project. April 1984.

Higginson, W., Burnett, D., et al. "Interim Report". Queen's University, Faculty of Education, Kingston, Ontario, Canada, 1984.

Hillel, J. "Observations, Reflections and Questions About Children's Logo Learning". Department of Mathematics, Concordia University, Montreal, Quebec, Canada.

Krasnor, Linda R. and Mitterer, John O. "Logo and the Development of General Problem-Solving Skills". The preparation of this article was funded in part by Social Sciences and Humanities Research Council of Canada strategic seed grant #499-82-1010.

Leron, Uri "Some Problems in Children's Logo Learning". University of Haifa, Israel.

Moursund, d. "Logo Frightens Me". *The Computing Teacher*, 11(5), 1983, pp. 3-4.

Pea, Roy D. "Logo Programming and Problem Solving". Center for Children and Technology, Bank Street College of Education, Technical Report No. 12, 1983.

Pea, R. D. and Kurland, M. "On the Cognitive and Educational Benefits of Teaching Children Programming". New Ideas in Psychology, May 1983. 11(5), 1983, pp. 2-14.

# THEORIES OF LOGO

Guy Groen
McGill University

A strange paradox regarding Logo is that, while the overall approach has its roots in cognitive science, the empirical research has made very little use of the techniques that have evolved in this area (eg. Kintsch, Polson & Miller, 1984). This is in contrast to the situation in related areas such as research on children's' mathematical thinking (eg. Ginsburg, 1983), in which these techniques are beginning to be quite extensively used. The result of this use been the emergence of a body of research in which theory and data are closely linked. This is because the methods of cognitive science provide two things. The first is a language for talking, in a precise fashion, about the knowledge and processes that a person uses when performing such activities as thinking or problem solving (es. Hayes-Roth, Waterman & Lenat 1983). The second is a set of methods, such as Protocol analysis [Newell & Simon, 1972] and propositional analysis [Kintsch, 1974, Frederiksen, 1975], for making the transition from empirical data to a model or theory expressed in this language.

The purpose of this paper is to argue that a more extensive use of these methods might be of considerable value in research on Logo. First, I will discuss some problems due to the lack of an adequate theory that arise in attempting to interpret current research on Logo. Then, I will give a brief sketch of what an adequate theory might look like.

## Some problems with the interpretation of research findings

Most studies have tended to fall into two extreme categories. The first consists of extensive observations which are then used, in an informal fashion, to provide anecdotes that illustrate some aspect of the Logo approach. The second consists of studies in the tradition of educational evaluation, in which some hypothesis about the outcome of students' interaction with the Logo environment is tested by collecting behavioral measures and subjecting them to appropriate statistical analyses. In both cases, the problem is that we do not know why the observed results occurred. This can only be done in terms of a theoretical framework. Without this, it is impossible to say precisely what is being learned in the Logo environment. Most importantly, it is impossible to evaluate accurately the claims about general transfer of training (i.e.. the carryover from Logo to activities unrelated to computers) that have been made by many proponents of Logo, or even come up with an accurate interpretation of what such claims really mean.

Unfortunately, the Logo approach is not a theory. As defined in *Mindstorms* [Papert 1980], it is a set of statements about the benefits of Logo and the best ways to teach it. However, Papert does provide a large number of hints as to what an appropriate theory of Logo should contain, especially in his discussions of microworlds, powerful ideas and the relationship with artificial intelligence and Piaget's theory. Should we develop a theory by filling in the gaps between the hints, or is it better to ignore the hints and develop an original theory that nevertheless takes the Logo approach seriously?

The most ambitious and sophisticated recent attempt to develop a theory of this latter kind is the work of Pea and his associates at the Bank Street College of Education [Pea & Kurland, 1984]. It is essentially a theory of the development of programming skills and the kind of thinking that must underly the development of a successful program. While it does attempt to come to terms with the general transfer problem, they restrict their attention to the kind of transfer that might be due either to the necessity of developing certain kinds of planning and organizational skills in order to write successful programs or to the kinds of control structures (such as recursion) that Logo uses. There are, however, two problems with this approach. The first is that it is essentially a theory of learning computer programming. From this point of view, all languages are more or less the same except for their structural properties, their mnemonic power and their user-friendliness. However, languages also differ in the kinds of tasks that are simple to accomplish. Logo is a nice language for drawing pictures. Other languages are much nicer for manipulating two-dimensional arrays. The theory gives no basis for examining differences of this kind. The second is that it does not provide a way of predicting or explaining effects due to the task environment. In other words, it ignores the possibility that the kinds of tasks a student programs and the teaching method that is used may affect the outcome of a student's interaction with Logo.

If Papert's hints are taken seriously, a completely different kind of theory emerges [Groen 1978, Groen & Kieran 1983]. What is learned in Logo is not primarily a programming language. Its educational value, especially with children, comes from the fact that it provides a way of exploring microworlds. Powerful ideas are not generalized programming skills but ways of coordinating a microworld with its analogues in reality, or ways of coordinating between different representations of microworlds. For example, turtle geometry is a microworld. An analogue in reality is the world of drawing with ruler and compass. What the child is learning when involved in this microworld is not a programming language but a way of establishing correspondences between a concrete world and one of abstract representations.

## A sketch of a theory of microworlds

An adequate theory must turn these rather vague concepts into something precise and testable. This necessitates some kind of representation of the Piagetian notions inherent in Mindstorms together with some kind of notation for representing knowledge. It seems possible to do this by combining two notions. The first is an approach to formalizing Piaget's theory proposed by Groen [1978]. The second is the frame notation originally developed by Minsky [1975]. It should be noted that the idea of applying frame notation to Logo is not new. It originated in the work of Goldstein [1974] and has been used occasionally since then [Groen & Chait 1978, Miller 1982].

The basis of the theory is a formalization of Piaget's notion of structure. This is too complex to be described here in detail. The most important aspect is that a structure is a set of states and transformations between states. An important property is that the transformations should be modular. In other words, they should be easily decomposable into chunks.

A microworld is a structure with certain additional properties. The two most important are:

1) A transformation can be undone to go back to the previous state.

2) There should exist mappings (in the precise mathematical sense of the term) to other structures that are representations of concrete actions in the real world.

A definition along these lines is sufficient to distinguish between microworlds and non-microworlds. Turtle graphics is a microworld. The states are the possible positions of the turtle on the screen. The commands are the transformations. The procedures are chunks. Every turtle movement can be undone. Every procedure can be mapped onto a drawing with pencil and paper. On the other hand, numbers, words and lists are not microworlds (although subsets with suitable transformations might be). Neither are programming languages.

The frame notation can be used to model the knowledge that a student is using and developing by interacting with a microworld. A drawing consists of a sequence of actions of the following form:

```
    MOVE PEN (FOLLOWED BY) DRAW SOMETHING
```

This a very simple example of a frame. In general, a frame is a structure consisting of slots that are linked by relations, together with a set of rules that determine what can occupy the slots. In the above example, MOVE PEN and DRAW SOMETHING are slots while (FOLLOWED BY) is the linking relation. There is a corresponding frame in turtle graphics:

```
    PROCEDURE WITH PENUP (FOLLOWED BY)
      PROCEDURE@WITH PENDOWN
```

There is a third frame, which involves the child's experience with objects in the real world:

```
    SHIFT ATTENTION (FOLLOWED BY) LOOK AT OBJECT
```

Most frames are far more complicated. However, these very simple frames are sufficient to give a precise definition of something that might be a powerful idea. The three frames can bė joined together in a single superframe, by introducing a new relation (CORRESPONDS TO). It might be drawn with the real-world frame on top, followed by the drawing frame in the middle and the Logo frame underneath. The powerful idea, then, is the process by which this single frame is constructed in the student's mind from the three separate frames.

This process, and others like it, cannot be specified more precisely without empirical evidence. However, we are now squarely within the realm of cognitive science since the use of some variant of the frame notation is the standard technique for representing knowledge both in artificial intelligence and in cognitive psychology. Moreover, the propositional analysis techniques mentioned at the beginning of this chapter provide a method of analyzing verbal protocols that yields frames as its end product [Patel, Frederiksen & Groen, 1983]. Such techniques are far more rigorous than any that have been used to date in the analysis of students' interactions with Logo. However, they grew out of research on comprehension and their use may result in process models that are somewhat different from those normally encountered in the literature on problem solving, in which frames (when used at all) define conditions that cause processes to "fire". Our experience with propositional analysis techniques suggest that the kind of model that emerges is one of processes that transform frames, with final frame essentially defining the solution to the problem.

## References

Frederiksen, C. H. (1975) Representing logical and semantic structure of knowledge acquired from discourse. *Cognitive Psychology*. 7, 371-458.

Ginsberg, H. (1983) *Development of mathematical thinking*. New York, Academic Press.

Goldstein, I. (1974). Understanding simple picture programs. TR 294. MIT AI Laboratory.

Groen, G. J. (1978) The theoretical ideas of Piaget and educational practice. In Suppes, P. (ed) *Impact of research on education: Some case studies*. Washington, DC, National Academy of Education.

Groen, G. J. & Chait, S. (1978). Goals and strategies in a computer based graphics environment. American Educational Research Association Meeting.

Groen, G. J. & Kieran, C. (1983) In search of Piagetian mathematics. In Ginsberg, H. (ed) *Development of mathematical thinking*. New York, Academic Press.

Hayes-Roth, F., Waterman, D. A. & Lenat, D. B. (1983) *Buildings expert systems*. Reading, Mass., Addison Wesley.

Kintsch, W. (1974) *The representation of meaning in memory*. Hillsdale, N.J., Erlbaum.

Kintsch, W., Miller, H. & Polson, P. (1984) *Problems of methodology in cognitive science*. Hillsdale, N.J., Erlbaum.

Miller, M. (1982) A structured planning and debugging environment for elementary programming. In D. Sleeman & J. S. Brown (eds), *Intelligent Tutoring Systems*. London, Academic Press.

Minsky, M. (1975) *A framework for representing knowledge*. New York, McGraw Hill.

Newell, A. & SIMON, H.A. (1972) *Human Problem Solving*. Englewood Cliffs, N.J., Prentice-Hall.

Papert, S. (1980) *Mindstorms*. New York, Basic Books

Patel, V.L., Frederiksen, C. H. & Groen, G. J. (1983) Relationship between comprehension and medical problem solving. Research in Medical Education Conference, Washington, DC.

Pea, R. M. and Kurland, D. M. (1984) On the Cognitive Effects of Learning Computer Programming. *New Ideas in Psychology* 2 (2).

# SYMBOL SYSTEMS AND THINKING SKILLS: LOGO IN CONTEXT

Roy D. Pea
Center for Children and Technology
Bank Street College of Education

Imagine yourself as a visitor to a traditional farming society in West Africa. You have arrived as a cross-cultural psychologist to study whether and how literacy affects the way people think. Let us begin by peering into your mind to find out why you are here.

The acquisition of literacy had long been claimed to promote the development of intellectual skills. Prominent historians and psychologists had long argued that written language has many important properties that distinguish it from oral language, and that the use of written language leads to the development of highly general thinking abilities, such as logical reasoning and abstract thinking. Piagetian studies in other cultures had made clear that the kind of abstract thinking associated with formal operations did not develop in oral cultures. By contrast, when one looked at cultures that used written language, various cognitive tasks revealed high logical competencies.

But you had observed that studies bearing on this claim had always been done in societies such as Senegal or Mexico, where literacy and schooling were confounded. Perhaps schooling is responsible for these changes in thinking, rather than the use of written language *per se.*

The reason you have traveled to Africa is that you plan to test,for the first time, the cognitive effects of literacy *independently* of schooling. The society you are studying--the Vai--does not transmit literacy in the Vai written language through formal schooling. Their reading and writing are practiced and learned through the activities of daily life.

The Vai invented their written language a mere 150 years before, and have continued to pass on literacy to their children without schools.

Like all the psychologists before you, you have brought along suitcases filled with standardized psychological testing instruments and stimuli for experiments on concept formation and verbal reasoning. Results from performances by the Vai with and without written language experience will tell you whether possessing literacy affects the way these people think.

But as you look over your results from several years of work, you see no general cognitive effects of being literate in the Vai script. For example, the literate Vai were no better than the nonliterate Vai in categorization skills or syllogistic reasoning. Literacy *per se* does not appear to produce the general cognitive effects on higher thinking skills you expected.

So you mull over this fact for some time. How could this be? The arguments were so plausible for why written language would affect the way people think. You wonder--could the studies have been done more carefully?

But before continuing this research strategy, you realize that there is a radically different way to think about your project. When you arrived you took for granted the grand theory that literacy will have its intellectual benefits. But with several years of survey and ethnographic observations under your belt, you have come better understand the tasks that Vai literates encounter in their everyday practices of literacy. But how does this relate to your experiments?

What you decide you could do instead is to actually look to see how literacy is *practiced* in the Vai culture. What is done with the written language? And then you ask a very different type of research question: How could what the Vai people do *specifically* with the written language affect their processes of thought? You decide to let your fieldwork on literacy practices dictate the design of "outcome" tasks and you gain a great deal of precision in your hypotheses for the cognitive effects of literacy.

This reorientation literally turns your theory-driven paradigm of looking for general cognitive effects of literacy on its head. You have shifted from making general predictions in terms of developmental theory about concrete behaviors, to starting with concrete observations of literacy behavior and building up to a general *functional* theory of literacy's effects.

With this new approach you find that the Vai use their written language primarily for letter-writing, and for recording lists and making technical farming plans.

Then you begin a new phase of your research project, seeking out cognitive effects of specific literacy *practices* rather than literacy per se. You design new tasks for assessing literacy effects that draw on related skills to those required by the practices you observe, but which involve different materials.

What you find when directed by this new functional perspective are dramatic cognitive effects of literacy. But they are more local in nature. For example, letter writing, a common Vai literacy practice, requires more explicit rendering of meaning than that called for in fact to fact talk. So you refine a communication task where the rules of a novel board game must be explained to someone unfamiliar with it, either face to face or by dictating a letter for an absent person. You find, lo and behold, that performances of Vai literates are vastly superior on either version of this task to those of nonliterates.

This is no mere parable. It is an account of an extensive five-year research project carried out by Professors Sylvia Scribner and Michael Cole (1981). It is the account of an intellectual voyage not so far removed from what I have to say about what children learn with Logo, for we can fruitfully apply the schema of this Vai story to questions about the cognitive effects of programming.

Here, too, there are persuasive and intuitively appealing arguments for why people should become better thinkers by virtue of the use of a powerful symbol system such as the Logo programming language. It is alleged that children will acquire general cognitive skills such as planning abilities, problem solving heuristics, and reflectiveness on the revisionary character of the problem solving process itself. The features of programming literacy assumed here include the necessarily explicit nature of writing program instructions, the strategic and planful approaches ingredient to modular program design, and experience with the logic of conditionals, flow of control, and with program debugging.

But for programming languages, unlike written language, we do not have the benefit of known historical and cultural changes that appear to result in part from centuries of use of the written language. The symbol systems provided by programming languages are relatively new. They have certainly changed the world; we now live in an information age because of achievements made possible by these languages. But what does it mean for how individuals think and learn?

Let us move our West African story to the context of the American Classroom. Here again we enter as psychologists, looking for general cognitive effects, much like the first literacy questions of the African enterprise.

Of course we assume that we know what kind of a mind-altering substance programming is (having been so affected ourselves), and we assume that "programming intelligence" and the kinds of programming activities carried out by adults will affect children too.

But we should give pause--for we have entered another culture. What will children do with a programming language in a discovery--learning situation, Logo's "learning without curriculum" pedagogy, without benefit of being shown what kinds of things can be done, or being taught about the powers of the system or of thinking skills?

Nonetheless, without benefit of such hindsight, what do our psychologists in the Logo classroom do? They too look for programming's "effects," guided by somewhat the same kind of thinking that possessed the first phase of the Vai studies. The primary difference was that instead of testing for increments in general intelligence, or concept formation, they thought they were looking at more specific effects, quite plausibly linked to programming activities. Planning skills were the central focus, not abstract reasoning, which is only indirectly related to programming.

The psychologists' reasoning went something like this: Both rational analyses of programming and observations of adult programmers show that planning is manifested in programming in important ways. Once a programming problem is formulated, the programmer often maps out a program plan or design that will then be written in programming code. Expert programmers spend a good deal of their time in planning program design, and have many planning strategies available, such as problem decomposition, modular documentation, subgoal generation, retrieval of known solutions, and evaluative analysis and debugging of program components (e.g. Pea & Kurland, 1983).

Our psychologists studying the cognitive effects of Logo created planning tasks to reveal the development of different planning strategies, and of skills at plan revisions analogous to program revisions. In two different studies, after a year of Logo programming, these psychologists found no effects of programming on performances in these planning tasks (Pea & Kurland, 1984). Children improved with age and practice on the planning tasks, but non-programmers did just as well after a year's time as did Logo programmers. Once again, like the researchers in West Africa, we must reflect on our first set of assumptions for framing the research questions, and reconsider the meaning of our research findings.

Let us take a different, functional or activity-based approach to programming. Consider "programming" not as a *given*, whose features we know by virtue of how adults do it all its best, nor as what it looks like in its ideal text-book forms, but as a *set of practices* that emerge in a complex goal-directed cultural framework of thought, emotion, and action.

Viewed in that way, by analogy to the Vai studies on literacy practices, we see that programming is as various and complex an activity matrix as literacy. Just as one may use one's literacy in Vai society to make laundry lists rather than analyze and reflect on the logical structures of written arguments, so one may achieve much more modest activities in programming than dialectics concerning the processes of general problem solving, planning, precise thinking, debugging, and the discovery of powerful ideas. One may, in particular, write linear brute-force code for drawing in turtle graphics.

Stated baldly, from a functional perspective we may see that powerful ideas are no more attributes inherent"in" Logo than powerful ideas are inherent "in" written language. Each may be put to a broad range of purposes. What one does with Logo--or written language--or any symbol system, for that matter--is an open matter. One must come to these powerful ideas and *potentially* fertile grounds for developing general thinking skills through discovery, or through learning with the guidance of others. Independent discovery and practice of Logo recursion, for example, may be a very rare spontaneous occurrence. The Vai have not spontaneously got onto the logical features of written language, philosophy, and textual analysis that written language *allows*. Likewise, most of our students--from grade school up through high school--have not spontaneously got onto the programming practices, such as structured planful approaches to procedure composition for reusability as building blocks in other programs, use of conditional or recursive structures, or careful documentation and debugging, that Logo *allows*.

For the Vai, one could imagine introducing new logical and analytic uses of their written language. Similarly, one could imagine introducing to children the Logo programming practices many educators have taken for granted will emerge. In either case, we would argue that without some functional significance to the activities for those who are learning the new practices, there is unlikely to be successful, transferable learning. Serving some purpose--whether being able to solve problems one could not otherwise, satisfying an intrinsic interest in complex problem solving, or achieving solidarity with a peer group who define their identity in part by "doing" Logo or written language--is a necessary condition for the symbolic activities we are interested in promoting to be ones our learners find a commitment to.

It is my hunch that wherever we see children using Logo in the ways its designers hoped, and learning new thinking and problem solving skills, it is because someone has provided guidance, support, ideas for how the language *could* be used. They will have pointed the way through examples, rules, and help in writing programs and discussing the powerful ideas. To call these rich activities "learning without curriculum" is misleading, and an overly narrow view of what constitutes curriculum, for any projected path toward greater competency that another person helps arrange can be thought of as a curriculum.

There are many profound consequences of this more general account of what is involved in thinking about Logo as potential vehicle for promoting thinking and problem solving skills. A functional approach to programming recognizes that we need to create a *culture* for Logo in which students, peers and teachers talk about thinking skills, display them aloud for others to share and learn from, a culture that continually reveals how programming is a vehicle for learning general thinking skills, and that builds bridges to thinking about other domains of school and life. Such thinking skills, as played out in programming projects, would come to play functional roles in the lives of those in this culture. Dialog and inquiry about thinking and learning processes would become second nature, and the development of general problem solving skills so important in an information age would be a common achievement of students. This vision could be realized. I imagine that important cognitive effects of programming, or of literacy are *possible*, but only when certain uses of these symbol systems are practiced, not the ones most engaged in today. There is far too much faith today that Logo carries with it guarantees of cognitive outcomes, and I have fears that when these profound changes are not found, educators will be prematurely discouraged.

Where are left after these two continents of travel? With the bright sound of an optimistic chord. There are many streams of Logo activities and research that should go on, for plurality and diversity provide exciting grounds for emergent ideas. Communication among groups, such as this forum provides, will help in the formation of a broad community exploring these issues. These streams will no doubt embody a diversity of assumptions about what will best help create the culture of Logo I have referred to, in which one will be more likely to find the cognitive effects on thinking skills so many take for granted. Similar Logo cultures may arise that center on math learning, or programming.

It is uplifting that there are so many positive energies in education today. The enthusiasm for Logo as a vehicle of cognitive change is an exhilarating part of the new processes of education one can see emerging. Cultures with thinking tools like Logo can be created. But we must first recognize that we are visitors in a strange world--at the fringe of creating a culture of education that takes for granted the usefulness of the problem solving tools provided by computers, and the kind of thinking and learning skills that the domain of programming makes so amenable to using, refining, and talking about together.

## Acknowledgments

I would like to take this opportunity to thank the Spencer Foundation and the National Institute of Education (Contract #400-83-0016) for supporting our research programs on the development of Logo programming and its relation to other cognitive skills. My colleagues at the Center for Children and Technology have been a continuing source of encouragement and stimulation.

## References

Pea, R. D., & Kurland, D. M. (1983). *On the cognitive prerequisites of learning computer programming.* Interim Report to the National Institute of Education (Contract #400-83-0016).

Pea, R. D., & Kurland, D. M. (1984). *Logo programming and the development of planning skills.* Technical Report No. 16. New York, NY: Center for Children and Technology, Bank Street College of Education.

Scribner, S., & Cole, M. (1981). *The psychology of literacy.* Cambridge, MA: Harvard University Press.

# LOGO AS AN EMPIRICAL WINDOW

Sylvia Weir
Massachusetts Institute of Technology

AN EXCERPT FROM THE DRAFT OF THE FORTHCOMING BOOK
"UNTRAPPING INTELLIGENCE"

## How can we say what children learn until we know what they know already?

Observation of children's responses in computer-based environments can make their thinking more accessible than it would otherwise be. This has two interesting corollaries. Observing children in computationally oriented educational settings can help *teachers* see more clearly what is going on during the learning processes of their students; and it can help *researchers* into children's understanding see more clearly what is going on in the heads of their subjects, whose learning in these "real", classroom environments is likely to be more relevant than the sparse laboratory experiment allows. There is a shift of emphasis on the part of both groups. For the teacher, the concentration on "how can I make my explanation clearer?" moves to "How can I set situations up so as to enable my students find their way to an understanding?" For the research worker, "How can I test for the presence of skills A, B and C?" broadens to "What must I provide so that what my subjects know already, as well as what they can come to know, will surface in a clear way for me to observe and probe?"

One approach is illustrated by the several partnerships that are emerging between the MIT Logo group and schools in the area: the Cotting School in Boston; the Carroll School in Lincoln; and the Quincy and Ohrenburger Schools in the Boston Public School System. The researchers have participated in the teaching, and this shared teaching, together with joint planning and training sessions, has become an integral part of the process of introducing computers into the school curriculum. The teacher's rich store of informal anecdotal information can be explored and formalized. The school administration is involved with suitable financial support in accommodating the research by appropriate release time for teachers to participate, not as a chore but as a chosen interest.

## Catalysing the surfacing of intuitions

Let's focus on the role of Logo in helping a child find a meaning for numbers. During the process of constructing a Logo program, it is necessary to choose numbers as inputs for the commands. The thesis is: A number comes alive when you see it as something that does a given amount of work for you. Forward 10 takes the turtle twice the distance that Forward 5 does. A typical student protocol goes something like this:

> How much do I need to go forward here?
>
> That was Forward 20.
>
> This space is just a little more than twice that ..
>
> Twice 20...40
>
> Let's try Forward 45.

So the meaning is related to the purpose being served, and the purpose may be generated by the child as part of his active engagement, or by the adult teacher or experimenter. Several studies over the past few years have concerned students' choice of numbers during their early contact with Logo. Our first study looked at spontaneous, open-ended choice of number in situations where the adult refrained from suggesting a goal. We went on to look at number choice when the goal was set by the teacher-experimenter. What has been of interest to me is the way this environment provides a way to get a look at the explanations people develop about the effect of their actions, whether they stick to these explanations and how much it takes for them to change their behavior and their explanations.

## Spontaneous open-ended choice of number

What can be learned by the first few numbers used by a beginning Logo student? How much will this depend on age, on mathematical sophistication, and on personal style? Upon being introduced to the command FORWARD, individuals frequently choose a *single digit* input. Sometimes, the user resists choosing, and asks: what's the scale here? How do you mean: "pick any number"? An invitation to: "let's try anything and see", usually produces a response. Having chosen this single digit, the user often shows surprise at how small a distance the turtle travels in response to say FORWARD 9. After all, how is a beginner to know that the scale for the size of turtle steps chosen by the implementors of Logo allows the user to fit 200 or so steps on to the screen? Incidentally experiencing the arbitrariness of the choice of scale is part of sorting out what the computer knows and how to change that.

Out of a group of 15 children aged 6 - 10 years,[1] the first choices for inputs to FORWARD were as follows: 3 chose numbers greater than 10 (39, 100 and 1000); 3 chose 1; 6 chose 9; and one each chose 3, 4, and 8 respectively. This suggests that to most of these children, what counts as a number is any digit from 1 to 9. Furthermore, it is interesting to note the high proportion of choices that lie at the extremes of the 1 to 9 range, reminiscent of the privileged place that first and last letters in a word have. I wrote this I noticed that I too used 9 as the input to FORWARD in the previous paragraph.

Let's propose an ultrasimple theory to account for this. Suppose the basic representation of numbers in the very young child is in a string-like structure, each element connected only to the one in front of it and to the one behind it, except the first and the last elements in the string. The last element is connected only to the one in front of it. The first connects to the one behind it, and itself forms the access point to the whole string, so that the only way to get at the numbers in the string is by entering the string at its first element. The process of finding a given number would need to start with the first element, see if it matches the given number, in this case, three; if it does not, then the process moves on, takes the next number and checks for a match. When the three is found, the process then simply takes the one that follows for the answer. At this stage, the child would be able to count forwards and not backwards; and when asked for, say, the number after 3, would say: "one, two, three, four. Four". That is to say, the child needs to start from the beginning each time when counting up.[2] Later, a second point of access to the string is established, namely, at the tail end of the string. Now going backwards along the string is possible. As the child grows, increasing knowledge about number manipulation is recorded as a set of rules, each rule describing a test to be applied to each number obtained in the search along the list.

In open-ended activity of the sort we are dealing with in Turtle Geometry, no constraints are placed on what number is to be used by the adults in the situation, and so no particular test applies. So our children lapse into the

---

[1]Seen by Kellerson, a member of the project and one of a group of students supported by MIT's Undergraduate Research Opportunities Program

[2]See Aaron Sloman, *The Computer Revolution in Philosophy*, The Harvester Press, 1978, for a discussion

situation of the very young child and simply pick the first number encountered in the string. They can pick either 1 or 9 since the string can be accessed at either of these points, but tend not to move along it.[1] This suggests a little piece of research that Logo teachers could carry-out. Look to see if there is a developmental trend in the first number chosen, that is to say if the pattern of number choice changes with age. The sample I have above is much too small to decide this, and the youngest children are not young enough. All we can say is that the trend is there: 2 of the 3 who began with 1 were at the youngest end of the age range -- 6 years, while 5 of the ones who began with 9 were 8 years or older. The point I am making here is that we have in the past not had many opportunities to observe spontaneous choice of number in children, and so each Logo classroom is a potential source of empirical data about what happens in these circumstances.

When it comes to choosing the second number, a much more constrained situation can arise, since this reflects the student's reaction to seeing the effect of the first number on the turtle. What did the turtle do? What did the number mean to the turtle? Some children do react to feedback, and make appropriate choices, jumping immediately to a considerably larger number. Thus, among the 12 in our sample who started with a single digit, four made the jump right away: 1 to 44, 9 to 99, 9 to 100, 9 to 999. However, a surprisingly large number stayed virtually in the same part of the number string, sticking with a single digit number. Three even stayed at 9. Subsequent progress varied with the child. Some gradually but spontaneously built up to the recognition of the effect of larger numbers. Still others need considerable prompting before using a two digit number. Here are the first few numbers of five children beginning Logo (ages in parentheses), to illustrate the range of response obtained.[2]

PAUL (9): FD L2 FD L00 FD L00 FD L00 BK L000 RT 1 RT 99 FD 10,000 LT 9

ELIZA (14): FD 6 FD 9 FD 22 RT 33 FD 55 LT 90 FD 90

JEAN (11): FD 7 FD 9 FD 8 FD 20 FD 50 FD 40 FD 100 FD 120

---

[1]"lapse into" could mean that the early representation, with its two access points at the beginning and end of the string 1 - 9, is the same representation that in older children becomes the string 1 - 9 that generates the elements out of which all numbers are constructed.

[2]Reported by Joyce Kelly, one of a group of students on the project supported by MIT's Undergraduate Research Opportunities Program

MANNY (9): FD 9 FD 7 RT 9 RT 9 RT 98 FD 99 RT 100 FD 100 RT 60

LEAH (8): FD 9 FD 3 FD 6 FD 9 FD 7 FD 8 FD 8 FD 7 FD 6 FD 4567

There was a variation in *how long* a child continued to use single digit input before realizing that a larger number would produce a more useful effect.

i. Paul required feedback only once, and then made the jump spontaneously, first with respect to inputs to FORWARD -- from 12 to 100; and then for inputs to RIGHT -- from 1 to 99.

ii. Eliza used single digits twice before she jumped to a larger number, again on her own initiative. Her jump was smaller than Paul's -- from 9 to 22 as input to FORWARD; but then she stayed with two-digit numbers for the RIGHT command.

In contrast, the other children were given varying degrees of prompting by the teacher.

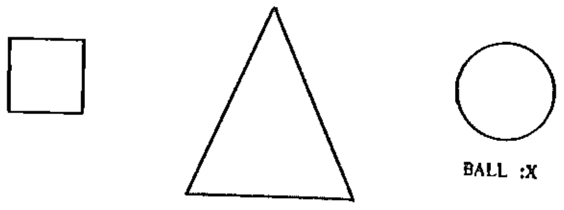
iii. Jean chose 7, then 9, followed by a retreat to 8; at each step, she complained about the very short distance the turtle was traveling. This led to an intervention by the teacher, and her move the two digit number 20.

iv. Leah was quick to develop a goal. Her goal was to reach the top of the screen. Yet she used 9 single digit numbers in a row, and then when prompted by the teacher to use larger numbers, she chose FORWARD 4567 -- very inappropriate relative to the feedback she was receiving about the turtle's scale.

v. Manny made four small moves, sticking at the 9 "barrier", at which point, again, the teacher intervened to suggest that higher numbers were allowed. That was all he needed.

In addition to looking at the size of the number chosen, we can distinguish several *kinds* of number chosen. Four out of the above five children moved towards an almost exclusive preference for *multiples of ten*. In contrast, many children use numbers like 45 67 23 12, or 33 88 44. They are either pushing the same key twice, or pushing adjacent keys. As far as I can tell, this kind of choice seems to have more to do with the proximity of those keys to one another, i.e., with typing convenience -- for one- finger typing, this tends to produce two same numbers, while typing with several fingers produces runs of consecutive numbers. It is interesting to note that, when these runs of consecutive numbers occur, they more frequently go up rather than down.

This tendency to go in a sequence up the number string in a way that is unrelated to the immediate goal of the activity occurred in an extreme form in a child at the Cotting School for Handicapped Children. James, a 14-year old quadriplegic, showed a surprising and idiosyncratic number choice. There are many places in his work with Turtle Geometry where he started with 14, which is his age, as the argument for a turtle command, and then continued along the number line regardless of whether he was instructing the turtle to turn right, to go forward or to go back, and regardless of the needs of the problem. James built his Christmas tree out of procedures he had been given. He was required to provide numbers as input to these procedures. Using the ball procedure with different inputs, he produced figure 1. His program instructions, given alongside the figure, show his peculiar choice of numbers. His solution is to treat the problem as though it is a counting forward problem rather than choosing numbers appropriate to the need. In terms of the theory enunciated earlier, he appears to have no rules for modifying the number generated by his basic number-string accessing mechanism. Incidentally, such behavior was not anticipated, and its existence would not have been suspected had Logo not provided the appropriate stage on which he felt free to perform, and show us what numbers he would choose *spontaneously*, rather than being asked to work with numbers supplied by his teacher.



```
BALL  9
FD     10
BALL    11
HT
ST
FD 12
BALL    13
FD     14
```

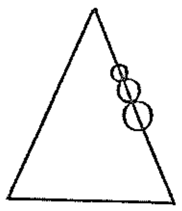


FIGURE 1

## Constraining the Choice: Introducing Goals

Situations arising from *student-generated* goals form the standard Logo fare, and are useful to study. During his second lesson, Brian, a Cotting School student, decided to see how much it would need to get the turtle from the top of the screen to the bottom of the screen. An appropriate sequence of moves would be to first get within striking distance of where one is aiming at, and then home in on the target -- by successive approximation. Brian did just that. He went:

```
BK 75 BK 65 BK 34 BK 6 BK 5 BK 4 BK 5 BK 6 BK 3 BK 3;
```

followed immediately by a series of moves to get back to the top of the screen

```
FD 196 FD 2 FD 4 FD 5 FD 3;
```

and then

```
BK 196 BK 13
```

to reach the bottom again.

The 196 input to FD is interesting. Why didn't he just add up all the numbers (75 + 65 + 34 + 6 + 5 + 4 + 5 + 6 + 3 + 3) and give the exact figure? Did he make a rough computation, or did he attempt a complete addition, and perform this inaccurately? Or was it an estimated figure? Answers to these kinds of questions require more systematic observation. They do not just fall out of the regular activity, but require us to contrive a situation in which these various possibilities can be investigated. Once we have a rich situation that yields relevant information, we can try to get some systematic group data. Our aim is ambitious. We are interested in having a highly structured situation while at the same time retaining the characteristic feature of Logo, namely that the activities have meaning for the student, and thus retain a strong sense of involvement in solving the problem. Getting the right situation is not easy, because there are rather a large number of variables in any Logo problem-solving setting.

## Number Choice As Constrained By Teacher-Experimenter Goals

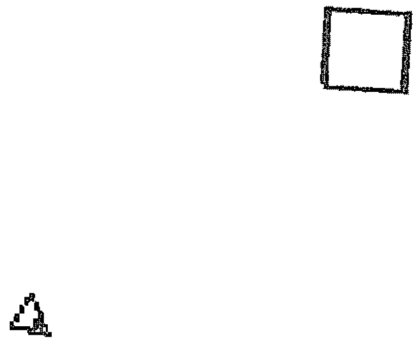


FIGURE 2: THE TARGET GAME

One setting in which number choice is constrained by a goal external to the student is the *target game*, in which the child is asked to move the turtle to the target. Since the turtle starts off pointing up, and the target is off to one side, this game requires inputs to the RIGHT and LEFT turn commands. Refining number inputs to the TURN commands so as to change the direction the turtle will move presents problems for most students. In the example quoted above, we saw a marked contrast between Paul's sure touch in choosing numbers for FORWARD movements, and a less than impressive choice when estimating degree of turn. With his first few FORWARD moves he showed an immediate response to feedback by adjusting the size of his input, and chose multiples of ten for his inputs. This was not the case for the TURN commands. RIGHT 99 is not a particularly "useful" angle for getting a regular figure, or for keeping track of where you are in spatial navigating tasks. But how can a beginner know that?

Figure 3 shows the usual depiction of an angle.

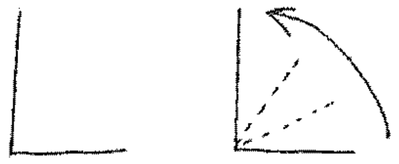


Figure 3a Figure 3b

Here is the usual depiction of an angle.

In turtle geometry, the angle between two lines is the amount you would need to turn if you were to do the following:

1) start off facing in the direction of one of the lines

2) turn on your heel until you are facing in the direction of the other line

Seeing angles as the amount of change in direction gives a good intuitive basis for an understanding of the concept, gives a physical meaning to add to the visual information. But where is the information about *amount of turn* to come from? You have to be told the convention that the special angle of 180 degrees is the amount you need to turn to face in the direction opposite the one you are facing now; and that 360 degrees of turn gets you all the way round. Sometimes a clear intuition about angle meaning is coupled with the "wrong" number.

> Six-year-old Peter answered the question:
> "How much to turn for a square?" with this
> reasoning.
> After thinking a moment, he turned his body
> all the way round.
> He then turned 90 degrees and mumbled:
> "That's a quarter of the way round.
> Let me see. If the whole is 100,
> then that will be 25.
>
> Lets try RIGHT 25".

Well, why not! 100 is a very good number to hazard. Indeed I have been claiming that choosing multiples of ten shows some number skill. The whole business of angle is much more difficult and certainly the children playing the target game found choosing inputs for RIGHT and LEFT presented more difficulty than did choosing those for the FORWARD and BACK commands.

Initial string of numbers chosen for target game:

MANNY: RT 198, LT 198, RT 189, LT 99, LT 77, RT 25.

PAUL: LT 143, RT 16, RT 165, RT 143, RT 13, RT 13

CARL: RT 452 LT 24

Now we look at the effect of the target goal on choice of inputs to the FORWARD and BACK commands. As with Brian's spontaneously adopted task of finding the edge of the screen, a child with a good sense of scale will use larger numbers to start with, and then, as he approaches the target, will use smaller numbers to home in on it. Many of our students did this (Figure 4). In addition, some children used multiples of ten as input to the commands, until they were sufficiently close to the target to need single digits (Figure 5). In examining the overall strategy and commands used in the target game, Kelly defined a unit and called it an "attack". An attack consists of any number of consecutive turns right and left from one forward (or back) up to, but not including, the next forward (or back). For example, FD 20 RT 7 Lt 21; FD 5 RT 2 LT 74; is two attacks. Kelly analyzed responses to the target game in terms of this unit, and drew three conclusions:

1) A smaller number of attacks per target (Figure 6) would seem to indicate a 'good eye' for lining up two points.

2) A larger number of turns per attack (Figure 7) would seem to indicate a lack of knowledge of the effect of particular numbers chosen and their relation to size of turn.

3) A larger number of forward's per attack (Figure 8) would seem to indicate a poor grasp of linear scale.

A comparison of the overall performance of individual children suggests trends. As the tables show, Joe does well in the target game. He uses successively smaller numbers as he approaches the target; chooses multiples of ten predominantly; has a small number of attacks per target; small number of turns per target with a moderate number of forwards per attack. Kelly describes Joe as "very taken with Logo". Judging from his Logo work, he seemed to have a spatial aptitude. He enjoyed learning new concepts, for example, the REPEAT command and sub-procedures, and employed them readily and appropriately in several projects. In contrast, a child such as Manny was able to aim the turtle very accurately, that is to say, he could judge when the line-up-was good. But he had no idea of what numbers to use to achieve that line-up, i.e., he was not economical in his number choice to get there. So he took nearly six turns to get to where he wanted the turtle to face, but once there, did not need to turn again, i.e., he had lined up accurately. Furthermore, when facing the right way, he then needed 6 forward steps before he hit the target, i.e., he could not fit numbers to the distance he wanted to cover.

## Perceptual Computation

Franky, a 10-year-old learning disabled child, illustrates the point. His reading and spelling are several years below expectations for his age and grade; he knows some mechanical processes for computation, but these often break down. He is described in school records as having *behavior problems, a short attention span, and a low tolerance for frustration*, a triad familiar to teachers of children with special needs. In his Logo work, Used Successively Smaller Numbers as Approached target

| always | sometimes | never |
|---|---|---|
| Joe | Pat | Hazel |
| Dan | Paul | |
| Manny | Carl(yes w/fd's no w/turns) | |
| Eliza | | |
| Philip & Matthew | | |

figure 4

Used Predominately Multiples of 10

| YES | NO |
|---|---|
| Joe | Manny |
| Dan | Paul |
| Hazel | Carl |
| Pat | |
| Eliza | |
| Philip & Matthew | |

figure 5

Average # of attacks/target:

| < 2 | < 3 | < 4) |
|---|---|---|
| Joe (3) | Eliza (3) | Carl (2) |
| Manny(1) | Paul (2) | Hazel (1) |
| Pat | P & M (5) | |
| | Dan (2) | |

figure 6

Average # of turns/attack:

| < 2 | < 3 | < 4 | < 6) |
|---|---|---|---|
| Joe | Dan | Paul | Manny |
| Hazel | Eliza | | |
| P & M | Pat | | |
| | Carl | | |

figure 7

Average # of fd's/attack:

| < 2 | < 3 | 6 |
|---|---|---|
| Dan | Joe | Manny |
| Hazel | Pat | |
| Carl | Eliza | |
| P & M | Paul | |

figure 8

Franky[1] was quick to remember individual commands as well as sequences of commands after a single exposure to them. He showed an unexpected flair for using numbers appropriately to create relationships in space, to invent units larger than one to suit his own purposes, and to readily use yardsticks for estimating length. In addition to a good sense of scale and accuracy of estimation of extent and aim, he displayed a very good appreciation of symmetry. At one point during his Logo work, he needed to move a certain distance on the screen in order to place his next star where he wanted it, so as to complete his multiple star design.

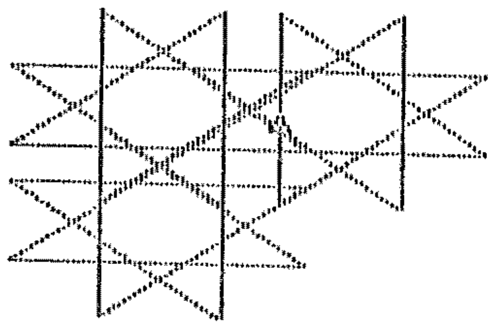


FIGURE 9

The distance was half of 75 -- a problem he was unable to compute using standard arithmetic operations. He looked at the distance on the screen and said, "Oh -- it's about 37." He could readily estimate half the length of a line in turtle units, but could not divide by two.

One can divide a line in half, in a Logo setting, by visually deciding where the half point is, and separately deciding how many turtle units correspond to the resultant line. Nowhere has one actually applied the arithmetic operation: divide the number by two. Franky seemed to have some of the skills he needed to manipulate numbers when he used a spatial framework, but did not seem to know how to operate in a purely numerical situation. When math was presented as a series of mathematical sentences, such as 33 + 48 = ?, he showed poor numeric reasoning skills. When given a spatial model he could use to figure out a problem, he could use it successfully, but if he was presented only with the numbers, the *idea* of using the spatial model did not occur to him independently.

[1]Taught by Susan Jo Russell

Logo provided the empirical window through which we were able to view an ability previously hidden. There are many Franky's in our school-system, penalized by a heavily language-based curriculum, yet unaware of the connection between their spatial ability and doing math. The connection needs to made explicit, rather than left to emerge during standard Logo activities. Bridges need to be built so that the Logo work is intimately integrated into the standard classroom curriculum, in a way that builds on the ability of children to perform perceptually based computation. An example of a collection of pencil-and-paper activities developed at the Carroll School will be the subject of a poster session at this conference (Weir and Arey, in preparation).

# EXPECTATION IS PART OF THE ENVIRONMENT

E. Paul Goldenberg
Lincoln-Sudbury Regional High School

Okay, so everybody (who's anybody) is already convinced that the so-called "new literacy" is, at the most generous best, a red herring. Not only are the contents of such courses suspect, but even the name itself invites misconceptions (see May/June 1983 *Classroom Computer News'* bowdlerized version of Brian Harvey's "Stop Saying Computer Literacy" or, better yet, the original version that was apparently too hot to print).

But the computer mystique seems to march right on. Intelligent kids, parents, teachers, administrators, and educational planners (not to mention writers and conference participants!), even while hotly disputing what is the best way to use computers, *act* as if they already *know* that schools with computers are better than schools without. For starters, this reminds me of the computer literacy debate in which all sides tacitly agree that there is a literacy somewhere to be found, and struggle only over the content. But I'm not so interested in the logical flaw as in the practical problems that this causes. All those intelligent kids, parents, teachers, administrators, and educational planners have considerable momentum. And they have been watching TV ads that threaten perdition for all who do not know that 16K is not a breakfast cereal.

I used to believe that computers could revolutionize education. If they can, I no longer believe that it will be their effects *in* schools that make the difference. The computer is certainly revolutionary in a way that our other recent technologies have not been, but new technology or technique--even revolutionary new--is not what changes institutions.

Believing is seeing; people see and learn what they expect to see and learn. Though novelty attracts attention, things that are too novel--experiences that differ too radically from the expected--are generally written off as anomalous. This is not to say, of course, that students learn nothing in such environments. After repeated exposure to radical new ways of thinking and learning, these "new ways" are no longer so radical, no longer anomalies to be ignored. But I am now describing a slow process in which people do the changing largely outside the classroom, and school, yet again alas, follows society rather than leading it--evolution, not revolution.

Another way of looking at this specifically invokes Piaget. We are not the creators of our students' experiences; *they* are! We can fill a room with whatever we like and run our classes however we choose, but The Environment is a product of both what we have provided and what our students construe it to be. Any educational planning, program, or curriculum that does not deal with plain real people as those plain real people are is misguided and arrogant, no matter how intellectually "right" the new program may be. Perhaps more to the point, I think it will fail.

## A case history: Lincoln-Sudbury Regional H.S.

This is all a bit abstract, so let me present a concrete example. I direct a computer department in a high school in an affluent suburban community near Boston. From its initial development (under the direction of Brian Harvey) to the present, this department and its computer center have represented radical departures from standard school policy -- even in this school with a liberal reputation.

## The "environment"

Our PDP-11/70 runs UNIX, an operating system that is very popular in universities and rapidly gaining ground in industry, but almost unheard of in a high-school. The language we promote is Logo, the press about which suggests that it is "Child's Play" (quoted from some Apple advertising) and suited only for introductory courses or young children. We do not teach Pascal (Apple's "Structured Sophistication") even though it has been sanctified by Educational Testing Service. On rare occasions, we have taught courses in C, APL, and LISP, but the most common way that students develop proficiencies in any language on our system is through their own independent study, a small portion of the time for which is sometimes bought off in the form of credits toward graduation.

The school administration gives us essentially total autonomy in developing our curriculum, managing our computer center, and even choosing how we interact with the rest of the school. There are no required computer courses, yet students have been offered a variety of options to satisfy credit requirements (e.g., Math and English) with courses that they elect to take with the computer. Students (this year 5 of them) maintain that complex time-sharing system, fixing bugs in it, adding features to it, developing new software for it, and having access to *everything* that is on that system, including all of the grades of all of the students in the school. Students (this year about 40 of them) have keys to the computer room and have access to the room on evenings and weekends year-round. Students who are not taking computer classes inhabit the computer center throughout the day, often even during classes that are in session in the center. The faculty continues to be prolific in its creation of curriculum ideas, position papers, and books. In virtually every aspect of our work, from our philosophical stands to the practical matters of money and management, we receive strong support from the administration.

We also have a unique product about which I enjoy bragging mercilessly: large numbers of students who are extraordinarily competent and knowledgeable C language programmers (from hacking UNIX) who have, while still in high-school, *designed on their own* as well as programmed such sophisticated programs as a text editor (the best we know of for UNIX, now distributed by UCB and soon to be a commercial product), a compiler for Logo (no mean task), an interpreter for LISP, a directory editor, and more.

But, everybody has computer wiz-kids. If we have more or better than the average, it is because we have provided extraordinary facilities for the kids to grow on and unlimited access to give them the time they need to develop. All of their real excellence develops (naturally!) outside of classes in the underground of our computer center, and among students who, like the best musicians, artists, and French-speakers in the school, are relatively few in number. What are we doing for the rest of our students?

With those students we have problems, and expectation may be much of the cause.

## 64K students on a two bit budget

Apparently harboring the suspicion that St. Peter is partial to programmers, a veritable host of students--two to three times as many as we can accept--registers for our Introduction to Computing course. Some, of course, are absolutely aching for the chance to get to work with the computer, to word-process or to learn to program. But our requests for enrollment are staggeringly high. Certainly, not all these students want to be programmers! Few have little interest in computers, *per se*, except, perhaps, for a vague curiosity, and none of them have *any* interest in having their way of thinking or learning mucked with. All that most of them want is to be blessed properly so that they can continue with their lives without being left out of the dreaded apocalypse. Yet, there are some who enter for weak reasons, have no idea what they want from the course, and become really excited by what they learned or did. Since we cannot predict which ones these will be, we accept students according to the priority system that governs enrollment in all classes in our school, even though this priority system seems to have little predictive value in knowing who will get what out of the class.

## The Intro course

Because there is no *essential* content, and also because we want to offer students real choice in the intellectual domain to which they apply our problem-solving techniques (partly in the spirit of educational reform and partly moved by a straightforward concern for engaging students in problem-solving in areas of particular interest to them) we suggest to students that they may follow any of several paths. What this means to some is that they have choices to make. What it means to others, even serious bright students who have learned the school game too well, is that all of the paths are equally unnecessary. Students *expect* no choice.

Because we have found letter grades to be such an educational distractor, we decided to offer all the computer courses as credit/no-credit courses without grades. What this means to students like the ones about whom I bragged earlier is that they can work for their own learning free of the threat/bribe system that dominates so much of education. What it means to others, is that when the pressure is put on in their English or Math class, they withdraw their attention from Computer, even if they care about "doing well" because any other choice would be a bad economic decision. That observation is translated by still other students, particularly in the Intro course, (those who don't care in the least about our subject matter) as a chance for an easy credit. These are not all losers but their motives and expectations are certainly dubious, and they further inflate our already unservable enrollment requests. In an attempt to eliminate them, we are sometimes tempted to offer our courses without any credit at all, so that there can be no purely economic motivation for taking Computer courses. But credit is the currency of education, and no-credit courses are luxuries that only the credit-rich can afford. Students who are not so fortunate as to have earned more credits than they need for graduation could not "waste" their earning time taking a no-credit course, even though they might have serious interest and might both contribute to and benefit from the course.

## What kind of "learning environment" is right?

I certainly know what kind of class *I* want to teach: I know what I value and don't value in thinking, and I also have psychological theories about how people reach the goals that I value. Together, these and a few practical matters dictate a pedagogical style that I prefer. I want to teach students how to explore intellectually, how to be researchers and scientists, to be "an idea hacker" (though I don't at all care whether they become computer hackers or what specific intellectual territory, e.g., linguistics, mathematics, psychology, they hack). I even believe I know how the computer can help do that very well, better than it could be done without the computer. And my school is so supportive that it has allowed us to teach a course aimed specifically at this purpose, even devoting three teachers co-teaching to a mere twelve students!

But I have not answered my question. The course I just described is not the Intro one that has the 64K students applying to it. The one thing our school (rightly?) does not allow us to do is drop the course that is in such heavy demand. What kind of "learning environment" is right for students who sign up in droves but are *not* asking for Linguistics (enhanced by the computer) or Mathematics (enhanced by the computer) or any other specific Intellectual Discipline (enhanced by the computer), but Computer (whatever that is).

A client-centered approach might have us try to figure out what Computer is and teach it. Alternatively, we could resist the temptation to meet the need as it is expressed, and try to meet the need that we, with our greater sophistication and wisdom, believe is there--converting the heathen or curing the ill, depending only on your choice of metaphor. Yet another approach, already mentioned, is to accept only the already converted, admitting only those students who are signing up for "pure" reasons.

Possibly not for the explicit purpose of being client-centered, the "literacy" crowd has taken the first approach. Computers are going to be part of everybody's life, and any "relevant" curriculum must acknowledge the need of students to be prepared for their own inevitable future. Educators differ about whether this course is more like social studies or mathematics, more like driver education or like auto shop, but students and teachers in this camp are working harmoniously toward a future they all suppose is around the corner--probably something like everyone programming their mortgage payments or Fahrenheit to Celsius conversion in BASIC. The orthodox Logoite vows never to give in to computer literacy, to teach or even tolerate BASIC, or to use Fahrenheit to Celsius conversion as an example of anything at all, but I, for one, am less and less convinced that something of this sort isn't exactly what at least some high school students "need." (Oh, there can be no doubt that the Logo language, as a computer language or as a tool for thinking, is vastly superior to BASIC, and, all else being equal, one might as well choose the best tools regardless of the purpose of the Computer course. But *if* we agree that we are teaching a social study and *if* we are not operating with some ulterior motive like teaching clarity of thinking, or developing a computer science perspective, the difference is really not so critical, is it now? But don't tell anybody that I said this.)

My waffling position on the client-centered literacy course is thus not so much philosophical as personal. *I* don't find the course interesting enough to teach. But I bet my students would be happy if I did.

As for the second approach, I am just enough of an evangelist to want to convert people to clearer ways of thinking. The courses I want to teach in our high-school would be titled something like "Transformational Grammars" or "Recursive definition and mathematical induction" or "Abstract Algebra" or "Writing for Publication" and they would use the computer because the computer is the best tool with which to manipulate the subject matter of the course. In the first three courses I named, the student would use Logo programming as a medium for laboratory investigation of the topics; the writing course would take advantage of the computer as word-processor. But expectation gets in the way. People who think that they want a course *about* the computer don't sign up for these other courses. Besides, as one official in our school put it, these courses are too specific -- what about the kid who wants a general introduction? I don't believe that there is a general introduction that is worthwhile, and I don't mind much asking kids to think hard, but the very large numbers of kids who want *something* causes me a problem. Though I believe that the need that they feel is pumped up by irresponsible TV advertising, misguided popular press, and general educational paranoia, their *need* is real. Even if I were to conclude that the program that would serve them best would most closely resemble a kind of personal psychotherapy to reduce their anxiety about the computer revolution, I must acknowledge that these students represent a large and *needy* clientele. That may not obligate me personally, but does, I believe, obligate the school.

I waffle here, too. I know that some of my Introduction to Computer students would enjoy specifically focused, intensive courses very much, and would be even happier if they found themselves in these courses rather than in the Intro course that they chose. I also know that many of the students who would benefit the most from courses such as these would not have chosen them.

Why is all this important in thinking about the learning environment?

Someday, we will have computers aplenty and everybody (all teachers, al students) will be comfortable with them. At that time, issues of learning environment will again be what they always have been: concerns o pedagogues, psychologists, and students, but little or no more so or account of the computer than on any other account. Oh, there will continue to be concerns about the autistic person-machine interaction tha computers can foster, but such issues antedated computers and are only slightly exacerbated by them. Besides, classes in Transformational Grammar or Writing for Publication don't sound to me like they would discourage classroom interaction. It is also true that with computers one can create new learning environments, like our computer center, in which students teach themselves and each other in a totally (maybe overly) oral culture (nobody reads the C language manual or the Logo manuals) and even communicate via the computer with people in other institutions. But here, too, the pedagogical issue is not new; interaction, communication, and student initiated independent learning are all possible without computers.

Right now, however, there are not computers aplenty, and hoards of people are distinctly *un*comfortable with the few computers there are. That means that there are many more people trying to get into computer courses than can get in, and more, in fact than really want to get in. That means that some students who really do want to get in are being kept out by others who were luckier in the draw. Unlimited access to limited machines means that the most aggressive (the boys) will win. Restricted access loses the very features that we believe have helped to create the excellence that is the hallmark of our center. Preselection to gain access drives up the value of competitive behavior and also selects out many students who should have the opportunity but can't demonstrate that before the fact. No selection -- attempting to provide for everyone -- requires either great wealth or vastly watered down services (e.g., restricted access). Unrestricted interaction among students in the computer laboratory breeds the wonderful learning environment that our best hackers develop in. It also drives the not-so-brave newcomers right out of the room and makes *teaching*, when it is needed, utterly impossible.

No. "What is the right learning environment" is altogether the wrong question, as clinically sensitive educators have known all along. Designing leaning environments is not a matter for theoreticians but for clinicians who organize their minds along theoretical lines, but organize their practice eclectically, knowing that people are more complex than the best of our current theories can model. For our hackers, we already have a wonderful, if not perfect learning environment. For other students, we need another environment. For still others, still others.

I'm waiting for the day when computers will be ubiquitous, and facility with them will be widespread eliminating the need to talk about *a learning environment specific to the computer*. At that time it will again make sense to think freely about environments (plural) for students to learn in.

It can't be too far off. After all, in a mere ten years today's 15-year-olds will be 25.

# EXPLORATIONS IN MATHEMATICAL THINKING: SOME IMPLICATIONS FROM LOGO CLASSROOMS

Richard Noss
Advisory Unit for Computer-Based Education
Hatfield, Hertfordshire, U.K.

For the past two years, the Chiltern Logo Project[1] has been investigating the ways in which primary-school children learn to program in Logo. In this paper, we aim to do three things:

1) Outline the approach of the project, and to give some flavor of the role that we have found for Logo in five UK primary classrooms.

2) Offer some generalizations, based on our observations, of the ways in which children have learned to program, and to suggest some implications for the development of a teaching strategy.

3) Suggest some avenues for future research.

## The project's approach

The approach of the project has been determined by our wish to examine the role of the Logo programming within the natural classroom environment, and to integrate the teachers into the research by the formation of a 'project-team' who could learn from each other, as well as alongside their children. Five schools were chosen from a variety of geographical areas, (two 'inner-city,' two suburban, and collectively incorporated children from a wide spectrum of social and cultural backgrounds.

The children (aged 8+ to 10+) programmed in pairs of threes (latterly exclusively in pairs), for a median time of about 75 minutes per week (35-40 hours per year). During the first year, 118 children were involved in the five classes. Programming took place throughout the school day, irrespective of the specific activities in which the rest of the class were involved. This arrangement offered a relatively high degree of machine-access to the children despite the provision of only one computer per classroom. It also

[1]This project is funded by the Microelectronics in Education Programme, UK

allowed the teacher to incorporate the children's Logo work into his or her normal teaching. children were able, if they desired, to link their normal classroom work with Logo activities, by choosing programming projects which related to other school-work, or by contributing to the Logo projects of others in the class.

The work has been based on an unstructured approach which has encouraged children to pose their own programming projects, and to talk and maintain control over their learning. Within this framework, the project teachers have developed strategies which were appropriate for their own teaching styles. These approaches were based on a policy of responding to children's programming needs, and of minimizing active intervention in the learning process.

## Research issues

We have adopted an illuminative research strategy based on participant observation. Our central concern has been to observe children as they learned to program, and to pursue issues as they arose, rather than to test hypotheses in any formal sense. The key objective has been to illuminate the kinds of thinking which children engage in as they explore and apply the powerful ideas of programming in Logo. In doing so, we have examined the extent to which children programming in Logo can be said to be 'doing mathematics' in Papert's sense. There are two aspects to Papert's claim. Firstly that learning Logo programming can develop a 'mathematical way of thinking;' secondly, that such thinking may serve as a basis for learning the content of mathematics [Papert 1972]. Throughout the first two years of the study, we have concentrated on the first of these aspects, and will focus on this in what follows. Latterly, we have begun to look more closely at the second aspect. We will return briefly to this issue when we outline our future research plans.

We have interpreted the idea that learning Logo develops children's mathematical thinking, in two ways. The first concerns the content of the mathematical ideas embedded within Logo (e.g. procedures, inputs, recursion, etc.). To this end, we have analyzed the nature and extent of the children's encounters with the ideas of the language, and examined children's strategies in gaining power over them [Noss 1984]. We are aware that there may exist a gap between the mathematical content of these ideas as perceived on the one hand by mathematicians, and on the other by the children, and our investigation of children's mathematical perceptions.

Secondly, the study has focused on the process aspects of mathematical thinking. By process aspects we mean activities which underpin the way in which mathematicians do mathematics. For example, generalizing (extending the scope of an idea), specializing (seeking special cases), conjecturing (seeking pattern and structure as a basis for generalizing), and verifying (testing the truth of a generalization or conjecture) [see Mason et al. 1983]. Our interest has focused on the ways in which the search for structure and pattern within a programming context has provided an environment for the formalization of mathematical thinking.

We have also been interested in developing strategies for intervening in the learning process, and the involvement of five teachers with as many different teaching styles has offered some insight into the effects of such styles on children's learning. Through examination of both teacher and researcher interventions, it has been possible to provide some preliminary generalizations on the construction of intervention techniques in the building of an effective Logo environment.

## Children's learning modes

Our early work indicated that many children required a relatively long period of introductory activity with a floor-turtle. Although no attempt was made to 'conceal' procedure-definition, less confident children spent a considerable time drawing in direct-drive. During this time, a number of generalizable strategies emerged. These included that of 'homing-in' on desired lengths or angles (during which important mathematical and programming ideas were explored), the self-imposition by students of restrictions on problems (in order that they could be solved with current knowledge), and the adoption of 'deplanning'strategies, in whcih end-goals were adapted or errors incorporated as a precursor to debugging strategies later on). Teacher and researcher intervention was minimal, and children were introduced to new ideas only as their need arose. Many, especially the slower learners, needed time to get a feel for the ideas, to handle and gain control of them. Further details of this are available in Noss [1983a].

We have characterized these early months as a time in which the children were making sense of the fundamental ideas of control over the machine, and the essentials of turtle-geometry. Subsequent work has suggested that this kind of activity, which we have called 'making sense of', has recurred naturally as children encountered other, more sophisticated Logo ideas. For example, a child making sense of the idea by playing with the syntax, trying 'extreme' values, or using outlandish names.

We have identified two further approaches which, together with making sense of new ideas, constitute three distinct strategies, or learning 'modes'.

The first of these we have called 'exploratory' in order to convey a sense of children conjecturing the effects of particular actions ('What happens if...'). Such activity has been a valuable means by which children have forged links between their new and existing knowledge. Typically, this has made considerable use of Logo's extensibility, extending an idea by incorporating it into new procedures, or by integrating it with other programming concepts with which the child was already familiar. Such strategies have been particularly important in generating for the child a sense of ownership of the ideas involved.

This kind of exploratory activity is in contrast to the goal-directedness of the 'problem-solving' mode, which is concerned with finding solutions to problems ('How do I get the computer to...'). Such problems were invariably posed by children themselves, and formed the major component of the programming activities we observed.

While we do not underestimate the importance of the Logo environment for the posing and solution of problems, our work has led us to place a high value on the exploratory mode. The remainder of this paper will suggest a rationale for this emphasis, and examine its implications for the development of intervention strategies, and the focus of future research.

Exploration has helped to generate a learning environment in which hypothesis and conjecture are the basis of Logo activity. It has offered children the opportunity to experiment with the structure of the ideas embedded within the language, and to provide a basis for the twin activities of specialization and generalization which are the essential characteristics of mathematical thinking. We have noted that children switch learning modes freely. At times, exploration has been an essential precursor to problem-posing, and thus to problem-solving. On other occasions, children engaged in a project have switched into an exploratory mode which acted as a stimulus for problem-solution.

This emphasis on the exploratory learning mode is not intended to devalue goal-directed activity. On the contrary, our contention is that both exploratory and goal-directed programming are key components within the problem-solving environment created by Logo. From this standpoint, the two kinds of activities become modes in which children are doing mathematics, either by solving problems which utilize Logo's content and structures, or by exploring the relationships between them. We suggest that such a viewpoint adds weight to the view of programming as a 'new and powerful operational universe for mathematical experiments' [Feurzeig, Papert, et. al. 1969]

## Intervening in the learning environment

In general we have found that helping children to learn Logo has provided the project-teachers with the opportunity to observe and reflect on their children's learning. However, we have noted one particular issue which needs to be addressed by the Logo community in the UK. Within the framework of relative autonomy which UK primary schools enjoy, there remains a strong tradition of formal, non child-oriented education. Our experience suggests that teachers in such schools can be introduced over time to the idea of subordinating teaching to learning. However, for such teachers. teaching may be synonymous with instruction: they may have difficulty in perceiving any alternative. There is thus a danger that the message of Logo is interpreted as being one of abdication from teaching altogether. For this reason, we have viewed it as important to study interventions more carefully in order that alternative and fruitful models can be offered to teachers who require it.

While all but a very few children learned to program at a functional level (define and edit procedures, interpret error messages etc.), many children found a number of Logo concepts and processes difficult to use. In particular, the concept of modularity is, as Leron [1983] has pointed out, one with which many children have considerable difficulty. In particular, our work has illustrated that (at least in the computer-poor culture in which we are operating at present), the idea of modularity is far from natural for most children. In a more general context, we have encountered many children who have reached "plateaus" in their programming, either in terms of their unwillingness to pose problems which demand new ideas, or a reluctance to explore unfamiliar territory. Such plateaus have arisen independently of the ways in which children switched between learning modes (two further viewpoints of a similar phenomenon are provided by Solomon 1982 and Rampy 1983).

It seems clear that the plateau-problem is only very rarely overcome without help (such help does not, of course, have to emanate from the teacher). We have therefore recently focused our interest on gathering case-study material from a small number of relatively Logo-experienced children (aged 10+), in order to examine in detail their acquisition of new ideas. and to investigate the kinds of researcher-intervention which prove fruitful. For a subset of the children, this has included the introduction of list-processing activities.

The central difficulty has been to find ways of initiating intervention, while keeping within our policy of letting control rest with the child. A [preliminary analysis of the case-study material has suggested that effective help in the learning process does not need to be restricted to responsive intervention (responding to questions from the student), but can usefully include interventions initiated by the teacher or researcher. It has been proposed elsewhere that such intervention may be based on the posing of problems by the teacher [Bull and Tipps 1983]. Our suggestion is to include in the teacher's repertoire of intervention strategies, those which are based on exploratory rather than goal-directed activities.

The challenge has been to mesh the responsive/initiated dimension of researcher intervention, with the exploratory/goal-directed dimension of children's learning modes. Our findings suggest that researcher/teacher initiated intervention may be helpful in helping children to explore a new idea or process which they may otherwise either ignore or reject. Such exploration may be guided by, for example, suggesting sub-goals ("Why don't you try . . . ") or motivating hypotheses ("What do you think would happen if . . ."). On the other hand, responsive intervention is most effective when it arises out of a need expressed by children in the course of goal-directed activity.

In contrast, initiating intervention during goal-directed activity ("Let me show you a good way to do this"), or responding too positively to the "What if . . . " questions of the exploratory mode, have tended, not only to wrest control from the child, but also to be counterproductive. The situation is summed up in the figure below, which is intended as a focus for further discussion, rather than a prescriptive model for an intervention strategy.

| | | LEARNING MODE (child) | |
|---|---|---|---|
| | | Exploratory | Goal-directed |
| INTERVENTION (researcher) | Initiated | 1 | 0 |
| | Responsive | 0 | 1 |

## Implications for Research

Our emphasis on the exploratory learning mode, may help to counterbalance the view that Logo learning consists mainly of acquiring problem-solving heuristics such as planning [see e.g. Pea 1983]. On the contrary, our contention is that the processes of mathematical thinking which are offered by learning Logo are heavily dependent on the scope which the language offers for exploration and experimentation. In consequence, we are convinced that there is no substitute for continued longitudinal studies of children learning Logo in their classroom environment, in order to further illuminate the relationship between programming, and the opportunities it affords the learner to explore the underlying mathematical concepts and processes.

At the same time, we feel ready to tentatively probe the possibilities of children using Logo learning to explore and experiment with mathematical ideas in the context of the more conventional mathematics found in secondary schools. Preliminary interview-transcripts with children (aged 10+) who have had 55 hours Logo experience over 18 months, have suggested that children are capable of making use of their Logo knowledge to make interesting algebraic generalizations in a non-Logo context. We do not expect such "transfer" to take place automatically. Neither are we concerned to compare some Logo-based algebra curriculum with a non-computer one. Rather we are hoping to examine ways in which children who have experienced the kinds of mathematical metaphors embedded in the Logo environment, can be helped by teachers (particularly those who have shared in their experience) to make use of such ideas in the context of exploring and investigating concepts and problems of the mathematics curriculum. It is hoped in this way to throw light on the role of Logo as an experimental and investigative tool in the learning of mathematics.

## Acknowledgments

I would like to thank Dr. Celia Hoyles for her helpful comments on an earlier draft of this paper.

## References

Bull, G. & Tipps, S. Problem Spaces in a Project-Oriented Logo Environment." *The Computing Teacher*, December 1983.

Feurzeig, W., Papert, S. et.al. "Programming Languages as a Conceptual Framework for Teaching Mathematics." Bolt, Beranek and Newman Inc., Cambridge, MA, 1969.

Leron, U. "Some Problems in Children's Logo Learning." *PME* 7 (proceedings). Israel, 1983.

Mason, J., Burton, L., Stacey K. *Thinking Mathematically*. Addison Wesley, 1982.

Noss, R. "Starting Logo: Interim Report of the Chiltern Logo Project." *AUCBE*, 1983a.

Noss, R. "Doing Maths While Learning Logo." *Mathematics Teaching* 104, 1983b.

Noss, R. "Children Learning Logo Programming: Interim Report #2 of the Chiltern Logo Project." *AUCBE*, 1984.

Papert, S. "Teaching Children to be Mathematicians versus Teaching about Mathematics." *Int. J. Math. Educ. Sci. Tech* 3, 249-262, 1972.

Pea, R. "Logo programming and problem solving." Bank St. College, 1983.

Rampy, L. "The Programming Styles of Fifth Graders Using Logo." Paper presented at the N.E.C,C. Baltimore, Maryland, June 1983.

Solomon, C. "Introducing Logo to Children." *BYTE* Vol. 7, No. 8, 1982.

# FROM THE CLASSROOM TO THE LABORATORY

Joyce Tobias
Public Schools of Brookline, Mass.

While the ideal in a Logo environment is to have one computer per student available for student use during and after school, this ideal seldom exists. The norm in a school setting is usually one computer per classroom. This paper will examine the three most common computer environments in which Logo occurs and provide some guidelines to help you organize your classroom for learning and working in a Logo environment.

Generally, three models of computer environments have developed in schools: individual classrooms with one or two computers, library media centers or resource rooms with several computers, and computer laboratories as well as combinations of these models. In some research settings a fourth model is found in which one computer is available for every student in the classroom. This is an "ideal" but is not a likely model for the near future.

Classroom Model: In this setting one or two computers are usually situated in a classroom. Sometimes these computers may be shared by a number of classrooms. Students are scheduled to use the computer while other competing activities are going on in the classroom. Having students work in pairs is advisable so students will have someone to help when creating or debugging their activities or procedures. A student can be appointed a Logo helper for the day or week to answer questions and assist the teacher as needed.

Papert maintains that one-computer per classroom is insufficient to create a Logo environment, however, it's a beginning. Students working in this environment will need to spend additional time working with Logo away from the computer: planning projects, keeping a comprehensive journal, watching teacher demonstration lessons, and taking notes. Hands-on computer time will have to be maximized. In this and other similar situations involving scarce computer resources, care must be exercised by the teacher to prevent Logo from becoming a lock-step, cookbook programming course which fails to meet the philosophical considerations of a Logo environment.

Library/Media Center Model: This setting usually finds several computers located in the library and supervised by the library staff. Students are scheduled to use the computers at regular times. Again, pairs of students are advisable. In some situations, students are sent to the library for their hands-on time while the teacher remains in the classroom. In this model the library staff has usually been trained, or needs to be included in staff training on using computers and Logo. It is essential that all persons involved in the Logo program understand and share similar philosophies and have frequent opportunities to share observations and plans with one another. In cases where teachers are able to accompany their students to the library they may prefer to plan several simultaneous activities for students as the class rotates through their computer time. Some alternative activities may include library related activities, such as book discussion groups, free reading, research, and materials selection or students can use this time for planning new projects or debugging old ones.

Computer Laboratories: This is the model that comes closest to the ideal of "one computer per child." Teachers and students are all involved at the same time. When a student discovers something the entire class can immediately share it. When a student is having a difficult time debugging a program, the entire class can take part in the process. This model provides the most effective environment for learning and sharing the Logo microworld. In elementary schools most laboratories have been organ organized so that no more than two students share a computer.

In the library and laboratory models reviewing, assignments, and demonstration lessons should occur in the classroom *prior to* a laboratory period. Time on the computer needs to be maximized. For any worthwhile educational experience to occur, the maximum number of students at a computer should be two.

Students should be provided with *regular, frequent time* on the computer every week rather than a short intensive mini-unit. A regular 4-5 day/week exposure is the most beneficial. Students should have a *minimum* of 30 minutes on the computer if you expect a child to be able to develop his ideas. When students work in groups, each partner should share in the planning, exploring, and keyboarding experiences.

Students need to keep a notebook or journal or all their work. If worksheets and handouts are to be used, a three ring binder is the best vehicle. This allows additional pages to be added as appropriate. The notebook or journal might contain:

1) records of student explorations

2) plans and possible procedures for their projects

3) possible outputs of sets of commands or procedures

4) rough sketches of project ideas--(best done with graph paper)

5) classnotes

6) a bug or error message log

7) handouts on using Logo

Logo is a microworld that is meant to be shared. Children learn from one another, need time and space to think through solutions, and benefit from sharing "their" solutions and problems with one another. Watching the excitement generated in a class when one child makes a discovery and share it with classmates helps everyone actively share in the learning process. Logo does not isolate children but creates opportunities for cooperative efforts. Interaction is a key element of successful learning experience. This is also true of a Logo environment. Respect for one another's efforts must be maintained and fostered.

There are appropriate times for group projects as well as individual projects. Development of an "adventure" game is a marvelous activity for a whole class project. In this type of program, the final product often combines individual projects into complex scenes. Students benefit from both these individual and group experiences.

There are no limits in a Logo environment as far as programming is concerned. Students should be encouraged to proceed at their own rate and not be held back to wait for the rest of the class or the teacher. Teachers need to use an open approach, recognizing that some students are able to forge ahead, and work abstractly while others take longer to experiment and need more time to work concretely. Students should be encouraged to explore and search for connections to other curricula.

In no other area of education has the phenomenon occurred in which the elementary child can know more than the teacher. Yet, the teacher need not be intimidated by this knowledge, the teacher holds the key as questioner and guide. It is the teacher who helps the child reason, probe, and analyze--while not giving all the answers. Asking the right question, suggesting the child rethink a process, check spelling, spacing, or other details of their work is of more value than supplying an immediate answer.

Teachers provide strong support through their knowledge of the learner's developmental level and their experience with the child's learning style.

While children often have more "programming" competence, they lack the judgment and honed reasoning skills of the teacher. Allowing a child to explore, means allowing the child to try different solutions, strategies, make some missteps, and react to unexpected outcomes. Allowing the child to experiment can result in several advantages. The student learns that <u>every</u> plan does not always work smoothly the first time, and that unexpected results can sometimes be very interesting (and not necessarily wrong). This opens new avenues of exploration for students, and gives them an opportunity to analyze and reflect on why something happened.

The teacher keeps this process flowing by providing encouragement and intervention before frustration sets in. It is okay for students to know more than the teacher. Take advantage of your student's expertise in Logo to help other members of the class. Two cautions need to be noted, however, as accelerated students still a) need their own allotted computer time, and b) need to understand that their role is one of questioning and guiding, not doing.

Logo encourages new dimensions in children's relationships with their teachers--one in which the fine line between teacher and learner blurs. Students find learning fun, relevant, and intellectually challenging. They are not afraid of technology. The Logo microworld provides an unusually humanistic introduction to the world of technology.

# LOGO AND YOUR FAMILY

Griff Wigley
Family Computing Inc.

GRIFF WIGLEY IS UNABLE TO PARTICIPATE IN THE PANEL "THE LEARNING ENVIRONMENT: FORMAL & INFORMAL" DUE TO A SCHEDULING CONFLICT

## Introduction

Is Logo any different than any other educational activity that families engage in? If so, how? What is its potential as a tool to promote learning in the home environment? This paper attempts to examine these and other questions.

## My experience

My experiences with Logo are from two different perspectives. I have taught Logo to many children and adults over the last two years, including approximately fifty parent/child pairs (children ages 3 to 15). As a husband and father, I have learned Logo along with my wife and fellow teacher, Robbie, as well as with two of my three sons, Collin age 7, and Tyson age 5.

## Parent Modeling

While it's true that a child's school experience in large part is the determining factor in his or her attitude towards learning as a parent are equally important factors.

There are two distinct kinds of parent modeling regarding learning. Probably the most common involves your support of the school's endeavors. This would include helping with homework, attending conferences and school related activities, expressing interest in report cards and test scores, and in general, making statements and acting in such a way to indicate to your child that his or her school life is important.

The second kind of parent modeling may be in addition to or exclusive of the first. It involves a parent activity that indicates an interest in learning in and of itself, more closely akin to the feeling typically associated with the word "hobby", as opposed, unfortunately, to the word "study." Learning activities are engaged in for the pleasure they give, not out parental duty or responsibility. In fact, you can distinguish this kind of modeling from the first by determining if you would enjoy engaging in the activity if there were no children around. Examples include getting books and other materials from the public library both for yourself as well as for your children, and engaging in conversations and activities that reflect your delight and/or curiosity in a topic.

Computer activities which are interesting for both you and your child fall into this category. The "informal learning" software companies have clearly found a huge market for software that promotes the kind of learning I'm talking about. Programming, and in particular, Logo programming, is particularly suited for this kind of learning as well.

## Logo's range

Logo is infinitely adaptable to any level student. This feature allows it to fit in a family with more flexibility than most other software. In my family, our interests vary greatly, and so do our Logo interests. I'm currently much more interested in Logo's list processing abilities, while Robbie continues to have a strong fascination with turtle graphics. The boys are, of course, primarily interested in the turtle and sprite graphics.

Describing our Logo interests in this way paints an individualistic picture that's only partially accurate. We are a family that values both separateness and togetherness. Robbie and I both work separate much of time, yet we collaborate on each other's projects quite often. The boys, on the other hand, nearly always want to work with one us when doing Logo. They even love it if four of us are crowded around the computer. It becomes a social activity in much the same way as a game of cards. As their Logo skills increase, they may begin to work on projects together without parent involvement, much as they do now with other games and projects. As they get older, and if Logo remains an interest, they possibly will engage in separate Logo programming projects. At first glance then, Logo seems no different than any other interesting activities that families engage in. But look closer.

I know of a family whose members all play different musical instruments. I always envied them because I saw how music played this flexible role in their lives. Parents and children each play different instruments and enjoy solo performances, yet they often perform as a group. And their skills progress as they grow older, allowing for more satisfaction both individually and as group. It seems that Logo has this potential as well.

It can grow with us, not be outgrown by us. As each member's skills increase, everyone can benefit. "Duets" and "solos" alike can be "performed." It can become a common thread for conversation. It can have a history and a future with us.

## Conclusion

All this is speculation, of course. I'm seeing potential, not reporting facts. Logo may be replaced by something better. Our interests may go in opposite directions of Logo. But it has touched our lives in a way that will last forever, whether we stay connected to it or not. Computers and Logo have sparked a love of learning in our house - and that has made all the difference (apologies to Robert Frost).

# IN PRAISE OF FINGERTIPS

John R. Allen
The Lisp Company (TLC)
and
Ruth E. Davis
University of Santa Clara

## Introduction

Mathematics and logic teach certain formal ways of thinking about the world. Rhetoric teaches less formal, but equally important, techniques for articulating one's thoughts. Much of current Logo practice misses on both accounts: bad mathematics and atrocious style. Unless the situation improves, Logo will take its place in history next to the New Math.

* Locally, we believe the notions that make Logo of interest have not been understood--there are deep issues of notation and expressibility which Logo attempts to draw upon, but all too often comes up dry.

* More globally, we believe that the purpose and difficulty of programming is sorely undervalued--more serious attention must paid to underlying ideas and as a result, the quality of preparation.

* Finally, and in conjunction with the last two points, Logo is victim of a myopic view of the world that pervades the computing field in general. This defect of vision spans the spectrum from structured programming to hacking.

In this paper we crystallize our concerns around a paper of the late Derek deSolla Price, "Of Sealing Wax and String." We argue for "enlightened pragmatism" based on a much stronger grasp of theory and substance than we have seen in current Logo work.

* * * * *

1984 certainly began with a Bang.

The opening salvo appeared in the January issue of *Science84*. There, Steve Olson interviews E.W. Dijkstra in "Sage of Software." He reports:

> For more than 20 years, Dijkstra has been fighting against the kind of programming that inevitably leads to bugs in computer software. To him, the way organizations like NASA program computers is foolhardy at best, and perilous at worst. He believes

> there is another way, a better way. It involves structuring how a person thinks about programming so that programs themselves acquire a firm mathematical basis.

And again:

> Computing science definitely went sour with the birth of the computer industry, which designed its products in a nonmathematical environment for a nonmathematical market... In this nonmathematical environment, debugging became the accepted way of programming.

Dijkstra continues.

> There are two views of programming. In the old view, the purpose of our programs is to instruct our machines; in the new one, it is the purpose of our machines to execute our programs.[1]

Speaking as practicing computer scientists and people highly supportive of a mathematical theory of computation, we find it hard to imagine that we'd yet be to the moon if we'd practiced the rigor that Dijkstra extols. The problem of building complex system has its answers on a much deeper level than just mathematical correctness. The answers come from within a individual (or set of individuals) who are competent and confident in their trade of programming, and who can draw upon a strong blend of theory and practice. Dijkstra's view of structured programming--what we might call the "scientific method" of programming--does not directly address these problem-solving issues. That "non-mathematical environment" to which Dijkstra alludes is called the real world; and the real-world has side-effects and bugs.

It would seem that an appropriate response to this structured view of the world could come from the Logo community. Here's a language supposedly based on Lisp--a language based on theory but hardened with the realities of real-world programming ... But no.

"Pop!" goes the January issue of the *National Logo Exchange Newsletter* where we find Glen Bull and Steve Tipps in "Problem Spaces in a Project-Oriented Logo Environment" stating the following:

> The Logo community has avoided describing Logo projects [in content areas] for fear that descriptions might become recipes --explanations that shift into rigid lockstep programs deemed to be 'the right way to do Logo'...

---

[1]EWD512--from *Science 84*

Is Logo's philosophy of learning really so ethereal that the mere act of committing other's experience and applications to paper might damage it? Is this really discovery learning? If each generation had to "discover" the previous one's results, we'd also not yet be to the moon.

And while we would agree with Papert's *Mindstorms* that:

> The debugging philosophy suggests [that] errors benefit us because they lead us to study what happened, to understand what went wrong, and, through understanding, to fix it

we find the continuation of that passage harder to defend:

> Experience with computer programming leads children more effectively than any other activity to 'believe in' debugging.

Such a claim begs for substantiation, as do other claims we'll address later.

And so the battle lines are drawn-- programming takes no prisoners. Just when we're beginning to doubt the existence of common sense, the January issue of *Natural History* arrives.

With a sigh of relief, we read in Stephen Jay Gould's "Just in the Middle:"

> I have often been amused by the vulgar tendency of the human mind to take complex issues, with solutions at neither extreme of a continuum of possibilities, and break them into dichotomies, assigning one group to one pole and the other to an opposite end, with no acknowledgment of subtleties and intermediate positions--and nearly always with moral opprobrium attached to opponents.

A light in the darkness!

We read on. And later in this issue of *Natural History Magazine*, our persistence pays off. We find Derek deSolla Price's "Of Sealing Wax and String;" there he argues that the textbook view of scientific discovery--"The Scientific Method"--misrepresents the real sequence of events. He states that it has been curiosity, in the guise of experimentation, that has driven theory; not theory that turns to experimentation to corroborate hypotheses. This struck a responsive chord.

We've seen the effects of the scientific method in mathematics, where authors, in the name of elegance, obliterate all traces of the thought processes that went into their discovery. It is important that we not let such behavior move into the domain of thinking about complex system--i.e problem-solving and programming. This seems like a natural opening for the discovery learning position advocated by Logo devotees. And yet there are indications that is not working: the position mentioned by Bull, the quality of the materials being produced to support the ideas, and as we'll argue later, the quality of the ideas themselves. But still, there's a deeper problem, beyond the particulars: how or why does discovery wòrk at some times but not at others? Let's go back to deSolla Price.

deSolla Price drew his examples come from the traditional scientific disciplines, not from mathematics, and not from computer science. He spoke of "a band of ingenious craftsmen, with brains in their fingertips, who exploited a great many little-known properties of materials and tricks of the trade." He continues:

> These tricks not only made all the difference in what could or could not be done in a laboratory; to a large extent, they determined what was discovered.

Lightning strikes!

If we could think of individuals in the computing profession who met these criteria of "brains in their fingertips" and analyze their contributions and their surroundings, perhaps we could isolate some of the factors that we should emphasize in future learning environments. Though there are certainly others, one particular individual came to our minds: Steve R. Russell.

Steve Russell was one of the original members of the MIT Artificial Intelligence Project, working as an assistant to John McCarthy. As McCarthy developed Lisp, it was part of Steve's job to understand how to implement it. The initial plan called for an Algol/Pascal-like syntax with data being represented as parenthesized lists (as is currently done). The implementation of the language was expected to involve a compiler and a run-time mechanism--what is now considered "typical," but at that time (1960) was a major undertaking. The only large compiler project that had been completed at that time was Fortran. However, one of the theoretical results that McCarthy had established was the "universality of Lisp computation." This was a common exercise in mathematical logic; the essential idea was to show that there was one computing device that could simulate the behavior of any other device. The critical step involved encoding any program so that this super program could simulate its behavior. The critical ingredient of the notation was an ability to represent any program as data--as something that the master program could operate upon.

Since Lisp's data objects were lists, McCarthy represented programs as lists; he then produced that single omniscient program to mimic any Lisp program's behavior. McCarthy's only intention was to demonstrate that Lisp was eligible for a mathematical pedigree. Steve Russell recognized that McCarthy's Lisp function--called eval--contained the specification for a practical Lisp interpreter. It was Steve who implemented eval (over McCarthy's misgivings); it was Steve who recognized that the external Algol-like syntax was really not necessary for Lisp programming. Needless to say, Steve prevailed; the interpreter was completed in short order; the compiler project was dropped, to be revived as a program written in Lisp rather than hand-coded in assembly language. The rest is history.

As the AI Project personnel became more fluent in the language, more and more ambitious programs were attempted. And, as with any experimental system, there were limits in the initial Lisp implementation. One feature that Lisp advertised was the notion of functional objects. By way of comparison, a traditional language deals with numerical objects; we can perform operations on numbers, creating new numbers; we can use numbers as inputs to programs that give numbers as their outputs. Lisp said it would accept functions--i.e. programs--as inputs and could produce functions as outputs. But this Lisp's eyes were bigger than its stomach. Simple applications of these functional objects worked as expected, but one graduate student soon discovered a bug. From student to McCarthy came the bug--"it's not a feature, it's a bug;" from McCarthy to Russell came the request--"fix it, please." From Russell came the solution, called the "funarg hack" in Lisp parlance. This happened in 1961 and, over the next ten years the solution was understood to be the answer to some reasonably deep theoretical problems in computer science. In fact, even twenty years later many languages still "don't get it right."

Out of this primeval soup called Lisp came many other modern ideas: (1) garbage collection--the automatic management of storage; (2) first-class objects--the idea that objects exist independent of some specific storage management; and (3) the beginnings of graphical languages--Lisp's list-notation is a linear form of a two-dimensional (tree/graph) notation. And of course Lisp has become synonymous with Artificial Intelligence in the United States.

Steve Russell's fingertips were busy in domains other than language design. In 1961 he invented a game called "Space," widely recognized as the first video game. Written for the DEC PDP-1, one of the machines designed with interactive programming in mind, Russell's game demonstrated the power of simulation for instilling a good sense of physics. Incidentally, of course, it was fun to play.

So how do we characterize Steve? Not a theoretician; not a programmer *per se*. Truly an individual with "brains in [his] fingertips." He had incredibly good insights and a rich environment. And that environment was true discovery learning, not a warmed-over "sharing experience." The key, it seems to us, was that Steve worked in an environment where the goals and motives were understood by those in charge. It was a question of leadership, not management. And those critical ingredients of understanding and leadership are too often missing in "discovery" settings--not just Logo and programming, but more globally.

If those in charge do not understand the concepts, then discovery learning becomes a fraud. How many times must students rediscover the number system, or the wheel, or table manners? How worthwhile is it to have students rediscover modularity under the guise of "subroutine," "procedure," or "function?" Where does discovery learning stop and sadism begin?

We need to develop a program that strives to develop that notion of "brains in fingertips"--an enlightened pragmatism--if computational devices are to really have a deep impact on the way we think and learn. We don't see the present delivery of Logo working towards such a goal; and it is not from a lack of results upon which to draw.

There is a strong theoretical heritage that computation can draw upon for the "brains" side. It is theory that abstracts, clarifies, and explains one iteration of experience to give us a toehold for a new iteration. More generally, it is theory that gives us a way to gauge the aptitude of a potential disciple. It is the basis of mind- training, like calculus is for mathematics.

For the "fingertips," there is a growing body of empirical work that can be integrated into such a program. Of particular note is the "flavor" of the design of large systems that is coming out of the Artificial Intelligence" experience; we say "flavor" because it's not AI, *per se*, that's important, but rather the understanding of how to control the complexity of large, integrated, interactive systems.

Complex systems are complex; they yield their secrets only to those who have mastered the tools of their craft and who have sufficient self-control and perseverance to explore and experiment with those tools in an intelligent manner. The problem we see is the delivery system--much like the New Math situation: another set of "powerful ideas," but dispensed by inadequately prepared instructors. Programming is a difficult task; even turtle graphics is difficult. To say otherwise is a mistake.

To make our point, we go back to the ties between Lisp and Logo. The spirit of interactive program development is one of the deeper "fingertips." The porting of Lisp's list-structure is in the grey area between "brains and brawn," and it also highlights one of Logo's muddled understandings of what language represents in general and what Lisp represents in particular. We could let this go except one frequently hears that Logo is "Lisp with graphics, but without parentheses." Unfortunately, Logo got mostly "bath-water" and little "baby" when it rejected parentheses.

In order to state our case it is necessary to analyze programming languages. For that's really the critical issue: "Is Logo a [good] language for learning?" Will it lead to the kinds of insights that has made Lisp so useful; will it grow "brains in their fingertips?"

To answer these questions we have to be more precise about what a programming language is. It's a tool, of course, but more than that, it is (or should be) a notation for expressing ideas. What kind of ideas? How do these ideas relate to other disciplines? How long-lived are these ideas?

Until the second half of this century mathematical notations were the major way of expressing formal ideas. With the rise of programming languages, a new, more dynamic media is developing. However, those languages still owe a strong debt to their mathematical precursors. In fact, we can characterize two classes of programming languages in terms of this heritage: the functional languages and the relational languages. There is also a third category, which we call the procedural languages, whose ancestry is more dubious. We will discuss all three classes in the next paragraphs. This classification is important in its own right, but it also highlights the issues that Logo should address if it is to retain the title as a "language for learning."

Procedural languages are represented by the traditional "high level, general purpose" languages: Pascal, Ada, BASIC, and FORTRAN ..., wherein the computing style has as its main feature, state-change, side-effect, statements (commands), and assignment. We must be careful here, for the word "procedure" is used in at least three distinct ways in the literature:

1) "procedure," as in side-effect, statement-oriented, assignment-driven languages;

2) "procedure" as in algorithm or process or operation; and

3) "procedure" as in subprogram, or component of a complex program.

The second usage is indeed a powerful, intuitive idea, one that makes it possible for us to look at a program and figure out "what it does." We'd rather call this second usage of "procedure" a "process," or "operational" interpretation. The first usage is an idea whose time has gone. "Procedural" in this sense is an imperative or command-driven view of computation and at the root of the phenomenon that John Backus called the "Von Neumann Bottleneck." The third usage is a corruption of the notion of modularity. It equates a programming technique of these procedural languages with the more general idea of encapsulating an idea or process and giving it a name. These latter two uses of "procedure" are more clearly treated in our next two language categories.

Functional languages are based on the notions of function application and value-producing computations. The functional family includes: APL, leading to the FP languages that have been developed by John Backus; and the Lisp family leading to Scheme and TLC-Logo. These are characterized by simple semantics, with complex computations being built up by combinations of operators or functionals. In a purely functional language the notation only describes relationships between components and makes no demands on how these relationships are computed.

The final element of the trio is the relational family, whose most widely recognized sub-category contains the logic programming languages. The most well-known example of such a language is PROLOG, the starting point of the Fifth Generation Language effort.

In a logic programming-language, we use relations to describe a computation in the same way that the functional languages use functions to describe situations. What is the advantage of a relational language over a functional one? done? A relational language expresses problems as a collection of logical assertions--typically assertions about individual objects and relationships between objects. Though these assertions appear to be purely descriptive, such notations are executable.

Relational languages are closely related to "constraint-based" programming, a technique that underlies (in a very weak fashion) the phenomenon of "spread-sheet" programs. Relational languages are "higher level" than their procedural or functional cousins, since relational languages state PROBLEMS while the others state SOLUTIONS. This distinction between problem statement and problem solution is a key one. It is the difference between "description" and "prescription," and is the driving force behind the move to relational languages.

Unfortunately, terminology is confused about "description" versus "prescription." For example, on p.100 of *Mindstorms*, Papert says:

> ... thus computer scientists have devoted much of their talent and energy to developing powerful descriptive formalisms ... most of [computer science] is not the science of computers, but the science of description and descriptive languages.

He then goes on to discuss the building of a stick figure in Logo in a very order-dependent, prescriptive fashion.

Thus, to summarize, all three language families have executable models. But while procedural languages only have a process interpretation, the functional and relational languages also have a mathematical (non-process) interpretation. That is, they can be looked upon as abstract descriptions of phenomena independently of how (or even if) they are executed on a machine. This abstraction means they have notational/expressive power that may be reasoned with and about. This ability is what has made mathematics the Queen of the Sciences. It is the kind of thing we should strive for in computation--not making computation as a subset of mathematics, but to develop a free-standing theory of computation.

However, the procedural languages, by their very nature, are tightly coupled to a specific kind of execution device--the von Neumann architecture. The coupling of notation to lock-step sequential, side-effect driven computation forces the programmer to over-specify the solution, and disallows the potential that relational languages have to specify the problem.

But this old view of language is particularly deadly in introductory settings. Those who come from a procedural introduction to computing have a much more difficult time "letting go" to exploit the freedom of functional or relational languages.

And thus from this perspective on languages and the future of computing, we finally get to the root of the problem with Logo: instead of retaining their functional Lisp roots, traditional Logos have emphasized the characteristics of the procedural language class. This makes it quite easy for those who know and like BASIC to make the shift to Logo: they can continue to use their old procedural perspective!

It took a long time to realize that this was what was going on. We knew that Logo was supposed to be a Lisp "dialect;" we knew that Lisp had a reputation for being difficult; so how could a "difficult" language be so easily transformed into a language that BASIC-ites could clasp to their bosom by just adding a turtle and removing parentheses? We believe that this procedural/functional flaw is at the heart of the problem.

MIT-based Logos have made procedure (side-effect) the default action module, and have made function (value-producing) a second-class citizen; to compute a value you must use OUTPUT and STOP. This pair has direct roots in the FORTRAN and BASIC commands, RETURN and STOP. So, could traditional Logo be tagged as "BASIC-with-graphics?" We think so; the underlying models are compatible.

And must a "functional Logo" be more more difficult? Our experience with TLC-Logo has not shown that to be the case. The procedure versus function distinction doesn't even appear at the turtle-graphics level. Functionality first appears when you use TLC-Logo as an arithmetic calculator; but that's an immediate carry-over from school work anyway. How many algebra books use OUTPUT and STOP?

With the return of functions to first-class syntax, the tie into traditional mathematics is immediate: functions look like functions. They can be written, composed, and understood with the same perspective that one brings to high-school algebra. Thus a frequently expressed disassociation between mathematics and programming ideas can be nipped in the bud. This decoupling is reinforced by the procedural view that drives a wedge between operational and denotational, between the intuitive and the abstract. The process-interpretation of functionality can be driven both ways: from functional Logo to mathematics, and conversely.

We would argue that functional processes are easier to "grok" than a side-effect driven collection of procedural code. And most important for an integrated education, the jump from process/executable notation to mathematical/abstract notation can be made from functional languages, but not from procedural ones.

Functionality seems to go against the grain of the traditional Logo camp. The real world--so their argument goes--is best understood in a side-effect situation: we command; we do things for effect, rather than value. For example, a typical argument against functionality states: "When I say 'go to the store and get some milk and bread' that is a command; and there's no real 'value' produced. Or when you command a Logo turtle to move 10 steps, there's no meaningful value produced." The actions, they say, are executed for their (side) effect on the world. The answer is that both of these commands are, in fact, part of a solution to a problem; the problems are (1) I am hungry, and (2) I'd like to create a particular graphical effect. These two problems are descriptive.

But clearly, if a descriptive notation is to be at all useful in computation, it MUST be executable. Furthermore, it's quite reasonable to expect that non-procedural notations be implemented as procedures. We've no quarrel with the idea that SOLUTIONS have side-effects, but the problem statements are relations that we'd like to have satisfied, and they're the thing that's important to convey as the basis of an intelligent language. As descriptions, functions are closer to the ideal than procedures, and relations are closer yet. The key point is that the choice of implementation (solution) is not forced into the problem specification (description). The descriptive level of notation is the long-lived component; the particular notation, or hardware, or software is so transitory that it isn't worth wasting time on.

Compare the following remark taken from Dr. Marvin Minsky's Turing Lecture 1969 "Form and Content in Computer Science:"

> Until all this preoccupation with form is replaced by attention to the substantial issues in computation, a young student might be well advised to avoid much of the Computer Science curricula ...

From what we've seen, much of what is being offered in the educational computing arena makes Minsky's indictment valid for the pre-university years.

This indictment extends over much of what's being done under the aegis of Logo. Weak books, poorly prepared teachers and cries of the standardization (QWERTY-izing) of Logo around outdated and weak models of computation--these are fatal. Unless a deeper understanding of computational issues can be developed in the community that teaches Logo, then it will go the way of the New Math.

Our challenge is to understand how to engender that exploratory experiential excitement in our educational system, mixing it with the more traditional curricula of the arts and sciences so that "brains and fingertips meet."

If not, 1984 will end "not with a bang, but a whimper."

# ITERATION IN LOGO

Brian Harvey
Atari Sunnyvale Research

*Iteration* is the process of doing something repeatedly in a computer program. Different programming languages provide more or less flexibility in their iteration facilities. In BASIC, iteration is generally done with the **FOR-NEXT** loop. Pascal provides **FOR**, **WHILE**, and **REPEAT-UNTIL**.

In Logo, the most elementary form of iteration is provided by the **REPEAT** command. This command allows a list of Logo instructions to be carried out several times:

```
REPEAT 4 [FORWARD 100 RIGHT 90]
```

For more complicated requirements, Logo programmers generally write recursive procedures like this:

```
TO POLYSPI :SIDE :ANGLE :NUMBER
IF :NUMBER=0 [STOP]
FORWARD :SIDE
RIGHT :ANGLE
POLYSPI :SIDE+1 :ANGLE :NUMBER-1
END
```

The purpose of this paper is to explore how programmers can use Logo's extensibility to create more powerful iterative forms.

Sometimes, **REPEAT** would be perfectly adequate if only it kept count of the number of repetitions. For example, we could count down to a rocket launch like this:

```
MAKE "REPCOUNT 0
REPEAT 10 [MAKE "REPCOUNT :REPCOUNT+1
  PRINT 11-:REPCOUNT]
```

But that's kind of ugly. It would be better if Logo provided an operation **REPCOUNT** which output the number of times through the current **REPEAT**. This isn't a Logo primitive, but we can make it available by defining a new version of **REPEAT**:

```
TO REP :NUMBER :COMMAND
LOCAL "REPCOUNT
MAKE "REPCOUNT 0
REPEAT :NUMBER SENTENCE [MAKE "REPCOUNT :REPCOUNT+1]
  :COMMAND
END

TO REPCOUNT
OUTPUT :REPCOUNT
END
REP 10 [PRINT 11-REPCOUNT]
```

Defining REPCOUNT as a local variable in REP makes it possible for invocations of REP to be nested.

REP and REPCOUNT allow you to perform instructions iteratively, but modifying the effect of the instructions each time based on a numeric value. A more general form of this numerically-controlled iteration would allow the programmer to specify starting and ending values for a variable, and perhaps an increment value:

```
TO FOR :VALUES :COMMAND
LOCAL FIRST :VALUES
FORLOOP (FIRST :VALUES) (FIRST BF :VALUES)
  (FIRST BF BF :VALUES) (STEP :VALUES) :COMMAND
END

TO STEP :VALUES
IF (COUNT :VALUES)=4 [OUTPUT LAST :VALUES]
IF (FIRST BF :VALUES) > (FIRST BF BF :VALUES)
  [OUTPUT-1] [OUTPUT 1]
END

FOR [NUM 10 1] [PRINT :NUM]
   10
    9
    8

    ****

    2
    1
    FOR [ODD 1 9 2] [PRINT :ODD]
        1
        3
        5
        7
        9
```

The first input to FOR is a list containing the name of the loop variable, its initial and final values, and optionally the increment value. As with REPEAT, the second input is a list of commands.

This iterative tool displays the power of Logo's RUN command to permit extensions to the language which retain the style of the primitive commands. Since the length of a list need not be declared in advance, it's easy to allow the increment value to be optional. By making the loop variable local to FOR, we preserve the modularity of programs using this tool; if the commands carried out by FOR invoke a procedure which itself uses for, there is no conflict even if the same variable name is used in the subprocedure.

In languages like BASIC and Pascal, in which the data aggregate is the array, numerically-controlled loops are usually used not to print the numbers as we've been doing, but to use them as array indices. The real purpose of the iteration is to do something with every element of an array; the index variables are of no intrinsic interest. In Logo, we can define a *mapping* procedure, which applies a *command template* to each element of a list:

```
    TO MAP :TEMPLATE :LIST
    IF EMPTYP :LIST [STOP]
    RUN LPUT QUOTED FIRST :LIST :TEMPLATE
    MAP :TEMPLATE BF :LIST
    END

    TO QUOTED :THING
    IF LISTP :THING [OUTPUT :THING]
    OUTPUT WORD "" :THING
    END
```

```
MAP [PRINT] [VANILLA CHOCOLATE STRAWBERRY]
   VANILLA
   CHOCOLATE
   STRAWBERRY
MAP [PRINT FIRST] [[ULTRA CHOCOLATE]
  [CINNAMON CHOCOLATE RAISIN] [BITTERSWEET ORANGE]]
   ULTRA
   CINNAMON
   BITTERSWEET
```

This form of iteration applies a command to each member of a list, no matter how big the list is, without requiring the use of an auxiliary variable to count the members of the list.

Since Logo allows operations (functions which output values) as well as commands, another powerful form of iteration over a list is one which applies an *expression template* to each member of the list, combines the results into a new list the same size as the input list, and outputs that new list. Programmers accustomed to BASIC may not even recognize this as iteration, but programmers accustomed to APL will feel right at home with it:

```
TO MAP.LIST :TEMPLATE :LIST
IF EMPTYP :LIST [OUTPUT []]
OUTPUT FPUT (RUN LPUT QUOTED FIRST :LIST :TEMPLATE)
  (MAP.LIST :TEMPLATE BF :LIST)
END

PRINT MAP.LIST [FIRST] [[ULTRA CHOCOLATE]
  [CINNAMON CHOCOLATE RAISIN] [BITTERSWEET ORANGE]]
   ULTRA CINNAMON BITTERSWEET
PRINT MAP.LIST [SQRT] [1 2 3 4]
   1 1.414 1.732 2
```

Many projects in cryptography, for example, require some transformation of each word of a sentence:

```
TO PIGLATIN :WORD
IF MEMBERP FIRST :WORD [A E I O U Y]
  [OUTPUT WORD :WORD "AY]
OUTPUT PIGLATIN WORD BF :WORD FIRST :WORD
END

PRINT MAP.LIST [PIGLATIN] [THE RAIN IN SPAIN STAYS
  MAINLY ON THE PLAIN]
   ETHAY AINRAY INAY AINSPAY AYSSTAY AINLYMAY ONAY
   ETHAY AINPLAY
```

A similar procedure can map an expression template over each letter of a word:

```
TO MAP.WORD :TEMPLATE :WORD
IF EMPTYP :WORD [OUTPUT "]
OUTPUT WORD (RUN LPUT QUOTED FIRST :WORD :TEMPLATE)
  (MAP.WORD :TEMPLATE BF :WORD)
END

TO LOWERCASE :WORD
OUTPUT MAP.WORD [CHAR 32+ASCII] :WORD
END

PRINT LOWERCASE "HELLO
   hello
PRINT MAP.LIST [LOWERCASE] [HELLO THERE]
   hello there
```

It would be easy, and slightly more in tune with the usual Logo style, to combine MAP.LIST and MAP.WORD into one operation which invokes one or the other, depending on the nature of its second input.

The mapping operations we've seen so far have preserved the "shape" of their inputs. That is, the output is a list (or word) of the same size as the input. Another kind of iteration on a list is to combine the elements of the list using a dyadic operation like SUM or PRODUCT. This is called *reduction* or *accumulation* of the list:

```
TO REDUCE :TEMPLATE :LIST
IF EMPTYP BF :LIST [OUTPUT FIRST :LIST]
IF EMPTYP BF BF :LIST
  [OUTPUT RUN SE :TEMPLATE LIST (QUOTED FIRST :LIST)
   (QUOTED LAST :LIST)]
OUTPUT RUN SE :TEMPLATE LIST (QUOTED FIRST :LIST)
  (QUOTED REDUCE :TEMPLATE BF :LIST)
END

PRINT REDUCE [SUM] [2 3 4]
    9
PRINT REDUCE [PRODUCT] [2 3 4]
   24
PRINT REDUCE [WORD] [ONE LONG WORD]
   ONELONGWORD
```

So far we have treated iteration on lists as if, like arrays, lists were necessarily *flat* (containing only words or numbers as members). We haven't taken advantage of the complex structures possible within a list.

For example, a list can be used to represent a *tree*, a data structure in which each branch can lead to further branches. Here is a procedure which is like **MAP.LIST**, in that it preserves the shape of its input, but it iterates over the *leaves* of the tree rather than the top-level branches:

```
TO MAP.TREE :TEMPLATE :TREE
IF WORDP :TREE [OUTPUT RUN LPUT QUOTED :TREE
  :TEMPLATE]
OUTPUT MAP.LIST LIST "MAP.TREE :TEMPLATE :TREE
END

PRINT MAP.TREE [FIRST]
  [[THIS IS] [A [VERY [STRANGE] WAY] TO GROUP] THE
  [[WORDS]] [IN [A] SENTENCE]]
   [T I] [A [V [S] W] T G] T [[W]] [I [A] S]
```

(Try **MAP.LIST** instead of **MAP.TREE** if you don't see that there is a difference in the results.) **MAP.TREE** is recursive in an unusual way; it doesn't invoke itself directly, but it uses its own name as part of the template input to **MAP.LIST**.

**MAP.TREE** maintains the shape of the tree which is its input because it is "made out of" **MAP.LIST**, a shape-preserving operation. Suppose we want to *flatten* a tree, i.e., to output a list of the words in the tree, but with the structure of the tree eliminated? We can build such a tool by using **REDUCE**, a procedure which itself performs a particular kind of flattening:

```
TO FLATTEN :LIST
IF WORDP :LIST [OUTPUT :LIST]
OUTPUT REDUCE [SE] MAP.LIST [FLATTEN] :LIST
END

PRINT FLATTEN [[THISIS] [A [VERY [STRANGE] WAY]
  TO GROUP] THE [[WORDS]] [IN [A] SENTENCE]]
    THIS IS A VERY STRANGE WAY TO GROUP THE WORDS IN A
    SENTENCE
```

(By the way, **FLATTEN** relies on the one-level flattening effect of Logo's **SENTENCE** primitive as well as the flattening effect of **REDUCE**.)

This selection of tools certainly doesn't exhaust the possible forms of iteration in Logo. It does, perhaps, give some idea of the range of possibilities. It may help overcome the idea that iteration must be numerically controlled; instead, this project brings to light the intimate relationship between the data structures of a language, and the kinds of iteration which are appropriate to it. Lists are a more flexible form of data aggregate than arrays, and they give rise to more flexible forms of iteration.

In passing, these procedures show some of the power of Logo's **RUN** primitive in extending the language.

# ADVANCED LOGO PROGRAMMING?

Peter Ross
University of Edinburgh

PETER ROSS IS UNABLE TO PARTICIPATE IN THE PANEL
"ADVANCED PROGRAMMING" DUE TO A SCHEDULING CONFLICT

I had some trouble producing this 'position paper' on "advanced Logo programming." After all, what is advanced programming, in Logo or any other language? To judge by some of the articles now turning up in the press, it is anything beyond pure turtle graphics. Others suggest that it can be recognized by the presence of serious (=? incomprehensible) list processing, or the presence of more than one hierarchic level of procedurisation. All these, however, are behavior measures: there is a depressing trend toward towards the rediscovering of behaviorism apparent in the expanding literature that describes the joys of Logo. For instance, how often have you seen the comment:

"she tried REPEAT 4 [ FORWARD 90 RIGHT 100] , obviously she can't yet distinguish between lengths and angles"?

One assumption in such comments is that a person cannot be said to understand something unless she can be articulate about it (and there are other assumptions, such as that it is an example of failure). Worse, she can be said not to understand if she fails to be articulate. If you think this is reasonable, try explaining how to ride a bike.

Obviously, behavior measures are very tricky to use in a credible way, and even the need for them is often suspect. I shall avoid the question, and say that there is a lot to gained by trying to be ambitious in Logo programming. For one thing, there is a lot of experience and many techniques that can be copied from the folklore and literature of Lisp; the many variants of Eliza and Animal are little more than straight transliterations of small but revered Lisp demonstrations. Such programs are an interesting form of communication, as much between people as between a person and a machine. Only one level of the communication is explicit; others are hidden. If you read other people's programs, or some of your own that are beginning to fade from memory, ask your self questions such as:

* "why is it done this way, not some other way?"

* "why are the steps in this order?"

* "why is there a procedure to do that?"

* "what is the aim of the program?"

* "is this the way I'd do it, now? interesting or dull?"

Compare the business of programming in a procedural language with the business of preparing a paper, or a talk. The main ingredients are similar:

* choosing the overall aim

* deciding what needs to be included and what should be left out

* deciding what parts need amplification

* constructing the order, so that it all makes good sense

* doing it

* a post-mortem to do something about the poor bits.

(Digression: you can push this analogy quite a lot. Think of **'GO "LABEL3'** and **'(cont.p. 94)'**). The high-level plan probably contributes more to the overall result than the low-level moment-to-moment utterances. However, the high-level skills are hard to acquire in isolation, and experience is undoubtedly the best way to learn. I feel that Logo is an excellent medium for this, not just for the standard reasons of having a straightforward syntax, few eccentricities and list processing. The marriage of the disparate worlds of turtle graphics, list processing and procedural tools is a very fruitful one, as the Lisp community can testify. Turtle graphics is a visual and spatial world (tried working in 3-D yet?), list processing is a world of structure and representations, procedures are about planning and flow of events. Very few interesting problems give way easily before a pure, single-world approach. Indeed, the interest very often stems from the hybrid, multi-world nature of the difficulties.

However, although a good implementation of of Logo is a very good tool for 'advanced' exploration and experiment, there is an educational bottleneck. More material needs to be widely published if Logo is going to be seen as a serious language in its own right. Here are a few examples of useful material:

* general and situation dependent matchers are easy to create and make simple language-processing and "reasoning" programs possible

* tools for manipulating network-like formalisms (however you choose to represent a network) are easy to create and get you started on simple compilers and simple parsers

* tools for manipulating lists structured in particular ways can be easy to make and will get you involved in interesting knowledge-representation questions (once you've decided what 'knowledge' is!).

It is always pretty straightforward to create a Logo interpreter written in Logo. Having that, you can go on to experiment with changes in the language--a microworld of Logo rather than a microworld in Logo. Some ideas:

* wouldn't it be nice to be able to say "draw that again, over there" or "draw that again, reflected in this line" or "erase that and do it again, a little bigger/smaller" where you don't actually have to qualify what you mean?

* wouldn't it (or would it) be nice to do without the colon? is it needed?

* wouldn't it be nice if Logo did what you meant, not what you said?

* wouldn't it be useful if Logo didn't mind when you failed to specify some parts of your program, but just asked you to fill in as much of the gap as was needed to carry on?

Each of these ideas is practicable, to some extent at least, if you have an adequate Logo such as one of the main ones that runs on an Apple II. You may feel that they definitely count as 'advanced,' yet the difficult part of each is clarifying the objective. The programming is not very sophisticated (see below), but--as with other languages--there is a distinctive flavor to the effective styles that people develop. Components of this flavor include such maxims as "build tools rather than products," "it's worth trying to generalize," "keep the chunks small, don't let them get out of hand." Unfortunately, these can only be taken on faith to start with--it's a case of "come on in, the water's fine." Inquisitiveness is a great asset here. For instance, are you one of the many people who have never worried about what the procedure COPYDEF (in LCSI Logos) is for? A short example:

```
COPYDEF "OLD.FD "FD
```

means that OLD.FD is a synonym for FD, so

```
TO FD :DISTANCE
  OLD.FD :DISTANCE * :SCALE
END
```

redefines FD, if REDEFP is TRUE. In future, your procedures that use FD will use the new version. Thus you can change the size of drawings just by changing the global variable SCALE (remember about BK and so forth, of course). The procedure has to use OLD.FD rather than FD, otherwise it would just recurse rather than draw. COPYDEF makes old definitions accessible.

This example is technically trivial, yet shows a very powerful idea: you can modify what existing programs do without editing them.

Not much is yet written about using Logo as an experimental implementation language--but some might argue that this is desirable because learning new techniques within a particular context tends to bias subsequent creative efforts to be within that context too. However, this gives you an opportunity to influence the future. Few other languages give users such a chance to play about with new features and ideas. and few are less constrained by standards committees and commercial demands. New features I might find a use for include user-definable infix operators, real-time aids, text-processing aids, windowing, better communication with external devices, and so on. It's easy to dream up ideas: the hard part is justifying them on the scales of generality, learnability and usefulness. I believe advanced Logo programming is more to do with the trade-off between conceptual cleanliness and usability than to do with technical skill. It's an art, and a great pleasure, but it's not a spectator sport.

# LOGO: PAST AND FUTURE

Cynthia Solomon
Atari Cambridge Research

## A vision

In the mid sixties a group of us began to work with Seymour Papert to develop ways of using computers to enhance children's learning. The computer, representing for us one of the most powerful and flexible tools for thinking, would help us create a mathland, where children would learn mathematics by doing mathematics and reflecting on what they do. In mathland, children would build on what they already know, and learn new things in the process of actual research projects. Mathland became a metaphor for all learning activities.

The computer would provide a rich array of interesting intellectual worlds, diverse enough and attractive enough so that a wide population of individuals would become involved and make meaningful and enriching extensions to these worlds. Links between the child's real world and a mathland or intellectual play world would be made through transitional objects such as computer controllable turtles and through transitional activities such as juggling and other circus arts activities.

Logo became a focal point for ideas which would lead to redesigning school as we knew it. We were planning to change the content and curriculum of elementary schools and in that way change the social as well as intellectual atmosphere influencing children. The children and adults together would shape and develop the curriculum. Computer sciences would provide the foundations for the new school activities.

The revolution, as we envisioned it, has not taken place. The job is larger than we thought and not enough people have been involved in making new content; the vision of the past is still one for the future. In that vein we have been engaged in new research at Atari.

What follows is a brief description of some of our research activities. Many people contributed to this research; I have acknowledged only a few of them.[1]

[1]Special thanks to Alan Kay.

## Atari Cambridge Research - Research Projects

At Atari Cambridge Research we had been building computing environments for the next generation of computers; we were looking toward the playstation of the future. In developing these environments, we had in mind giving powerful tools to children as well as adults. We saw these environments as providing a wide range of engaging recreational activities, allowing people to become musicians, painters, cartoonists, game designers, game players or even choreographers. We imagined different computer worlds which people will enrich in unique ways depending upon their personal interests and knowledge. In a sense, our research consisted of making machines smarter so that people could use them to make themselves smarter.

The computing worlds we were shaping could influence hardware features for the next generation of computers for people in homes, schools and offices with special hardware for communications, sound and animation. We anticipated computing environments with good graphics for moving pictures and for reading text which would be more powerful and cheaper than the IBM-PC or the Apple Lisa. (For our own research very large, powerful machines have been essential so that code could be easily written and debugged, and so that a wide range of experimental gadgets could be attached and controlled easily.)

As we explored the complexities involved in making machines smarter and easier to control, we drew upon research in many areas of computer science, artificial intelligence and learning and took advantage of our close ties with the MIT Artificial Intelligence Lab and the Laboratory for Computer Science to enhance our research program.

Our research fell into 6 major categories:

* Object-oriented Logo
* Music
* Gesture systems
* Robotics and Interactive Environments
* Animation
* Teaching and Learning

## Object-oriented Logo

This research has involved a team of people. Using Gary Drescher's design Ed Hardebeck has been coordinating a programming team including Stephen Hain, Jay Jones, Bill St. Clair, and Scott Layson. Mark Gross, Michael Grandfield, Ken Haase, students of Mark's and others have been developing new computing environments in this new programming language.

> We have been extending Logo both as a language and as a computing environment. We were exploring whether this new "object-oriented Logo" (called qLogo) could be the unifying element for our playstation of the future. qLogo retains the features of the present Logo within the language of dynamic objects so that you can build your own environments from window systems and dungeon kits to special purpose visicalc-like constraint systems.
>
> In qLogo the user can create different kinds of computational objects and write programs to control these objects in interesting ways. An example of a "computational object" is the Logo turtle. In a qLogo environment you can create any number of them. It is possible to create not only turtles, but other objects as well. Each can have attributes and meaning built up using descriptive procedures.

## Music

This research was done in conjunction with work under the direction of Marvin Minsky at the MIT Artificial Intelligence Lab. David Levitt a graduate student of Minsky's worked with ACR researchers Jim Davis and Tom Trobaugh.

> We see that most musical activity today is mainly passive. Like spectator sports, people mostly listen. We were asking whether we could change it to be more active. We believe that within a few years we could offer devices, hard and soft, to let each child be Conductor, Arranger, Composer, or whatever -- without requiring the years that now must be invested in music training. Using currently available synthesizers and other musical computer aids we were building systems which would allow amateurs to make, describe, compose, express, or perform music for themselves or their friends.
>
> The computer will serve as an aid in musical composition both

by composing music itself, and by serving as an instrumental accompanist, taking its cues from the users and modifying and embellishing their scores.

The computer will be able to play music either from scores or by improvising; and it will be able to do so in a variety of styles. The computer will serve as a composer's aid so that people can write pieces for the machine to play or teach the machine.new ways to play other compositions.

## Gesture Systems

Research on gesture has principally involved Margaret Minsky and Ed Hardebeck.

We have been exploring different ways of communicating with computers. One of our major projects involves the use of "gesture". The project raises several issues. How can gestures be input to the computer and translated? What sorts of gestures can the computer be made to understand? The answers to these problems require both hardware and software solutions. Research is needed to determine the best kind of language to support this form of dialog between computer and user. Our first explorations into the use of gesture and a "gesture language" have taken a standard touch sensitive screen and embellished it to be force sensitive. We have used this technology to develop a "software" button box which let's the user drive the turtle around the screen and write procedures for it.

## Robotics and Interactive Environments

The team of researchers working on these ideas included Margaret Minsky, Max Behensky and Doug Milliken.

We were developing interactive environments that permit the computer to "gesture" to the user. We have developed a forcefeedback joystick that can move a person's hand in any 2D direction. The joystick can be programmed for different game settings. For example, guess the object, are you tracing out a square, a circle, or a triangle? Are you in thick stew or thin soup? Can you get out of the maze? In one sample game that we developed, the joystick acts like a fishing pole and lets the player try to land the salmon.

We made a force feedback steering wheel (imagine learning to drive under program control).

Recently Mark Gross, Doug Milliken and Peter Cann have been experimenting with a computer-controlled marionette world.

## Animation

Animation research has been conducted by Ken Haase, Michael Grandfield, Mark Gross, Ed Hardebeck, Jim Davis and Bill St. Clair.

> This research focused on how to control multiple objects and parallel processes for use in real-time animation. These are issues for qLogo and for users of qLogo, as well as our music and animation research projects. 3D turtles and the beginnings of a choreographer's assistant have been built in qLogo.

## Teaching and Learning

All of the work in the lab is meant to provide people with computational power for their own development. This requires that we work with different populations -- in and out of school, in their homes and in other play and learning areas. Although we are particularly interested in young children, because of the potential for giving them a really powerful tool in their early learning, we are also interested in people of all ages.

In this research Susan Cotten, Annette Dula, and Lauren Young have been teaching children using Atari Logo. They have been working in different settings: in a community center environment, in a school setting and also in homes. We want very much to look at what happens in homes. What kinds of learning take place? What kinds of projects emerge? What kinds of help and materials are needed? Do kids teach their parents? How do parents interact with their kids around their home computers? What happens in homes of poor black or hispanic kids? What happens to kids in school who have computers in their home? What kinds of games do they play? What kinds of games would they like to build? What kind of socializing takes place? What do the kids talk about? Do they relate what they do at home with what they do at school?

Part of our research is to look for new and interesting computational worlds and linking them to people's everyday lives. We looked to qLogo to provide us with the tools to explore this goal. We had expected to spend the next two years debugging qLogo, designing microworlds in it and writing materials describing its use. A part of this process would be a further examination of the kinds of questions raised by Logo. Such issues transfer to qLogo.

# EXTENDING LOGO

Gerard Dahan
ACT Informatique - Paris

C'est etendre le pouvoir des individus et communautes a apprendre et a maitriser les connaissances liees a la revolution contemporaine de l'information et au chaos general des connaissances, des cultures et parfois des societes.

L'ordinateur offre, grace a Logo, la realisation d'un vieux reve, celui de la possibilite de l'autoapprentissage, de l'autotest de ses hypotheses, il a ouvert dans le domaine de l'apprentissage, le droit a l'erreur, a la construction de ses idees.

Etendre Logo aujourd'hui c'est etendre le support de l'usage social et culturel des utilisateurs des technologies.

* Support culturel par les reflextions sur les nombruex vecteurs de l'information, a tous ages et a debit de plus en plus grand

* Support logiciel par l'extension du champ des representations des connaissances et de manipulation des concepts

Dans ce cadre on s'interrogera sur les Logos a venir, les Logos autorisanty la manipulation d'iedees dans des micromondes aussi constructibles que celui de la tortue.

Et particulierement, on presentera VISILO. Essai de difinition d'extension de Logo dans le domaine de l'image, utilisant un Logo pour piloter, interroger, programmer la connection avec un videodisque et plus generalement l'image interactive.

Les representations de connaissance, les concepts de manipulation d'objets prennent dans ce cadre un sens qu'il s'agit de preciser. Etendre Logo, mettre en oeuvre le concept de 'micromonde' pour permettre aux enfants, aux individus de maitriser le 'real world'.

# TEN THINGS TO DO WITH A BETTER COMPUTER

W. Daniel Hillis
Thinking Machines Corporation

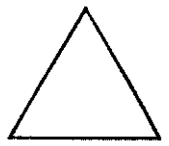
The following are a series of ideas on what could be done with a very "smart" computer, a TV screen, and a kid. It assumes that the computer is capable of understanding a very Englishish language in which it is possible to tell the computer to create creatures. Each of these creatures has a very precise set of actions, friends and relationships to other creatures. The user may both create creatures to his specifications and use stock creatures that are already known to the computer.

All of the creatures in the system are simultaneously doing their thing. (Actually a real world computer can only do one thing at once, but this can be invisible to the user.) The creatures can communicate by sending messages to each other.

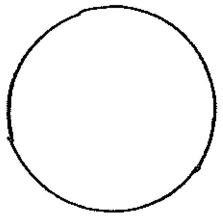
Unfortunately, such a language does not exist, but here are a few ideas of what it might look like, and what kinds of things could be done with it.

## 1. Build a Truck

In the Logo language, one kind of thing that is given to you is a TURTLE. A TURTLE looks like this:



If you tell the computer FORWARD 10, the computer tells the TURTLE to move forward ten steps. When you type FORWARD 10 into the computer what you really mean is "tell turtle FORWARD 10," but since the turtle is the only thing around which would make sense out of such a message, there is no reason to be specific. Why can't other things understand that message? A CIRCLE is a creature, just like a turtle. The main difference is that a CIRCLE looks like:



Just like a TURTLE, a CIRCLE can understand messages like FORWARD 10 and LEFT 90. A CIRCLE can also understand some messages that a turtle can't, like GROW and SHRINK. You can tell the computer:

```
    JOHN IS A CIRCLE.
```

(This causes the computer to make a creature called JOHN, who has all the properties of a CIRCLE. JOHN can now be seen on the screen).

```
    TELL JOHN "GROW 5".
```

(JOHN gets bigger)

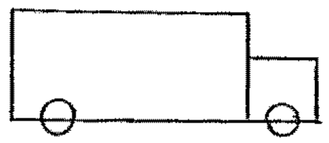
Other sorts of "primitive" creatures that might be given to a child are BOX, LINE and TRIANGLE. This is how to draw a truck:

```
    BACK IS A BOX.  TELL BACK "WIDTH 100 '
      HEIGHT 50".

    FRONT IS A BOX. TELL FRONT "FORWARD 50 RIGHT 90
      FORWARD 10".

    FWHEEL IS A CIRCLE.  TELL FWHEEL "RIGHT 120
      FORWARD 60".

    BWHEEL IS A CIRCLE.  TELL BWHEEL "LEFT 120
      FORWARD 50".
```



Now that the computer has this picture on the screen we can tell it that this is what a truck looks like by saying something like SNAP IS A TRUCK. The computer now knows about a creature called TRUCK. Like CIRCLE, TRUCK understands messages like FORWARD and GROW. Any creature that was made from a SNAP understands such messages.

So far all of the creatures that we have talked about can be seen on the screen. This is not always the case. Most creatures are invisible. For example, we could give the truck an engine:

```
ENGINE DOES NOT REPEAT, TELL TRUCK "FORWARD 1".
```

This sentence will define the behavior of ENGINE. It is not clear to me that it should also cause an ENGINE to be created. Perhaps it is necessary to say MAKE ENGINE before TRUCK will actually start to move.

## 2. Design an Alarm Clock

We are going to put on a play. The plot is very simple:

The face of a clock is on the screen.
The hands are moving (much faster than a real clock).
When the clock reaches 9:30 the bell
on the console rings.

The first thing to do when putting on a play is to decide who the actors are. In this case the actors are the big hand, the little hand, and the alarm. They each have an action and a cue. For example, the action of the alarm is to ring the console bell. Its cue comes when the big hand is pointing down and the little hand is pointing to the right.

There is also a little bit of scenery in the play: the face of the clock. If we like we can just think of this as an actor who doesn't do anything.

Now let's use the computer to direct this play.

```
   FACE IS A CIRCLE.  TELL FACE "GROW 100".
```

(We will start out with a simple face. Later, if desires, we can add numbers marking other frills.)

```
   BIGHANO HAS A LINE>
   TELL LINE "GROW 80". REPEAT, WAIT 1 TELL LINE
     "RIGHT 30".
```

(HAS A means that BIGHAND has a friend that is a creature of type LINE. Computer people may like to think of this as a local variable. There may be more than one line around, and they may not have names of their own. The only thing that distinguished this LINE from some other LINE is that it is a friend of BIGHAND.)

```
   LITTLEHAND HAS A LINE.
   TELL LINE "GROW 50".
   IF BIGHAND'S LINE'S ANGLE IS 0, TELL LINE "RIGHT 30".
```

(Now that there is more than one LINE around we must know which one we are talking to. If LITTLEHAND talks to a LINE it is usually talking to its own friend. But sometimes it may want to talk to someone else's LINE. This can be done with an apostrophe S. In the same way that BIGHAND has a LINE associated with it, each LINE has an ANGLE. This ANGLE is a friend of LINE that keeps track of where he is pointing.)

```
   ALARM DOES.  IF BOTH BIGHAND'S LINE'S ANGLE IS 180
     AND LITTLEHAND'S LINE'S ANGLE IS 270,
   TELL CONSOLE "RING".
```

(Do we really treat the console as just another creature? I think so, but it does sound a bit fanatic.)

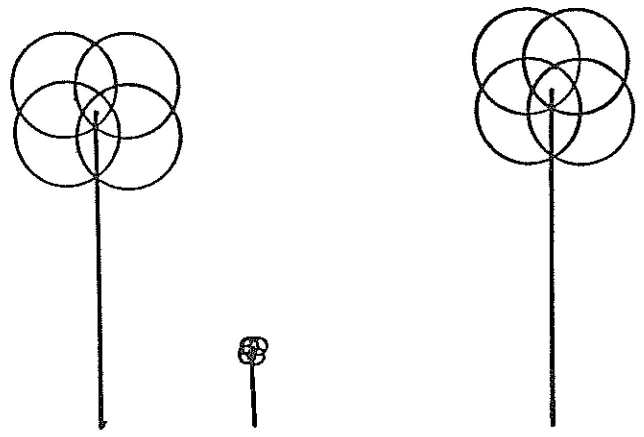
I think that all displayed creatures should really all rotate about their centers. If this is the case then "RIGHT 30" in the above example should be replaced by "BACK 40 RIGHT 30 FORWARD 40".

## 3. Grow a Garden

There can be more than one creature in the world at any given time. Also creatures can cause other creatures to be created and destroyed (made and unmade). Here is an example:

```
MAKE CIRCLE. TELL CIRCLE "RIGHT 45 FORWARD 10".
MAKE CIRCLE. TELL CIRCLE "RIGHT 135 FORWARD 10".
MAKE CIRCLE. TELL CIRCLE "LEFT 45 FORWARD 10".
MAKE CIRCLE. TELL CIRCLE "LEFT 135 FORWARD 10".
MAKE LINE. TELL LINE "GROW 20 BACK 10".
SNAP IS A FLOWER.
SEED DOES;
WAIT 10.
MAKE FLOWER. TELL FLOWER "RIGHT 90 FORWARD 100 *
  RANDOM LEFT 90".
REPEAT 15, WAIT 2 TELL FLOWER "GROW 1".
MAKE SEED.
WAIT 100 UNMAKE FLOWER.
```

Notice what a **SEED** does. It creates a **FLOWER**, moves it to a random position, grows it, makes another seed (Recursion!), and eventually destroys the flower. There are never more than ten flowers on the screen. Do you see why?



## 4. Build a Hole

One of the nice features of the Logo language is the fact that the screen does not need to be thought of as a cartesian plane. The way a turtle moves is specified relative only to where the turtle is, not in terms of any absolute system. This "Turtle Relativity" is one of the most powerful ideas in Logo.

Unfortunately, as Paul Goldenberg has pointed out, there is no easy way in Logo to find the relative position of some other point on the screen. For example, if a turtle wants to find out how far it is away from something it cannot do so unless it knows its coordinates on the absolute cartesian plane, the point's coordinates, and the Theorem. This seems like a crutch that is not in keeping with "Turtle Relativity."

To see how relative positions are a useful thing to know, especially in a world full of creatures, let's build a hole. A hole is a creature that sits on the screen. If any other creature comes too close, it falls in and disappears.

```
HOLE IS A CIRCLE.  FOREACH CREATURE,
IF DISTANCE FROM CREATURE TO HOLE IS LESS THAN 100
AND CREATURE IS NOT A HOLE, UNMAKE CREATURE.
```

At this point, we should stop and say a few words about relationships. A relationship is a comparison between two creatures. "IS A" and "IS LESSTHAN" are examples of relationships. It is possible to declare a relationship between two objects or to ask if a given relationship exists. There can also be rules that apply to relationships. For example, if A IS LESSTHAN B then B IS GREATERTHAN A, or if A IS A B then all of the properties of B are also properties of A. "ISNOT" in place of "IS" just asserts the absence of the relationship. More about this later.

More thought needs to be given to the relationship between a creature and its image on the screen. So far these have been treated as identical. This may not be the best way to think about it. One can certainly imagine an "IS PICTUREOF" relationship, but this might just serve to complicate rather than clarify.

## 5. Teach the Computer About Families

To understand how the computer deals with relationships, let's teach the computer about a very complex system of relationships: the family. We need not to restrict ourselves to the set of "primitive" relationships that the computer already knows. We can create our own. To do this we must define the rules that govern them. (I am not really happy with this format of stating the rules, but it will do for now. The word "OF" is used to make things easier to read. It has no syntactic value.)

```
IF A IS A CHILD OF B, THEN B IS A PARENT OF A.
IF A IS A PARENT OF B, THEN B IS A CHILD OF A.
IF A IS A CHILD OF B AND A IS A GIRL,
  THEN A IS DAUGHTER OF B.
IF A IS CHILD OF B AND A IS A BOY,
  THEN A IS SON OF B.
IF A IS PARENT OF B AND A IS A GIRL,
  THEN A IS MOTHER OF B.
IF A IS PARENT OF B AND A IS A BOY,
  THEN A IS FATHER OF B.
IF A IS PARENT OF B AND A IS PARENT OF C,
  THEN B IS SIBLING OF C.
IF A IS SIBLING OF B AND A IS A GIRL,
  THEN A IS SISTER OF B.
IF A IS SIBLING OF B AND A IS A BOY,
  THEN A IS BROTHER OF B.
IF A IS PARENT OF B AND A IS SISTER OF C,
  THEN C IS AUNT OF B.
IF A IS PARENT OF B AND A IS BROTHER OF C,
  THEN C IS UNCLE OF B.
JOHN IS BROTHER OF JIM.
MIKE IF FATHER OF JIM.
JILL IS DAUGHTER OF MIKE.
JILL IS MOTHER OF BOB.

JOHN IS UNCLE OF BOB?
   YES
```

(This is the computer's reply to the question.)

```
JILL IS A GIRL?
   NO
```

(The computer has not been told enough to deduce that this relationship exists, so it assumes that it does not.)

This may have been an unfair example, but it does seem possible that this type of a language can handle data bases in a powerful sort of way.

## 6. Crunch a Number

I sort of hate to give this example because it points out something that I don't really have any good ideas on how to do, specifically, how to explain to creatures how they should handle messages. As an example of this, let's create a recursive factorial creature. If you tell it a number it replies with the factorial of the number. Here is one way it might look:

```
   FACTORIAL HAS HELPER.
   HELPER IS A FACTORIAL.
   IF MESSAGE IS 1, ANSWER 1
   ELSE TELL HELPER MESSAGE - 1 ANSWER MESSAGE * REPLY
```

I don't really like this, but I like even less something like:

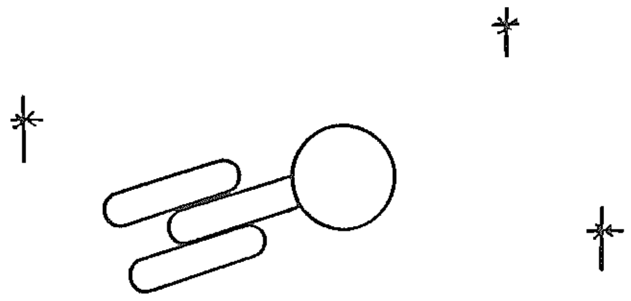
```
   (TO FACTORIAL "A: A = 1 ? (RETURN 1)
     RETURN A * FACTORIAL A - 1))
```

Maybe a child should be taught the top way first and then taught to abbreviate it to:

```
   FACT HAS H. H IS A FACT. IF : = 1, AN 1,
     ELSE TL H :-1, AN : * ?
```

This problem needs some thought.

## 7. Teach the Computer Spacewar



Here is a good project for a high school student. What we want is a game in which the players control spaceships. Each player has a control box with three levers. The first one controls the acceleration of his spaceship, the second controls the rotation, and the third fires missiles. The object of the game is to hit the other guy's ship with a missile while you are avoiding getting caught in the sun's gravity.

The format is different from the previous examples because this program would presumably be written by a more sophisticated programmer who would find this format preferable. The computer should be able to understand either.

Don't bother to read this example too closely. It was included only as an example of how a more complicated problem might be solved. (This example was written at a different time from the others and may be inconsistent.)

```
OBJECT HAS   MASS TURTLE
             HEADING ACTION
             XFORCE YFORCE
             XACCELERATION YACCELERATION
             XVELOCITY YVELOCITY
             XPOSITION YPOSITION

   REPEATS   TELL TURTLE

     "ERASE SETXY XPOSITION YPOSITION SETHEADING

             DO ACTION


RULE   HAS   ACTION

   REPEATS   FOREACH OBJECT DO ACTION


SUN    HAS   TURTLE
             TELL TURTLE

     "PD FD 10 BK 5 RT 90 FD 5 BK 10"


SHIP  IS A   OBJECT
      MASS   IS 50
      HAS    FUEL
             LEVERBOX
    ACTION IS IF FUEL > 0 ADDTO  XACCELERATION
                                 LEVER1 * COS HEADING
                          ADDTO  YACCELERATION
                                 LEVER1 * SIN HEADING
                          ADDTO  HEADING
                                 LEVER2
                          ADDTO  FUEL
                                 - LEVER1
             IF LEVER3 > 0 MAKE  MISSILE
                                 XVELOCITY IS
                                  10 * COS HEADING
                                  + SHIP'S XVELOCITY
                                 YVELOCITY IS
                                  10 * SIN HEADING
                                  + SHIP'S YVELOCITY
```

```
MISSILE  IS A  OBJECT
           MASS  IS 10
         ACTION  IS TELL TURTLE

      "PU FD 3 RT 45 PD BK 3 FD 3 RT 90 FD 3"


STARFIELD    HAS TURTLE
             DO 100 TELL TURTLE

      "PU FD 100 * RANDOM RT 36 * RANDOM PD FD 1"


ENTERPRISE   IS A   SHIP
             LEVERBOX IS CONTROL1
             ACTION IS TELL TURTLE

      "PD BK 10 FD 15 FD 5 LT 90 FD 5 LT 90"


KLINGONSHIP  IS A  SHIP
             LEVERBOX IS CONTROL2
             ACTION IS TELL TURTLE

      "PD FD 5 PD RT 150 FD 10 RT 120 FD 10 RT 120"


THIRDLAW     IS A  RULE
             ACTION IS  ADDTO  XACCELERATION
                               XFORCE / MASS
                        ADDTO  YACCELERATION
                               YFORCE / MASS
                        ADDTO  XVELOCITY
                               XACCELERATION
                        ADDTO  YVELOCITY
                               YACCELERATION
                        ADDTO  XPOSITION
                               XVELOCITY
                        ADDTO  YPOSITION
                               YVELOCITY
```

```
GRAVITY      IS A  RULE
             ACTION IS

                 XFORCE IS XPOSITION * MASS /
                    (XPOSITION ** 2 + YPOSITION ** 2)

                 YFORCE IS YPOSITION * MASS /
                    (YPOSITION ** 2 + YPOSITION ** 2)


WARPING      IS A  RULE
             ACTION IS

       IF ABS XPOSITION > 390 XPOSITION IS -XPOSITION

       IF ABS YPOSITION > 390 YPOSITION IS -YPOSITION


COLLISION    IS A  RULE
             MYX     IS    XPOSITION
             MYY     IS    YPOSITION
             MYNAME  IS    OBJECT
             ACTION  IS    FOREACH OBJECT IF ALL

                                  MYY IS  YPOSITION
                                  MYX IS  XPOSITION
                                  MYNAME NOT OBJECT

                            UNMAKE MYNAME
                            UNMAKE OBJECT
                            SETXY MYX MYY
                            DO 180 TELL TURTLE

                               "PD FD 50 BK 50 RT 2"
```

## 8. Biology

Imagine that we have two kinds of creatures running around, SHEEP creatures and WOLF creatures. Both creatures have friends called STOMACH. When a creature moves his STOMACH gets emptier. If it ever gets completely empty the creature disappears. A WOLF's STOMACH gets full when it eats a SHEEP. This causes SHEEP to disappear. SHEEP eats grass, but grass just fills up their STOMACH a little bit, so SHEEP can't waste to much time running away from WOLF creatures. A WOLF can smell exactly how far away it is from a SHEEP, but it has no way of knowing in what direction it is. A SHEEP can see a WOLF if it is near enough and if it is not coming from behind.

What should **SHEEP** do? What should a **WOLF** do? What kinds of things does it depend on?

You decide how many of each creature there should be. I get to pick which kind I want. We each tell our creatures what to do, put them on the screen, and watch what happens. Sounds violent, but then so is **SPACEWAR.**

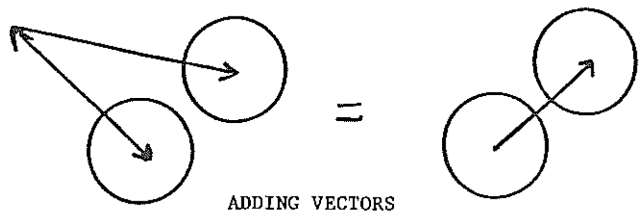
## 9. Build a Universe

Is it really possible that the whole universe is held together by just a few simple laws? I once really believed that if you knew F + MA everything else followed naturally. Taking physics at MIT disillusioned me. They kept on making up new ideas like torque, power, angular momentum, and fictitious forces. Then they did magic things with equations and integrals to "prove" to me that they were all derived. Yet somehow, I think that I might have been more convinced if I could have built a universe for myself, told it that F was equal to A, and watched the rest happen.

Let's start by building a particle. A nice way to imagine a particle is as a around thing that remembers all the forces that ever acted on it. There is actually a very good reason why a particle should be round. The only way that a force can act on it is exactly towards the center. Let's assume that each quantum of time a particle gets messages from all of the forces acting on it. The particle adds these to the list of all of the other forces that ever acted on it.

To figure out where it should be at t + 1, the particle just needs to point itself in the direction of the first force, walk forward the magnitude of the force divided by the mass of the particle, and then do the same for each of the other forces.

(What's this?? No x velocity and y velocity!! No sines and cosines and cross products!! I thought a particle at least had to know calculus.)


ADDING VECTORS

Now what is a force? Well a force is a creature that has two particles for friends. He sends a message consisting of a direction and a magnitude to one particle and then tells the other particle the same magnitude and the opposite direction.

But what happens when two particles try to be in the same place at the same time? We must do something to prevent that from happening. Let's imagine a particle running around tying to decide where it should be. In the process it bumps into another particle. Now the particle knows that it has N more steps to go in that direction, but it also knows that it can't go forward any more. It can reason as follows:

"If I had some force just big enough to push me N steps in the opposite direction then I could just pretend that I went those N steps and then turned around and came back. I guess I can just create myself a force big enough to do the job. But a force acts between two particles. Who should I give the other end to? How about this guy I just bumped into, he' the one that caused all the trouble in the first place..."

The particle now continues on, having added a force to both itself and the other particle. We have conquered collisions.

Of course our universe is not yet complete. We still need to tell it a few rules about when forces get created, especially gravity and the forces that bind particles into objects. But that is easy. We will not need to tell it about things like center of gravity, centrifugal force, and conservation of angular momentum. It will figure all of that out for itself.

Do you mean that's all you need to know to make a gyroscope work? Build a Universe and find out.

## 10. Recursion Step

Think up an even better computer.

## BUGS

Here is a list of problems that I have not even pretended to answer.

1) Are arithmetic functions creatures? (I have treated them as if they are not.)

2) What about numbers?

3) How do you debug a world if it doesn't work?

4) Should the syntax of the language be structured and computerish or vague and Englishish? (I have been using the latter, but I think I have changed my mind.)

5) What about bugs that happen because all of the creatures really don't act simultaneously?

6) How does one implement such a thing?

# NOTES ON THE FUTURE OF PROGRAMMING

Andy DiSessa
MIT Laboratory for Computer Science

I have been struck recently by conversations with a few people and by a few papers relating to the important topic of the future of programming and children. Opinions, of course, are all over the map. "Kids can't learn programming, its too hard." "Why teach Logo, it's unnecessarily simplified; there's no problem teaching young kids Lisp (or Scheme or Prolog or ...), which is, after all, more powerful and computationally pure than Logo?" "Programming is just a technical skill, truly useful to only a few; it is a fad in schools; it has no place in a liberal education and will die out." "Everyone will learn to program in the future; it will be a basic skill just like writing and reading, and will transform our intellectual culture just as profoundly as those 'skills' did."

I have my own positions on these issues, but that is not the point I wish to raise. What strikes me most about about these pronouncements is that almost all of them assume we know what programming is. They assume that, other methodological issues aside, present programming languages and contexts for learning programming are representative of future languages and contexts. I believe this to be false.

I do not wish to play futurist in this short article, and wax poetic about possibilities we can barely imagine. Instead, I wish to look conservatively to the near future[1] and make the simple remark: From an engineering point of view it is clear that, far from reaching an equilibrium, we have merely crossed a threshold in terms of making computation accessible and useful to students of all ages. If we wish to, we will be able to make rapid changes in what programming looks like, and in the sort of context students will have for learning programming.

I consider three dimension of change we can expect in the near future.

## Presentation:

The first type of change is *presentational*. These are not modifications to the underlying semantic of computation, but only to how it is visually (and potentially through other senses) presented and manipulated.

[1]In some cases, the recent past seems not yet to have been noticed.

Presentational changes will not in general affect the ultimate power or utility of a language, but most certainly can dramatically improve learnability and understandability. My arguments must be brief examples.

Logo's roots are in the teletype interfaces that were available when it got its start. The communication format between user and machine is a linear and "conversational." The user says (types) something, and the machine says something back. One of the problems with this format is simply that you cannot even point to something you "said" a few lines ago in order to "say" it again. Users must remember and type over (in the editor) what they just tried to make it into a procedure. There is no automatic trace of what was done which can easily be turned into a procedure.[1]

A second disadvantage of conversational interaction is that large scale structures are difficult to notate and manipulate as a whole. The Logo END command is really no command at all, but a syntactic marker of the boundary of an object (program). Unfortunately, END can easily be confused with parts of the program because it must look just like another piece of the conversation and cannot be connected visually with the previous part of the conversation, the start of the procedure definition, with which it should really be connected. A more profound but subtler problem is that you cannot directly see and manipulate the state of the system on the screen. Instead you must send a request to see some state (PRINTOUT), and if you want to change it, you must send another request (MAKE or TO). Recent microcomputer implementations of Logo incorporate a screen editor to ameliorate these problems to some extent, but this is only a patch. One still must make a major mode switch, entering the editor, if you want to at all pretend the the screen shows the state of the system. And the details of the relation of the editor buffer, the definition process and the defined state of the workspace are both invisible and subtle.

The bitmap display, a pointing device, and enough memory can solve all these problems. In Boxer, the language we are designing as a successor to Logo, we believe we have done this. Computational objects, such as programs, are visual units (boxes) and are trivially manipulable as a whole, somewhat like a large character in a text editor. More profoundly in Boxer we have changed the conversational interaction paradigm to "looking at and directly altering the state of the system." Boxer is "editor top level:"

[1]Of course, versions of *concrete programming* (as we call this way of making a program essentially by doing, step by step, what you want to have happen in the program) can be built, albeit somewhat clumsily. Various versions of what I started calling "instant" (single key activation) programs a number of years ago incorporate this feature.

you are always able to directly change or use anything you have put on the screen. Not only does this automatically give you a simple form of concrete programming, but we can support the profitable illusion that the user directly sees the system itself on the screen, and changes the system at any time, at any place, by just changing what he sees. We believe this not only makes the system easier to use, but easier to understand, and will allow students to progress to more advanced levels by, bit by bit, changing little pieces of a world they gradually create and see every time they use the system. Seeing your computational world is a much more important thing, especially in terms of long term development, than seeing only a recent audit trail of your conversation with the world.

Let me give one other simple but potentially important example of presentational changes in programming that show promise and can be trivially implemented with improved technology. Quite some years ago Radia Perlman and Danny Hillis constructed a device known as a slot machine. Children programmed by inserting cards representing commands into rows of slots, which represented programs. Not only could you see and directly manipulate the programs and their pieces, but the sequential activation of program steps, subprocedure calls and returns could be directly observed as lights lit up under each card when that step was executed. Apart from avoiding typing, and adding concreteness to computational objects, the slot machine provides tremendous help in making a model of the operation of a computational system like Logo.

With the advent of graphical objects like sprites, one can easily implement a slot machine on the video screen, moving icons around like cards. The screen slot machine can be freed of many of the limitations of the physical one: on a screen, it is easy to invent spatial/graphical representations for input parameters and conditionals which did not exist on the slot machine; adding symbols (e.g. by typing a new word) is trivial in the screen version (making a new card is not so easy); most importantly, the process of abstraction can be easily represented on the screen -- e.g. a spatial sequence of card icons forming a program, can slide together over one another, like a spread out deck of cards being pushed back together, and be given a top "cover" icon, becoming a single unit like all the supplied primitive card/icons.

It is not clear how much help these and other presentational changes can be, but certainly they will be of some help, and potentially a very great amount.

## Form:

The second type of change in programming is change in *form*. This kind goes beyond the surface presentational changes mentioned above and alters the fundamental computational semantic. Since these changes are more complex and difficult to describe, I shall have to be quite elliptical here.

For those who know actor-oriented programming, as represented by Smalltalk, or Logic programming, as represented by Prolog, it should be clear that significantly different programming forms are already in existence, waiting to be explored. While I do not think (though I will not argue it here) that these ideas fundamentally improve the understandability of programming, I think their potential lies in indirectly altering the context of computational learning, to be discussed below. So instead, I will briefly discuss two other paradigms of programming which are yet to be fully defined, but which I believe offer at least as great a promise as present *avant guard* languages.

*Device programming* is motivated by the image of an electronic or mechanical device consisting of a number of components of a few classes, like resistors, transistors and capacitors; or pipes, pumps and reservoirs. These components have relatively simple behavior and achieve the functionality of the device by being hooked together at their terminals into a network of components. Computationally, we want graphical components which can be manually assembled into devices by connecting their inputs and outputs with "wires" (lines drawn on the screen). The wires communicate messages of an arbitrary symbolic kind, which could, for example, represent flow of substance or electricity by passing numbers representing amounts. Each component knows when it gets a message at an input and can compute and send output messages as it sees fit. Device programming is a significantly different form than, say Logo or Smalltalk because of its explicitly parallel nature of computation, and its explicit representation of dataflow rather than, say, sequence. On the other hand, device programming can simulate, for example, a function as a component with one input and one output. Activation of the function amounts to simply giving it an input. Furthermore, a component can be built out of very little more than a procedural programming language to express the actions to be taken to compute outputs from inputs.

Naturally, it would be important to have a general abstraction mechanism so that devices could be made into components. Some set of free inputs and outputs in a device extend beyond the boundary of the device to act as terminals of the new abstracted component, and visually the parts of the device-become-component can shrink and/or acquire a new surface form to hide detail. Likely one would like the surface form to show some small part of the internal state of the device.

Device programming is attractive because it has such a simple and graphic method of combining element to make compound things. There is reason to believe it can have some intuitive accessibility that the hidden dataflow and complex sequencing of pure procedural programming does not. Lastly, it opens doors to more easily simulating and thus coming to understand an important class of physical computations our world does: One can even engage in the delightfully recursive task of constructing a computer out of computationally implemented components, emulating every level of abstraction of a physical computer. As a minimalist demonstration of the power of device programming, it is clearly a powerful generalization of the popular computer game Rocky's Boots, which would be a totally trivial task to construct in a device programming system.

I shall have to be very cryptic, I am afraid, with my second example of new forms of programming. It is motivated by the observation that one of the fundamental problems with sequential programming is that it requires one to imagine abstractly, with very little help from the system, all the possible contexts (states) a program can get itself into, and make provision for dealing with them. How many software bugs come from the programmer simply never imagining that a user would want to stop in the middle of doing one thing, and start something else that the program was, therefore, unprepared respond to. Timing bugs, "oh, I forgot that case" in multiply branched conditionals, etc., are all examples of this difficulty. The problem is, in its crudest form, simply not having any abstractions at all of the appropriate level and kind to deal with the large space of possible control states of an ongoing process. How much easier would programming be if contexts (states) were explicitly represented objects in the system? Even better, one should be able to instantiate a state by name or pointing to a context object. One could gradually lay out, one by one, all the contexts and subcontexts of the operating program, and name, reorganize, and in other ways deal with them concretely. Henry Lieberman's Tinker programming style would be ideally suited to a context programming system: When a new context arises in which one has not specified what to do, the system automatically reifies that state as an context-object, and one is thus spared even much of the burden of making all the appropriate context-objects in advance.

A simple version of context programming could solve some of the problems of controlling demon activated systems like some sprite Logos. A context could, at minimum, be a package of demons and processes which one can turn on or off at will, and within which one can specify transitions to other contexts (packages of parallel processes and demons). Without fleshing out the detail, still I hope it is apparent that some of the problems of controlling parallel processes helped if not solved at least be helped by *context programming*, making interactive game programming significantly easier.

I should remark in passing the Boxer also makes some profitable changes of the form of programming, but these are described elsewhere.

## Context:

Finally, the third immediately available dimension of change in programming is *context*, what you do with your programming system. Turtle graphics is a crucial part of the advance of Logo over previous languages. It is motivating; it allows children to immediately set goals they understand (drawing pictures), yet it can evolve naturally and slowly into a medium of contact with profound mathematics. Again the old story of natural language tells the tale. It is an incredibly complex and large learning task which, nonetheless, nearly every child masters because it can be mastered one tiny bit at a time, and is useful to the child at every step along the way.

Boxer enlarges the scope of programming's context to include text production and manipulation, and the organization and manipulation of many other sorts of data. If a programming system is literally also a child's book and pencil (text editor in modern parlance), and if he can, bit by bit, modify, extend and personalize not only what comes in his book, but also the form of the presentational medium, then programming becomes a learnable-in-tiny-increments and constantly useful extension to written communication, something with which children are in constant contact. A simple example of such modifications is to add a new editing command, or to use a variable as a means of keeping around a template for electronic mail messages or other "forms."

I hope the reader has enough imagination to see that device and context programming described above may open up contexts for learning and using programming at least as good as turtle graphics and the data worlds of Boxer. Prolog adds another model, where assertions and inferences in the computational system allow incrementally learnable advantage over "expertise about something" expressed in other media, say, conversational reasoning. The list can be made much longer, and each item requires a long discussion of its feasibility and impact. But energetic designers and optimists don't need the discussion, and skeptics will not be convinced in another page.

I have long since run out of time to present and argue my case. So I will simply restate it: *By all means, let us spend time investigating how to teach and what effects present models of programming can have. But be aware that, if we choose to continue walking, the ground will inevitably change under us. Let's not be so preoccupied with the sand on the shore, that we do not move at least some distance inland to see if anything will grow in this new continent.*

# ABSTRACTS

ABSTRACTS ARRANGED ALPHABETICALLY BY LAST NAME OF PRESENTER

ACHBERGER, Fred *Educational Service District 114, WA*

***Logo: Tricks or Topics***

This session will discuss ideas such as text-screen animation, auto-start microworlds, Logo version conversion, and non-keyboard input. Are these topics for discussion in a Logo class or are they tricks for the microworld designer?

ANDERSEN, Lyle and BLANKESPOOR, Gilbert *Augustana College*

***Integrating Mathematics and Computers (Logo Language) with Science Activities***

A series of science activities (mostly biology) into which mathematics and Logo computer experiences have been incorporated.

AREY, Temple *The Carroll School*

WEIR, Sylvia *Massachusotts Institute of Technology*

***Building Bridges from Logo to School Mathematics***

Logo provides the empirical window through which we were able to view an ability previously hidden. There are many children in our school systems penalized by heavily language-based curricula, yet unaware of the connection between their spatial ability and doing math. The connection needs to be made explicit rather than be left to emerge during standard Logo activities. Bridges need to be built so that the Logo work is intimately integrated into the standard classroom curriculum in a way that builds on the ability of children to perform perceptually based computation. An example of a collection of pencil-and-paper activities developed at the Carroll School will be presented.

BIRCH, Allison and DAVIDSON, Larry *The Phoenix School*

***Logo Explorations in Language and Algebra***

Demonstrations of a variety of non-graphics Logo projects at the high school level. Projects include linguistic and mathematical applications as well as data base system.

BLITMAN, Elaine *Punahou School*

***Logo at Punahou School***

A slide-tape presentation describing the Logo program for 800 kindergarten through fourth grade children at Punahou School in Honolulu, Hawaii.

BOUCHARD, Louisette and EMIRKANIAN, Louisette,
*Universite du Quebec a Montreal*

***Use of Logo in the Teaching of French***

Three experiments which use Logo in the teaching of French will be presented. The first experiment, a classical CAI approach to the teaching of relative clauses, makes use of the differences observed between proposed and expected responses to select helpful hints. The second experiment involves learning about clause coordination by using a language manipulator based on Logo. The third experiment involves learning by teaching the computer the grammar rules of relative clauses. Apple IIe Logo programs will be exhibited for each experiment.

BRISKMAN, David *Cornell University*

***Logo and Physics***

Simple physics simulations done in LCSI SPrite Logo demonstraté the power of Logo and the ability of Logo to be used in higher education as a learning tool.

BROWN, Eric *Logo Computer Systems Inc.*

***A Logo Authoring System***

Workshop on using Logo to program computer tutorials. Various programs will be described and the basic principles common to authoring systems will be discussed. Those utility procedures which would be transferable to many instructional programs will be made available to participants.

BULL, Glen *University of Virginia*

***Talking With Logo: Logo in Speech, Hearing and Language***

The following applications will be explored: 1) using a VOTRAX speech synthesizer to generate speech communications with a three line Logo SAY procedure; 2) using Logo sentence generators in language arts. with applications of words and lists in interactive language therapy (extensions include use with speech synthesizers); and 3) using the Apple game port for applications involving communication with the outside world. Examples include using Logo as an alternative communication device for patients without oral language, through use of a touch tablet, and using Logo to simulate the front panel of an audiometer by interconnecting the front panel to the Apple through the game port.

BURNETT, J. Dale and HIGGINSON, William C. *Queen's University*
***Logo and the Reality of Elementary Classrooms: A Report on the "Creative Uses" Project at Queen's University (1982-1984)***

In February of 1982, the Ministry of Education of the Province of Ontario issued a policy statement on the use of computers in teaching and learning. It stated that primary emphasis should be placed on "the creative use of computers by individuals in writing, composing, designing, analyzing and other extensions of original thought." It went on to add that "all students must be given opportunities to use computers in this way." From September of 1982 to June of 1984 a study of the implications of this policy was carried out in thirteen elementary classrooms by a team of researchers from Queen's University. Logo was the main focus for this study. A preliminary report on the findings of the study with particular reference to its implications for curriculum, classroom practice and teacher education will be given by the Project Co-Directors.

CARVER, Vicki
***Logo-Based Job Training for Inner-City Adults***

The session will focus on a six-month CETA job training program conducted last year at a black community center in Des Moines, Iowa. The program was controversial, successful and fun. It is hoped that the presentation will encourage the use of Logo in similar work with chronically unemployed or discouraged adults.

CHAPIN, Suzanne and HOLDEN, Susan *The Meadowbrook School*
***Interdisciplinary Logo***

This session will present uses of Logo in middle school subject areas. Ideas for Logo in music, poetry, language arts, social studies and math will be shared. This includes graphics and creating original programs using words and lists. There will be opportunities for the participants to try some of these activities and to discuss their implementation in the classroom.

CLEMENTS, Douglas H. *Kent State University*
***Effects of Logo Programming on Cognitive Style and Cognitive Development***

A study with first grade children assessed the effects of computer programming in Logo on children's cognitive style (reflectivity, divergent thinking), metacognitive ability, cognitive development (operational competence, general cognitive measures) and ability to describe directions. The children were randomly assigned to either of two treatments, Logo programming or computer-assisted instruction.

COHEN, Rina — *Ontario Institute for Studies in Education*

***The Logo Microworlds Project at OISE***

This presentation will report on a project related to the development of modules which are intended for use as resource materials by teachers using Logo. The modules provide bridges between existing curricula and teaching/learning techniques and those that Logo offers. They consist of two types of software packages and accompanying guidelines: 1) Logo microworlds which permit focused exploration in limited domains but with a broad range of possible goals, and 2) utilities which help the teacher use Logo in the classroom more easily. The modules are being developed at OISE and field tested in seven elementary classrooms.

DALE, Evelyn

***Logo as a Tool for Studying Physics***

Presentation of work performed while teaching a ten week Logo/Physics course to fifth and sixth graders at the Frontier School in Rio Linda, Calif. Logo topics included turtle graphics, arithmetic operations, and word and list manipulation. Physics topics included free fall, vectors, projectile motion and the CRT.

DAVID, Andrew

***Where are the Microworld Designers?***

The session will focus on a process of "micro-world design" reducing curriculum content areas to their unique, extensible primitive vocabularies, and other topics dealing with the issue of establishing a more effective flow of communication between developers and educators. Educators, programmers, developers, and other interested individuals are invited to share their ideas on these topics and to see examples of projects in progress.

DISESSA, Andy — *Massachusetts Institute of Technology*
GLOBERSON, Tamar — *Tel-Aviv University/MIT*

***Investigating the Effect of Age and Cognitive Style on Children's Intuitions of Motion Using Concrete and Computer Tasks***

This session will focus on an investigation of the development of intuitive notions of motion in third- and sixth-grade children specifically with regard to the following issues: 1) how systematic, theory-like, are children's knowledge structures across different task situations; 2) how robust these knowledge structures are *vis a vis* conflicting (perceptual and conceptual) information; 3) the extent to which they develop with age into a more systematic theory; and 4) to what extent do children's cognitive styles affect the above. Tasks consisted of both concrete materials and computer activities.

ELTSCHINGER, Michel and CROWTHER, Sandra
***Friends of the Turtle***

This presentation will look at various pieces of equipment (bought and homemade) that enhance the introduction and use of Logo. The session will center around a floor turtle that is remote-controlled by students at the computer.

FEURZEIG, Wallace *BBN Laboratories*
GOLDENBERG, E. Paul *Lincoln-Sudbury Regional High School*
***Learning Language with Logo***

The presentation will focus on the development of a new, qualitatively different language arts curriculum in which Logo concepts and projects are central to the presentation of the subject matter. We will discuss the course and give .extensive examples of the approach, methods, and materials being developed.

FIRE DOG, Peter *University of Minnesota*
***Logo Effects in Public School Classrooms***

Although Logo may enhance cognitive or intellectual capabilities in the individual, it is l;likely that Logo effects are mediated by the social and environmental contexts into which the language-philosophy is introduced. Data on a 1983 sample of sixteen K-12 mainstream and special education classrooms in the St. Paul, MN, public schools (N=385 students) indicate that 1) academic improvement as a result of working with Logo occurs independently of a student's achievement rank or social status in the classroom; 2) dramatic behavioral or learning changes may be expected for as many as 10% of the students in comparable (urban public school) settings; 3) Logo effects are more likely to be seen in classrooms where mean achievement scores are below average and student learning characteristics are heterogeneous; and 4) Logo effects may be most dramatic when teachers use it actively and intentionally as a social integration mechanism. 1984 panel data on a second sample of 28 classrooms (N = 620 students) further investigates variations in classroom environment and Logo effects, with special focus on primary grades and traditional low achievement subgroups.

FISHER, Glenn *Alameda County Supt. of Schools Office*
***Training Teachers to Use Logo***

Course outlines and approaches for introducing teachers to Logo and for using Logo as a problem-solving tool in the curriculum, with some samples of teacher-developed materials.

FLAKE, Janice L.
***Logo as a Part of an Elementary Teacher's Preparation***

In order for Logo to reach the children, it needs to become a part of elementary teachers' background. We have asked our undergraduates to learn some Logo as part of a *How Children Learn Mathematics* course in our undergraduate program at Florida State University. We are also building a course in problem solving via the microcomputer.

HAAS, Jeff *FOLLK*
***Advanced Logo and Artificial Intelligence***

Topics to be discussed/demonstrated include: 1) list processing used with turtle graphics, 2) real world applications of list processing in Logo, and 3) artificial intelligence using Logo (including Winograd's Blocks world).

HARPER, Dennis O. *University of California, Santa Barbara*
***Logo in Malaysia***

The session will focus on research completed in Malaysia concerning the use of Logo by teachers and students in rural Malaysian settings and the materials developed in Malay for use in the schools.

HILLEL, J. *Concordia University*
***Mathematical Concepts and Programming Skills Acquired by 8-Year-Olds in a Restricted Logo Environment***

Six pairs of 8-year-olds were each observed for twelve hours. Using an observation grid, hypotheses were generated after each session as to the emerging mathematical/programming knowledge. Specific, short tasks were then assigned to the children so as to verify the hypotheses and to provide continuous assessment.

HOYLES, Celia and SUTHERLAND, Rosamund
*Polytechnic of North London*

***Intervention Strategies and Collaboration in Learning Logo***

Case studies of 11- and 12-year-old children working with Logo in their "normal" mathematics lessons have been undertaken with the researchers acting as participant observers. A process as opposed to a goal directed approach to learning Logo has been adopted. The main issues of the research are: 1) how the dialogue between the two pupils in terms of its cognitive and communicative function relates to interactions with the computer and to the development of mathematical processes and programming skills; 2) when and why teaching interventions are necessary; and the different forms these interventions take. The overall research design and categories of analysis will be presented. Details will be given of one pair of children with extracts of their interactions and computer work as illustration of the categories developed and of sequences of collaborative learning.

KOZBERG, Geraldine *St. Paul Public Schools*

***Logo and Educational Change***

Logo, by itself, will not solve the problems of big city schools. Logo--as part of a larger change effort--is a powerful force in effecting substantive change in learning and learning environments. The Community/School Collaborative, a K-12 microsystem of seven schools in St. Paul, Minn., is a visible and viable expression of a long-range Logo-based educational change process.

KULL, Judith *University of New Hampshire*
STRONG, Joyce Shea *Oyster River Schools*
COHEN, Bernard *Little Harbor School*

***Learning and Logo: Collaborative Research in the First Grade***

Concern about what children are learning with Logo prompted the effort to sit beside them and watch, ask, listen, and learn. A collaborative research team including two first-grade teachers, a university professor of education and two graduate students documented the behavior of first-grade children learning Logo in their classrooms. Results of the year-long study including behavior related to gaining control over the system, problem-solving, self-concept and debugging will be discussed and illustrated. Also discussed will be implications for teaching and

applications of the field-based observational research model used in the study.

LERON, Uri — *Technion-Israel Institute of Technology*

***Quasi-Piagetian Learning in Logo***

In Papert's vision, children in a Logo environment learn in a totally spontaneous, non-directed fashion which he calls "Piagetian Learning." Experience has shown, however, that under such circumstances, children often do not acquire the "powerful ideas" that form the other significant component of the Logo package (eg. the effective use of subprocedures). Fortunately, it appears that there is enough "educational room" for the teacher to assume a somewhat more directive role, and for the learning environment to become somewhat more structured, without upsetting most of the attractive features of Piagetian learning such as being meaningful, exploratory, non-judgmental, and non-threatening. The talk will elaborate on this modified learning style, tentatively named "Quasi-Piagetian Learning," and will report on studies and development work that has been carried out over the last several years in Israel.

**LOUIE, Steve and LEFEVRE, Judy**

*National Advisory Council for Computer Implementation in School*

***NACCIS Performance Methodology Project***

The NACCIS Performance Methodology Project involves Children building microcommunities wherein they design, develop, produce and market products. The simulation permits students to gain experience in working with word processing, data base managers and the Logo programming language. These software tools are used in the creation of production journals, product design aides and communications vehicles in a real-world interactive prosocial learning environment. Particular attention has been paid in designing the program to enhance shifts toward an internal locus of control, and effective transfer of learning has been optimized through a system of progressive advancement from hypothesis generation to actual "trading" of products and services at Computer Recitals and Trade Fairs. The researchers will present a "work-in-progress" report of NACCIS research.

**MCCAULEY, Jim** — *Santa Clara County Office of Education*

***Introduction to List Processing Through Fantasy***

Logo's turtle is actually a lowly output device disguised by a remarkably effective fantasy, enabling learners to avoid a lot of "computerese" and move directly to learning interesting things about geometry and thinking. This session will focus on other powerfully enabling fantasies which have been developed to enable learners to skip the "computerese" associated with list processing. Included will be an adventure in recursive dragon-hunting in Logo, with handouts describing teaching methods and sample code.

MCDIARMID, Roxanne C.
***Languaging Through Logo***

A presentation/discussion will be given of a method for applying the Logo philosophy of natural thinking in the regular, primary classroom. Children's products will be used to demonstrate the potential for individualized curricula, through the illuminations of an expanding microworld.

MICHAUD, Nicole *Logo Computer Systems Inc.*
***Logo: A Mirror for Learning Personalities***

In this workshop, I will draw from my experience teaching Logo to normal and emotionally disturbed children, as well as to adults, to detail those aspects of emotional and intellectual style which the Logo learning experience reflects back to the observer. A central point here is how the learner can become aware of his or her emotional and intellectual approaches to learning, and how qualities like responsibility, self-confidence, initiative, dependency, resistance and rigidity can evolve in the course of learning to program in Logo.

MILLER, Laurence *Creative Learning Services*
***Why Logo Needs the Schools and the Schools Need Logo***

Over the past few years, there has been a growing alienation to the schools and school systems on the part of the Logo community based on the belief that schools are conservative by their very nature and are therefore repressive of the autonomy that the best learning requires. Based on experience with a variety of educational institutions, this view seems too pessimistic. This presentation will focus on the argument that school systems offer the most promising channel for achieving real reform, and that educational course materials suitable for use in the public schools should be developed which demonstrate, even in the traditional terms of academic achievement, the superiority of understanding learning within a psychological framework based on Piaget.

MILOJKOVIC, James D. *Stanford University*

***Learning Advanced Logo***

An illustrated discussion of strategies and tactics for exploring the advanced features of Logo.

REGGINI, Horacio

***Logo in Latin America***

The insertion of computers in society, in general and in education in particular, have had in Latin America distinctive and proper characteristics. In accordance with its political, economic, social and cultural reality. Respect for linguistic usages and local idiosyncrasies, and the new educational and computational perspectives inherent to Logo, have gained an ever increasing number of followers and are some of the reasons for the advancement of Logo as a social phenomenon within the Latin American community. Hence the first seed, sown in Argentina some years back, has spread all over the continent and is presently growing in a steady, natural way.

SHARP, Pamela *San Francisco State University*

***The Aesthetics of Logo and Instruction in the Arts***

Artistically relevant interaction with works of art requires the ability to recognize aesthetic qualities. Growth in this ability may be viewed as a process of differentiation of the features which distinguish one quality from another. Natural language plays an essential role in such differentiation. Logo is a computer language of form--descriptive as well as visual. It's use in developing response to sensory, expressive, and formal qualities of works of art in elementary and secondary programs of arts instruction is discussed, demonstrated and evaluated.

SILVERMAN, Brian *Logo Computer Systems Inc.*

***Perspectives on Turtle Graphics***

The presentation will focus on a project on perspective drawing in Logo. The project involved writing an extended set of Turtle Geometry procedures to make representations of three-dimensional objects on the computer screen. In the course of the project, various approaches were tried using different methods for mapping three-dimensional into two-dimensional space. The most recent version of the 3-D graphics program was designed to take into account an aspect of the problem which may not be obvious at first--the position of the observer in relation to the screen.

STAVELY, Tony — *Keene State College*

***List Processing Tools***

All Logo list processing procedures utilize the same underlying form, "the basic recursive module." Using such procedures it is possible to search, concentrate, sort, transform and otherwise manipulate words and lists in Logo to perform statistical calculations, alphabetize words, create interactive games and even make a kind of PacTurtle program. Participants in the session will be able to see and use examples of such procedures.

SUTTIN, Dan — *Cambridge Montessori School*

***Logo as a Medium for Creating Special Effects***

The students of the Cambridge Montessori School prepared a multi-media dramatic production of "Return of the Jedi" consisting of live acting integrated with videotaped special effects including film cuts, comic book pictures, and Logo programs. The students worked with the teacher on all facets of the production including the Logo programs and the videotaping, with the support of Atari Cambridge Research. A videotape of the production will be shown during the session as well as the Logo programs which were a part of it.

SYLLA, Fatimata Seye

*Ministry of Scientific and Technological Research of Senegal*

***The Senegalese Project: Computers in Education***

Initiated by the Senegalese government and the World Center for Computation and Human Resources, The Senegalese Project: Computers in Education was launched in March 1981 in Dakar. The purpose of this on-going experiment is to evaluate the impact of computers on the Senegalese educational system. It is based on the use of Logo by young children of different backgrounds aged 8 to 11 years old.

TECHNICAL EDUCATION RESEARCH CENTERS

***Math and Science Investigations Using Logo***

As an interactive language, Logo offers potential for Math and Science investigations. We will demonstrate several examples, including sharing what teachers and students have done using these programs.

TECHNICAL EDUCATION RESEARCH CENTERS

***Teacher Workshops - Examples of Workshop Challenges and Teacher Creations***

TERC conducts a wide variety of teacher workshops on using Logo in the classroom. This session will present some of the workshop material we use and show examples of teacher programs created within the workshop context.

TEMPEL, Michael, NELSON, Harry and MICHAUD, Nicole

*Logo Computer Systems Inc.*

***Logo Training: Some Experiences and Recommendations for Change***

This workshop will focus on an examination of the misconceptions and incomplete models which students often bring to advanced Logo work based on their introductory experiences with Logo, eg. a poor understanding of procedurality and recursion, weak models of the workspace and file system, and lack of an overall model of Logo. Based on observations of students in advanced Logo training sessions, some ways for improving introductory Logo courses will be suggested.

TEMPLAR, Chris *Johnson Bible College*

***Color Logo Animated Film Production***

This presentation will include a description of the way in which a group of sixth-grade students, who had been introduced to Logo last November, used standard Logo primitives together with SHAPE and HATCH to develop animated scenes. The stages of their project's development from single scenes to a completed story will be demonstrated. As a group, these students of varying abilities discovered ways to use expanded Logo to achieve their desired goals and as well as a knowledge of the language. Story writing and film making techniques were also covered during the process of transferring the material from the computer to the video recorder.

TIPPS, Steve *University of Virginia*

***Plotting With Logo***

Demonstration will include: plotting student projects with Logo; plotting classroom projects and banners for displays; plotting teacher materials; conversion of Logo to Plotter Logo; and hands-on plotting of your favorite Logo procedure. The Sweet-P plotter will be used to demonstrate. Other plotters may also be available for demonstration.

VALENTE, Jose Armando *Universidade Estadual de Campinas*

***Computer-based Environment for the Handicapped***

The presentation will include the use of Logo as a diagnostic and remedial tool for physically handicapped children. Case studies will be presented which will highlight such topics as educational philosophy, cognitive development and motivation. Current projects will also be presented which elaborate on the diagnostic aspect of Logo.

WAPPLER, Reinhold

***Teaching Structured Logo***

Presented are the workbooks and other materials evolved in two years of teaching a structured and goal-oriented course in math and Logo to second through fifth grades. Seventy-five pages of workbook exercises are devoted to regular polygons, coordinate geometry, variables, procedure writing, recursion, and other relevant disciplines. Selected video tapes of classroom activities will be shown.

WATT, Molly *Educational Alternatives*
WATT, Dan *Educational Alternatives/Popular Computing Magazine*

***Ten Steps to Creating a Microworld***

Presentation based on their forthcoming book *Teaching With Logo*.

WATT, Molly *Educational Alternatives*
WATT, Dan *Educational Alternatives/Popular Computing Magazine*
STAVELY, Tony *Keene State College*

***Creating a Logo Culture a la Monadnock Logo Users' Group***

Representatives from MLUG will present their group: why it was formed, how it works, issues and ongoing themes. The session will be conducted in the style of the group's usual format.

# BIBLIOGRAPHY

This bibliography has been compiled by Tom Lough, with assistance from Barbara Elias, Carolyn Markuson, Regina Sapona, Joyce Tobias and Griff Wigley. Corrections and additional entries can be submitted to Logo Bibliography, c/o National Logo Exchange, P.O. Box 5341, Charlottesville, VA 22905.

## 1. BOOKS

PUBLISHERS' ADDRESSES CAN BE FOUND AT THE END OF THE BIBLIOGRAPHY

### 1.1. Books about Logo

Abe, Setsuko, Logo, CBS/Sony Publications

Abe, Setsuko, Logo Book, Geodesic

Abelson, Harold, Apple Logo, McGraw-Hill

Abelson, Harold, Logo for the Apple II, McGraw-Hill

Abelson, Harold, TI Logo, McGraw-Hill

Abelson, Harold, and diSessa, Andrea, Turtle Geometry: The Computer as a Medium for Exploring Mathematics, MIT Press

Allen, John, et al., Thinking About [TLC] Logo, Holt, Rinehart and Winston

Bailey, Harold et al., Apple Logo: Activities for Turtle Graphics, publisher unknown

Bearden, Donna, A Bit of Logo Magic, Reston Publishing Co.

Bearden, Donna, 1,2,3, My Computer and Me: A Logo Funbook for Kids, Reston Publishing Co.

Bearden, Donna et al., The Turtle's Sourcebook, Reston Publishing Co.

Birch, Louisa, Introducing Logo to the Primary Child, Houghton Mifflin

Bitter, Gary, and Watson, Nancy, Apple Logo Primer, Reston Publishing Co.

Bitter, Gary, and Watson, Nancy, Commodore 64 Logo Primer, Reston Publishing Co.

Bitter, Gary, and Watson, Nancy, CyberLogo Primer, Reston Publishing Co.

Burnett, Dale, Logo: An Introduction for Teachers, Creative Computing Press

Burrowes, Sharon et al., Beyond Mindstorms: Teaching with IBM Logo, Holt, Rinehart and Winston

Burrowes, Sharon et al., Exploring IBM Logo: A Guide for Adults, Holt, Rinehart and Winston

Conlan, Jim, and Inman, Don, Sprites, A Turtle, and TI Logo, Reston Publishing Co.

Feurzeig, Wallace et al., The Logo Language: Learning Mathematics through Programming, Entelek

Feurzeig, Wallace et al., An Introductory Logo Teaching Sequence: Logo TeachingSequence: Logo Teaching on Logic, Logo Reference Manual, Bolt, Beranek and Newman, Inc.

Feurzeig, Wallace, et al., The Logo-S Language and the Portable Logo System, Volumes I & II, Bolt, Beranek and Newman, Inc.

Goldenberg, E. Paul, Special Technology for Special Children, University Park Press

Green, Carolyn, and Jaeger, Christi, Teacher, Kids, and Logo, EduComp Publications

Kwok, Annie et al., The Workbook for Learning Logo, Turtle Publishing

Martin, Kathleen and Bearden, Donna, Polyspi - Inspi, Martin - Bearden, Inc.

Martin, Kathleen and Bearden, Donna, Primarily Logo, Reston Publishing Co.

Martin, Kathleen and Bearden, Donna, The Rule of 360, Martin - Bearden, Inc.

Martin, Kathleen, and Bearden, Donna,.The Turtle Goes to Kindergarten, Martin - Bearden, Inc.

Masalski, William,, Run with Logo, Milton Bradley

Mass, Lynne et al., Kids Working with Logo: An Apple Logo Manual for Turtle Graphics, Trillium Press

McLean, John, Understanding Logo, publisher unknown

McDougall, Anne et al., Learning Logo on the Apple II, Prentice Hall International

McDougall, Anne et al., Learning Logo on the Commodore 64, Prentice Hall International

McDougall, Anne et al., Learning Logo on the Tandy Color Computer, Prentice Hall International

Moore, Margaret, Geometry Problems for Logo Discoveries, Creative Publications

Moore, Margaret, Logo Discoveries, Creative Publications

Musha, Diane R., et al., TI Logo Curriculum Guide, Texas Insturments, Inc.

Nelson, Harold, Advanced Logo: Computing About Thinking, Addison Wesley

Nevile, Liddy, and Dowling, Carolyn, Let's Talk Apple Turtle, Prentice Hall International

Nevile, Liddy, and Dowling, Carolyn, Let's Talk Commodore Turtle, Prentice Hall International

Papert, Seymour, Jaillissement de l'Esprit: Ordinateurs et Apprentissage, Flammarion Press

Papert, Seymour, Mindstorms: Computers, Children, and Powerful Ideas, Basic Books

Parker, Pat, Logo for Fun, Scholastic Inc.

Peddicord, Richard, Understanding Logo, Alfred Publishing Co., Inc.

Poirot, James, and Adams, R. Clark, 40 Easy Steps to Programming in BASIC and Logo, Sterling Swift

Reggini, Horacio, Alas para la Mente - Logo: Un Lenguaje de Computadoras y un Estilo de Pensar, Emece Distribuidora

Reggini, Horacio, Des Ailes pour l'Esprit - Logo: Language des Ordinateurs et Forme de Penser, Emece Distribuidora

Ross, Peter, Introducing Logo for the Texas Instruments 99/4A, Tandy Color Computer, and the Apple II Computer, Addison Wesley

Ross, Peter, Logo Programming, Addison Wesley

Rowley, Tom, and Leckrone, Ron, The Big Trak Book: Your Computer on Wheels, Reston Publishing Co.

Sharp, Pamela, Cyberlogo Turtle: A First Step in Computer Literacy, Cybertronics, International, Reston Publishing Co.

Sharp, Pamela, TURTLESTEPS: An Introduction to Apple Logo, and Terrapin Logo, R. J. Brady Company

Sharp,Pamela, TURTLESTEPS: An Introduction to Atari Logo, R. J. Brady Company

Sharp, Pamela, TURTLESTEPS: An Introduction to IBM Logo and Dr. Logo, R. J. Brady Company

Stanger, Donna et al., Learning through Logo, Sunburst

Stultz, Russell A., Illustrated TI Logo Dictionary, Prentice Hall Inc.

Tobias, Joyce and Markuson, Carolyn, Adventures in Logo, McGraw- Hill

Tobias, Joyce and Markuson, Carolyn, More Adventures in Logo, McGraw-Hill

Thornburg, David, Computer Art and Animation: A Users Guide to TI 99/4A Color Logo, Addison Wesley

Thornburg, David, Discovering Apple Logo: An Invitation to the Art and Pattern of Nature, Addison Wesley

Thornburg, David, Picture This! An Introduction to Computer Graphics for Kids of All Ages, Addison Wesley

Thornburg, David, Picture This Too! An Introduction to Computer Graphics for Kids of All Ages, Addison Wesley

Tipps, Steve, et al., Nudges: 100 IBM Projects for Children, Holt, Rinehart and Winston

Watt, Dan, Learning with Apple Logo, McGraw-Hill

Watt, Dan, Learning with Atari Logo, McGraw-Hill

Watt, Dan, Learning with Commodore Logo, McGraw-Hill

Watt, Dan, Learning with Logo, McGraw-Hill

Watt, Molly, and Watt, Dan, Teaching with Logo, Addison Wesley

Webb, Joan et al., Explorer's Guide to Apple Logo (LCSI), Hayden Book Co., Inc.

Webb, Joan et al., Explorer's Guide to Apple Logo (MIT), Hayden Book Co. Inc.

Weidenfeld, Gerard et al., Logo, Eyrolles

Weir, Sylvia, Untrapping Intelligence: Logo and Special Learning, Harper and Row (in press)

Zamora, Ramon and Albrecht, Bob, Teach Yourself TI Logo, McGraw-Hill

*Author unknown*, Turtle Tricks: An Introduction to Turtle Graphics and Apple Logo, Stokes Publishing Company

## 1.2. Books with significant Logo references

Hagen, Dolores, Microcomputer Resource Book for Special Education, Reston Publishing

Harper, Dennis, and Stewart, James H., "Run: Computer Education," Brooks / Cole Publishing Company

Papert, Seymour, The Computer Age: A Twenty Year View, MIT Press

Peterson, Dale (ed.), Intelligent Schoolhouse: Readings on Computers and Learning, Reston Publishing

Taylor, Robert P. (ed.), The Computer in the School: Tutor, Tool, Tutee, Teachers College Press

Yazdani, Masoud, New Horizons in Educational Computing, John Wiley & Sons Inc.

## 2. PERIODICALS

### 2.1. Periodicals dedicated to Logo

Boletin de la Asociacion Amigos de Logo, Salguero 2969, 1425 Buenos Aires, Argentina

Dr. Logo Newsletter, Digital Research, PO Box 579, Pacific Grove, CA 93950

FOLLKlore, PO Box 22094, San Francisco, CA 94122

Logo Journal, Hiroyoshi Goto, Uny Co, BYNAS Div., 3F Dainagoya Building, 28-12 3Chome Meieki, Nakamura-ku, Nagoya, Japan 450

Logo and Educational Computing Journal, 1320 Stony Brook Road, Stony Brook, NY 11790

Logo Some Z's, Geodesic, Inc., 401 Park Heights, 1091 Kumagawa, Fussa-shi, Tokyo, Japan

Logophile, College of Education, MacArthur Hall, Queen's University, Kingston, Ontario K7L 3N6

Logos, British Logo User Group, 33 Croft Gardens, Old Dalby, Melton Mowbray, Leicestershire LE14 3LE, England

The National Logo Exchange Newsletter, PO Box 5341, Charlottesville, VA 22905

Polyspiral, Boston Computer Society, Three Center Plaza, Boston, MA 02108

Schildpad Nieuws, Logo Centrum Nederland, Postbus 1408, 6501 BK Nijmegen, Netherlands

TI Source and Logo News, Microcomputer Corporation, 34 Maple Avenue, Armonk, NY 10504

Turtle News, Young Peoples' Logo Association, PO Box 855067, Richardson, TX 75085

TurtleTalk, 955 Greenwood Boulevard, Issaquah, WA 98027

### 2.2. Periodicals with regular Logo features

Closing the Gap: Microcomputers for the Handicapped, PO Box 68, Henderson, MN 56014, Logo column by Griff Wigley

COM3, PO Box 245, CEGV, Niddrie, Australia 3042, Logo column, "Turtle Talk," by Sandra Wills

COMPUTE!, PO Box 5406, Greensboro, NC 27403, Logo column, "Friends of the Turtle," by David Thronburg

The Computing Teacher, University of Oregon, 1787 Agate Street, Eugene, OR 97403, Logo column, "The Logo Center," by Kathleen Martin and Tim Riordon

Creative Computing, PO Box 789-M, Morristown, NJ 07960, Logo column, "Logo Ideas," by Robert Lawler

Hands On!, Technical Education Research Centers, 8 Eliot Street, Cambridge, MA 02138, Logo column by various staff members

Home Computer Magazine, 1500 Valley River Drive, Suite 250, Eugene, OR 97401, "Logo Times," a section of short Logo articles edited by Robert Ackerman

Micro Magazine, Shinkigensha Inc., Shinjuku-Youth Building 1-9, Shinjuku 4 Chome, Shinjuku-Ku, Tokyo 160, Japan

Softalk, 7250 Laurel Canyon Boulevard, North Hollywood, CA 91603, Logo column, "The Voice of the Turtle," by Donna Bearden

Teaching and Computers, Scholastic Inc., 730 Broadway, New York, NY 10017, Logo column, "Logo Notebook," by Tom Lough and Steve Tipps

Window, 469 Pleasant Street, Watertown, MA 02171, a "magazine" on a disk, with a Logo program each issue

99'er Home Computer Magazine, PO Box 5537, Eugene, OR 97405, out of print, Logo column by Henry Gorman

### 2.3. Periodicals with special Logo issues

Articles are also cited in the ARTICLES section.

Byte August 1982

* Abelson, Harold, "A Beginner's Guide to Logo"

* Bamberger, Jeanne, "Logo Music"

* Boecker, Heinz-Dieter, and Fischer, Gerhard, "Logo Project PROKOP"

* diSessa, Andrea A., and White, Barbara Y., "Learning Physics from a Dynaturtle"

* Goldenberg, E. Paul, "Logo - A Cultural Glossary"

* Gorman, Henry, Jr., "The Lamplighter Project"

* Harvey, Brian, "Why Logo?"

* Higginson, William, "Leading Fish to Water"

* Jewson, Jan, and Pea, Roy D., "Logo Research at Bank Street College"

* Lawler, Robert W., "Designing Computer Based Microworlds"

* Lemmons, Phil, "Logo Update"

* Leron, Uri, "The Group of the Turtle"

* Muller, Jim, "Young Peoples' Logo Association"

* Solomon, Cynthia, "Introducing Logo to Children"

* Watt, Dan, "Logo in the Schools"

* Williams, Gragg, "Logo for the Apple II, the TI-99/4A, and the TRS-80 Color Computer"

Classroom Computer News April 1983

* Bearden, Donna, and Muller, Jim, "Turtle Graphics: On and Off the Screen"

* Birch, Louisa, "The Turtle's Corner"

* Carter, Ricky, "The Complete Guide to Logo"

* Carter, Ricky, "Turtle Topics: Drawing Polygons"

* Lough, Tom, and Tipps, Steve, "Is There Logo after Turtle Graphics?"

* Moore, Mary Jo, "The Art of Teaching Logo"

* Olds, Henry F., Jr., "The Year of the Turtle"

* Steffin, Sherwin A., "The Instructional Applicability of Logo"

* Watt, Dan, "Learning with Logo"

The Computing Teacher November 1982

* Bandelier, Nellie, "TI Logo and First Graders - A Winning Combination"

* Billstein, Rick, "Learning Logo and Liking It"

* Bull, Glen, "Microworlds: Design of a Language"

* Coon, Merrianne, "Papert at the Faire"

* Lough, Tom, "WATERCROSS: A Logo Exploration"

* Martin, Kathleen, and Berner, Andrew, "Teaching Turtles"

* Martin, Kathleen, et al., "Turtle Graphics On and Off the Computer"

* Muller, Jim, "What Can the Computer and the YPLA Do for Handicapped Children?"

* Riordon, Tim, "Creating a Logo Environment"

* Upitis, Rena, "Logo and the Primary - Junior Pupil: One Student's First Encounter"

* Upitis, Rena, "Turtle Talk"

The Computing Teacher December 1983 / January 1984

* Bearden, Donna, "Optical Illusions - A challenge in Problem Solving"
* Billstein, Rick, and Moore, Margaret L., "Recursion, Recursion"
* Bull, Glen, and Tipps, Steve, "Problem Spaces in a Project Oriented Logo Environment"
* Clements, Douglas H., "Supporting Young Children's Logo Programming"
* Lough, Tom, "A Cure for Recursion"
* Lough, Tom, "Where is Logo?"
* Martin, Kathleen, and Riordon, Tim, "The Logo Center"
* McCauley, Jim, "Kepler"
* Moore, Margaret L., "A Recursion Excursion with a Surprising Discovery"
* Moursund, David, "Logo Frightens Me"
* Muller, Jim, "Dumping the Turtle"
* Muller, Jim, "No Threshold, No Ceiling ... Really!"
* Riordon, Tim, "Helping Students with Recursion: Teaching Strategies. Part I"
* Torgerson, Shirley, "Classroom Management for Logo"
* Torgerson, Shirley, et al., "Logo in the Classroom, Session 1"

### 2.4. Periodicals with occasional Logo articles

A +

Byte

Classroom Computer Learning

Educational Computer

Electronic Learning

inCider

Infoworld

Instructor

Microcomputing

Popular Computing

Teaching Learning Computing

## 3. ARTICLES

Abelson, Harold, "A Beginner's Guide to Logo," Byte, August 1982

Abbott, Hilton, "Three Key Logo," Classroom Computer News, May / June 1983

Ackerman, Robert, (ed.), "Logo Times," a section of short Logo articles in Home Computer Magazine

Ackerman, Robert, "What is Logo," Home Computer Magazine, Vol. 4 No. 1

Adams, Roe, "The New Shell Game," Softalk, July 1982

Adams, Tony, "Using Multiple Turtles to Create Games," COM3 magazine, November 1983

Allen, John, "An Overview of LISP," Byte, August 1979

Anderson, Judy, et al., "In-Service Workshop, Part VI, Programming in Logo," Electronic Learning, March 1984

Bamberger, Jeanne, "Logo Music," Byte, August 1982

Bandelier, Nellie, "TI Logo and First Graders - A Winning Combination," The Computing Teacher, November 1983

Barnes, B. J., and Hill, Shirley, "Should Young Children Work with Microcomputers - Logo before Lego(tm)?" The Computing Teacher, May 1983

Bearden, Donna, "Logo Type," regular column in Creative Computing magazine

Bearden, Donna, "Optical Illusions - A Challenge in Problem Solving," The Computing Teacher, December 1983 / January 1984

Bearden, Donna, "The Voice of the Turtle," regular column in Softalk magazine

Bearden, Donna, and Muller, Jim, "Logo Talks Back," inCider, September 1983

Bearden, Donna, and Muller, Jim, "Turtle Graphics: On and Off the Screen," Classroom Computer News, April 1983

Beasley, --., "Printing Logo Graphics," Creative Computing, October 1983

Berdonneau, Catherine, "But What's the Use of Asking Kids to Have the Computer to Draw a House?" SEEDBED, No. 12, Teachers Center Project, Southern Illinois University at Edwardsville, 1982

Berdonneau, Catherine, "Recueil des Pratiques Pedagogiques autor de Logo," INRP Agence de Informatique

Berdonneau, Catherine and Dumas, Rose-Marie, "Twelve Loves; Move," School Science and Mathematics, December 1983

Berry, Steven, "PRogressive Learning by COmputer," Personal Computing, December 1981

Bies, Richard, "Vectors in Logo: Of Stars and Sprites," 99'er Home Computer Magazine, Vol. 2, No 4 (February 1983)

Billstein, Rick, "Learning Logo and Liking It," The Computing Teacher, November 1983

Billstein, Rick., "Turtle Geography," Classroom Computer Learning, October 1983

Billstein, Rick, and Moore, Margaret L., "Recursion, Recursion," The Computing Teacher, December 1983 / January 1984

Birch, Louisa, "The Turtle's Corner," Classroom Computer News, April 1983

Birchall, Steve, "Logo: The Programmer Friendly Language," Softside, Vol. 6, No. 3

Boecker, Heinz-Dieter, and Fischer, Gerhard, "Logo Project PROKOP," Byte, August 1982

Bowen, Barbara, "Using Logo in the Classroom: The Project Approach," Teaching and Computers, Spring 1983

Bridger, Mark, "Logos for the IBM," Softalk for the IBM," Softalk for the IBM PC, April 1984

Brown, Steve, "Logo Column," regular column in SIG Bulletin

Bull, Glen, "Adventure Stories and English Skills," The National Logo Exchange Newsletter, January 1984

Bull, Glen, "Centering Titles," The National Logo Exchange Newsletter, March 1984

Bull, Glen, "Design of a Language," The National Logo Exchange Newsletter, September 1982

Bull, Glen, "Microworlds: Design of a Language," The Computing Teacher, November 1983

Bull, Glen, "ETV: Educational TurtleVentures Part I," The National Logo Exchange Newsletter, April 1984

Bull, Glen, "ETV: Educational TurtleVentures Part II," The National Logo Exchange Newsletter, May 1984

Bull, Glen, "How to Use Logo," The National Logo Exchange Newsletter, January 1983

Bull, Glen, "Is Logo a Tool?" The National Logo Exchange Newsletter, November 1982

Bull, Glen, "Introducing the Text Screen," The National Logo Exchange Newsletter February 1983

Bull, Glen, "Languages for Implementing Solutions and Languages for Finding Solutions," The National Logo Exchange Newsletter, December 1982

Bull, Glen, "Logo Characters," The National Logo Exchange Newsletter, February 1984

Bull, Glen, "Logo Grade Book, Part I: Grade Averager," The National Logo Exchange Newsletter, November 1983

Bull, Glen, "Logo Grade Book, Part II: Report Generator," The National Logo Exchange Newsletter, December 1983

Bull, Glen, "The Meaning of Meaning," The National Logo Exchange Newsletter, October 1982

Bull, Glen, "Numbers and Lists," The National Logo Exchange Newsletter, September 1983

Bull, Glen, "Teaching Grammar to a Computer," The National Logo Exchange Newsletter, April 1983

Bull, Glen, "Teaching Grammar to a Computer II," The National Logo Exchange Newsletter, May 1983

Bull, Glen, and Tipps, Steve, "Problem Spaces in a Project Oriented Logo Environment, Part I," The National Logo Exchange Newsletter, December 1983

Bull, Glen, and Tipps, Steve, "Problem Spaces in a Project Oriented Logo Environment, Part II," The National Logo Exchange Newsletter, January 1984

Bull, Glen, and Tipps, Steve, "Problem Spaces in a Project Oriented Logo Environment," The Computing Teacher, December 1983 / January 1984

Burke, Michael, "From Four Bugs to a Sociaty of Turtles," The National Logo Exchange Newsletter, October 1983

Burrowes, Sharon, "The Hazards of Hacking," The National Logo Exchange Newsletter, January 1984

Burrowes, Sharon, "Some Logo Drawing Ideas," The Computing Teacher, April 1983

Burrowes, Sharon, with Burrowes, David, "Ed Emberley and the Turtle," The National Logo Exchange Newsletter April 1983

Bussey, Jim, "Logo: A Language for Everyone," Commodore: The Microcomputer Magazine, April/May 1983

Cannara, A. B., "Toward a Human Computer Language," Creative Computing, May 1975

Cardwell, Jan, "Logo Stepping Stones for Young Minds," The National Logo Exchange Newsletter, November 1983

Carlin, James B., "The Turtle and the Children at Center Stage," Childhood Education, November / December 1981

Carlson, E. H., "Three Faces of Apple Logo," Micro - The 6502/6809 Journal No. 53, October 1982

Carter, Ricky, "The Complete Guide to Logo," Classroom Computer News, April 1983

Carter, Ricky, "Logo and the Great Debate," Kilobaud Microcomputing," September 1981

Carter, Ricky, "Turtle Topics: Drawing Polygons," Classroom Computer News, April 1983

Chace, Susan, "Technology: New Language Eases Writing of Microcomputer Programs," Wall Street Journal, August 27, 1982

Chadwick, Ian, "Logo's Lineage: The Canadian Connection," Antic, The Atari Resource Magazine

Chumley, James, "The Turtle as Tutor," PCjr Magazine, April 1984

Clarke, Valerie, "Children's Perception of Logo," COM3 magazine, August 1983

Clements, Douglas H., "Supporting Young Children's Logo Programming," The Computing Teacher, December 1983 / January 1984

Coon, Merrianne, "Papert at the Faire," The Computing Teacher, November 1983

Cron, Mary, "Trouble in Logoland," Teaching, Learning, Computing, December 1983

Daley, P. "TI Logo in the Schools," Micro - The 6502/6809 Journal, No 64, September 1983

D'Angelo, John, "Redefining the Learning Process with a Computer Language," Computer Graphics World, January 1982

David, Andrew, "NOVA's 'Talking Turtle,' A Compelling Documentary," The National Logo Exchange Newsletter, December 1983

diSessa, Andrea A., and White, Barbara Y., "Learning Physics from a Dynaturtle," Byte, August 1982

Dyrli, Odvard E. "Logo, A Language that Empowers Children," Learning, October 1983

Estes, Yvonne, "Logo and Trapped Intelligence," Computers in Mathematics and Science Teaching, Vol. 3 No. 2, 1983/84

Euchner, C., "Debate Grows on Logo's Effect on Children's Reasoning Skills," Education Week, November 23, 1983

Eyster, Richard H., "Seymour Papert and the Logo Universe," Creative Computing, December 1981

Fagan, Edward, "Writing / Reading: Logo's Syntonic Learning," The Computing Teacher, April 1983

Feurzeig, Wallace, and Lukas, George, "Logo - A Programming Language for Teaching Mathematics," Educational Technology, March 1972

Fiday, D. "Programming from Second Grade on .. and the Laraway Logo Experience," G/C/T, November / December 1983

Frank, Cathy, "Breaking Up Is Hard To Do," The National Logo Exchange Newsletter, April 1984

Friedland, Edward I., and Friedland, Mark, "The Future of Logo," TurtleTalk, Winter 1984

Fry, Jim, "How to Make Your Own Sprite Graphics," The Fry, Jim, "NLX ABC's Lead to List Processing Adventures," The National Logo Exchange Newsletter, December 1983

National Logo Exchange Newsletter March 1983

Fry, Jim, "Tessalate Your NLX ABC's," The National Logo Exchange Newsletter April 1983

Gervich, Shoneen, "Here Comes Atari Logo," The National Logo Exchange Newsletter, September 1983

Goldenberg, E. Paul, "Logo - A Cultural Glossary," Byte, August 1982

Goldberg, Kenneth P., "Different Versions of Logo," Family Computing Magazine, February 1984

Goodman, Danny, "Logo: A Language for Children of All Ages," PC Magazine, December 1982

Goodman, William, "Logo goes to Camp," Info Age, November 1982

Gorman, Henry, Jr., Avoiding Turtle Traps: Writing Better Logo," 99'er Home Computer Magazine, Vol. I, No. 5

Gorman, Henry, Jr., "The Basic Issue and the Tortoise's Retort," 99'er Home Computer Magazine, Vol. 2, No. 7, May 1983

Gorman, Henry, Jr., "Debugging in Logo," 99'er Home Computer Magazine, Vol 2, No. 3

Gorman, Henry, Jr., "The Lamplighter Project," Byte, August 1982

Gorman, Henry, Jr., "Logo Shoots for the Moon," Home Computer Magazine, Vol. 4 No. 1

Gorman, Henry, Jr., "Lyrical Logo," Home Computer Magazine, Vol. 4 No. 1

Gorman, Henry, Jr., and Bourne, Lyle E., Jr., "Learning to Think by Learning Logo: Rule Learning in Third Grade Computer Programmers," Bulletin of the Psychonomic Society, Vol. 21 No. 3 1983

Govier, H., "Language of Discovery: Logo," New York Times Educational Supplement, Ocotber 1, 1982

Grant, June, and Semmes, P., "A Rationale for Logo for Hearing Impaired Preschoolers," American Annals of the Deaf, Vol. 128 No. 5, 1983

Greth, Vidal, "Learning to Learn with the Father of Logo," Atari Connection, Fall 1983

Gwyn, R., "Logo Rhythms," New York Times Educational Supplement, March 4, 1983

Harris, Ross J., "An A-Mazing Logo Experiment," Hands On!, Summer 1983

Harvey, Brian, "Logo to Epson Screen Dump," Atari Teachers' Network Newsletter, January 1984

Harvey, Brian, "Why Logo?" Byte, August 1982

Harvey, Wayne, "Which Programming Language is Right for You?" Classroom COmputer Learning, April / May 1984

Hawkins, J. et al., "Microcomputers in Schools: Impact on the Social Life of Elementary Classrooms," Journal of Applied Developmental Psychology, Vol. 3, 1982

Hayes, Brian, "Computer Recreations: Turning Turtle Gives One a View of Geometry from the Inside Out," Scientific American, February 1983

Heller, Dorothy, "Young Peoples' Logo Association Promotes Learning," Infoworld, December 1981

Higginson, William, "Leading Fish to Water," Byte, August 1982

Jacquet, Francois et al., "Une Approche de Logo," available from Institut Romand de Recherches et de Documentation Pedagogiques, Faubourg de l'Hopital 43, 2000 Neuchatel, Switzerland

Jewson, Jan, and Pea, Roy D., "Logo Research at Bank Street College," Byte, August 1982

Judd, Dorothy H., "Programming: An Experience in Creating a Useful Program or an Experience in Computer Control," Educational Computer, March/Apri1983

Kellam-Scott, Barbara, "Papert Reflects on Logo, Past, and Future," Teaching Learning Computing, November 1983

Kellam-Scott, Barbara, "Papert Reflects on Logo, Past, and Present," TurtleTalk, Winter 1984

Kelman, Peter, "Journey through mathland: An Interview with Seymour Papert," Classroom Computer News, March/April 1981

Kelman, Peter, "Seymour, Has Your Dream Come True?" Classroom Computer News, April 1983

Kildall, Gary, and Thornburg, David, "Digital Research's DR Logo," Byte, June 1983

Kirchner, Roger B., "Problem Solving with Logo," 99'er Home Computer Magazine, Vol. 1 No. 5

Kirchner, Roger B., "Daisies: Math Flowers with Logo," 99'er Home Computer Magazine, Vol. 2, No. 4, February 1983

Kleiman, Glenn, "Teaching Johnny How to Program," Compute!, July 1982

Kirchner, Roger B., "The Fifteen Puzzle," 99'er Computer Magazine, Vol. 2, No. 6, April 1983

Kramer, Sharon, "Word Processing in a Logo Environment," Electronic Learning, March 1984

Kuchinskas, G., "Developing Interactive Language Experiences for the Computer," The Computing Teacher, June 1982

Lamster, Jane and Lamster, Hal, "The Doctor is Indisposed," PC Magazine, May 15, 1984

Lamster, Jane and Lamster, Hal, "Logo: A Language for Grownups, too," PC Magazine, May 15, 1984

Lamster, Jane and Lamster, Hal, "A Logo With The IBM Label," PC Magazine, May 15, 1984

Lawler, Robert W., "Designing Computer Based Microworlds," Byte, August 1982

Lawler, Robert, "Logo Ideas," regular column in Creative Computing, 1982 - 1983

Lemmons, Phil, "Logo Update," Byte, August 1982

Leron, Uri, "The Group of the Turtle," Byte, August 1982

Lohrman, Larry, "Turtle Graphics Tutorial, Part 1," TurtleTalk, June 1983

Lohrman, Larry, "Turtle Graphics Tutorial, Part 2," TurtleTalk, Winter 1984

Lough, Tom, "Can the Turtle Draw a Sine Wave?" The National Logo Exchange Newsletter January 1982

Lough, Tom, "A Cure for Recursion," The Computing Teacher, December 1983 / January 1984

Lough, Tom, "Exploring New Horizons with Logo: How One School is Charting the Course. Part II," Electronic Learning, April 1983

Lough, Tom, "Fractal Fun with Logo," The National Logo Exchange Newsletter, October 1982

Lough, Tom, "Give Me a Sine: Lissajous Logo," National Logo Exchange Newsletter, May 1984

Lough, Tom, "Logo: Discovery Learning with the Classroom's Newest Pet. Part I," Electronic Learning, March 1983

Lough, Tom, "Logo on Wheels," The National Logo Exchange Newsletter, November 1982

Lough, Tom, "Logo Quilting Party," The National Logo Exchange Newsletter, December 1982

Lough, Tom, "Logo Your Boat," The National Logo Exchange Newsletter, September 1982

Lough, Tom, "ParaboLogo," The National Logo Exchange Newsletter, April 1984

Lough, Tom, "Save Your Apple Logo Graphics," The National Logo Exchange Newsletter, May 1983

Lough, Tom, "Slow Turtle Moves Clearly," The National Logo Exchange Newsletter, September 1983

Lough, Tom, "WATERCROSS: A Logo Exploration," The Computing Teacher, November 1983

Lough, Tom, "Where is Logo?," The Computing Teacher, December 1983 / January 1984

Lough, Tom, and Tipps, Steve, "Is There Logo after Turtle Graphics?" Classroom Computer News, April 1983

Lough, Tom, and Tipps, Steve, "Logo Notebook," regular column in Teaching and Computers

Lowd, Beth, "Hints on Introducing Logo in the Classroom," Classroom Computer News, March 1983

Mace, Scott, "Papert Keynotes Computer Faire," Infoworld, March 1982

Mace, Scott, "Where is Logo Taking our Kids?" Infoworld, January 23, 1984

Makins, V., "Logo Man," New York Times Educational Supplement, June 4, 1982

Makins, V., "Screen Sprites," New York Times Educational Supplement, November 12, 1982

Malmberg, D., "Turtle Graphics for the VIC-20 and C64," Micro - The 6502/6809 Journal, No 64, September 1983

Markoff, John, "Logo overturns Old Computer Education Models," Infoworld, December 1981

Markuson, Carolyn et al., "Logo Fever: The Computer Language Every School is Catching," Instructor, January 1983

Markuson, Carolyn et al., "Logo Fever: The Computer Language Every School is Catching," Arithmetic Teacher, September 1983

Martin, Kathleen, et al., "Turtle Graphics On and Off the Computer," The Computing Teacher, November 1983

Martin, Kathleen, and Berner, Andrew, "Teaching Turtles," The Computing Teacher, November 1983

Martin, Kathleen, and Riordon, Tim, "The Logo Center," regular column in The Computing Teacher

Mason, Grace, "Widening Learning Horizons with TI Logo," Educational Computer, January / February 1982

Mass, Lynne, Students Write the Book on Logo," The National Logo Exchange Newsletter, December 1983

McCauley, Jim, "Kepler," The Computing Teacher, December 1983 / January 1984

McLean and Pagano, "Turtle Geometry without Hardware," Creative Computing, May 1975

McLeod, J., "Take the Turtle to School: A Child's View of Programming," Classroom Computer Learning, November / December 1983

McLees, Jock, "Where Do Turtles Come From?" Kilobaud Microcomputing, March 1982

Min-Chih, Earl, "Logo for the IBM PC, Part 1: Waterloo Logo," TurtleTalk, Winter 1984

Moore, Margaret L., "A Recursion Excursion with a Surprising Discovery," The Computing Teacher, December 1983 / January 1984

Moore, Mary Jo, "The Art of Teaching Logo, or When and When Not to Bother the Learner," Hands On!, Spring 1983

Moore, Mary Jo, "The Art of Teaching Logo," Classroom Computer News, April 1983

Moursund, David, "Logo Frightens Me," The Computing Teacher, December 1983 / January 1984

Menconi, Dave, and Richards, Ted, "Logo vs. BASIC: Which Language is BEST?" Atari Connection, Fall 1983

Muller, Jim, "Dumping the Turtle," The Computing Teacher, December 1983 / January 1984

Muller, Jim, "The Friendly Languages," Creative Computing, October 1981

Muller, Jim, "Logo, for the Imagination, Infoworld, March 1982

Muller, Jim, "Logo, The Voice of the Turtle," Softalk, July 1982

Muller, Jim, "The Million Dollar Smile," The Computing Teacher, February 1983

Muller, Jim, "No Threshold, No Ceiling ... Really!" The Computing Teacher, December 1983 / January 1984

Muller, Jim, "What Can the Computer and the YPLA Do for Handicapped Children?" The Computing Teacher, November 1983

Muller, Jim, "Young Peoples' Logo Association," Byte, August 1982

Muir, Robs, "NLXual Challenges," a monthly column, The National Logo Exchange Newsletter

Nelson, Harold, "Learning with Logo," onComputing, Summer 1981

Nelson, Harold, "Logo for Personal Computers," Byte, June 1981

Nelson, Harold, "Logo: Not Just for Kids," Microcomputing, Vol 6 No 3, March 1982

Nelson, Harold, and Friedman, Rich, "Seymour Papert: Spearheading the Computer Revolution," onComputing, Summer 1981

Nevile, Liddy, "Of Crab Canons and Turtle Hums," COM3 Magazine, March / April 1983

Nix, Linda, "Logo and the Single Computer: How to Cope," The National Logo Exchange Newsletter February 1983

Nix, Linda, "Logo Reinforces Geometric TABS Skills," The National Logo Exchange Newsletter, September 1982

Olds, Henry, "Through a New Looking Glass," Kilobaud Microcomputing, SEptember 1981

Olds, Henry F., Jr., "The Year of the Turtle," Classroom Computer News, April 1983

Pantiel, Mindy, and Peterson, Becky, "Learning Logo is a Family Affair," Family Computing Magazine, February 1984

Papert, Seymour, "And a Little Child Shall Lead Them," Instructional Innovator, February 1982

Papert, Seymour, "A Computer Laboratory for Elementary Schools," Computers and Automation, June 1972

Papert, Seymour, "Computers and Computer Cultures," Creative Computing, March 1981

Papert, Seymour, "Micros Becoming A Social Force," Infoworld, April 12, 1982

Papert, Seymour, "New Cultures from New Technologies," Byte, September 1981

Papert, Seymour, "Society Will Balk, but the Future May Demand a Computer for Each Child," Electronic Education, September 1981

Papert, Seymour, "Teaching Children Thinking," Programmed Learning and Educational Technology, Vol. 9, 1972

Papert, Seymour, "Teaching Children to be Mathematicians versus Teaching About Mathematics," International Journal for Mathematical Education, Science, and Technology, Vol. 3, 1972

Papert, Seymour, and Solomon, Cynthia, "Twenty Things to Do with a Computer," Educational Technology, April 1972

Pearce, Elizabeth, "Teach Your Turtle to Draw Hex Signs," The National Logo Exchange Newsletter May 1983

Peterson, Marty, "MIT Logo: Educational Program for the Apple," Infoworld, January 31, 1983

Poole, A., "Turtle Pilot: Part I," COMPUTE!, September 1982

Poole, A., "Turtle Pilot: Part II," COMPUTE!, October 1982

Poole, A., "Turtle Pilot: Part III," COMPUTE!, November 1982

Raines, J. R., "Introduction to Turtle Graphics in Apple Pascal," Micro the 6502/6809 Journal, No 53, October 1982

Rainey, Joan, "Concrete Turtling," The National Logo Exchange Newsletter, February 1984

Raleigh, Lisa, "Logo Isn't Just for Kids Anymore," ISO World, November 14, 1983

Raskin, B., "Logo for the Reluctant Learner - Me," Learning, October 1983

Razzano, Linda, "Children and the Turtle Down on the Farm," The National Logo Exchange Newsletter May 1983

Reggini, Horacio, "A Revision of Learning and Teaching," Revista del Instituto de Investigaciones Educativas, No. 43, November 1983

Reggini, Horacio, "Logo and Von Neuman's Ideas," Tecnologia Informatica, Ano 2, No. 15

Reggini, Horacio, "Logo: An Innovation in the Field of Computers," Mundo Informatico, Vol. 4, No. 74, August 1983

Reilly, Pat, "Cherokee Indian Symbols Come Alive with Logo Sprites," The National Logo Exchange Newsletter May 1983

Rifkin, Bonnie, "Turtle Folders Help Third Graders," The National Logo Exchange Newsletter, January 1983

Rifkin, Bonnie, "Where Does Logo Fit In?" The National Logo Exchange Newsletter February 1983

Riordon, Tim, "Creating a Logo Environment, Part I," The National Logo Exchange Newsletter March 1983

Riordon, Tim, "Creating a Logo Environment, Part II," The National Logo Exchange Newsletter April 1983

Riordon, Tim, "Creating a Logo Environment," The Computing Teacher, November 1983

Riordon, Tim, "Helping Students with Recursion: Teaching Strategies. Part I," The Computing Teacher, December 1983 / January 1984

Riordon, Tim, "Helping Students with Recursion: Teaching Strategies. Part II: Moving to the Computer," The Computing Teacher, February 1984

Riordon, Tim, "Helping Students with Recursion: Teaching Strategies. Part III: Teaching Students about Embedded Recursion," The Computing Teacher, March 1984

Rosch, Winn L., "Logo: The Advanced Course," PCjr Magazine, April 1984

Rosch, Winn L., "The Voice of the Turtle," regular column in PCjr Magazine

Rosch, Winn L., "A Tale of Two Loops," PC Magazine, May 15, 1984

Rosenthal, Steve, "Follow the Turtle," A+ Magazine, December 1983

Rousseau, J., and Smith, S., "Whither Goes the Turtle?" Kilobaud Microcomputing, September 1981

Rowe, Neil, "Grammar as a Programming Language," Creative Computing, January 1978

Rowe, Neil, "Sine POLYs: Some Geometrical Explorations," Creative Computing, December 1979

Rubenstein, Richard, "Using Logo in Teaching," Topics in Instructional Computing, January 1975

Russell, Susan Jo, "Had We but World Enough and Time: Logo in Special Education," Classroom Computer Learning, October 1983

Sandberg-Diment, Eric, "Logo, the Friendliest Language," Science Digest, November 1983

Sandler, Corey, "Day of the Turtle," PCjr Magazine, April 1984

Schrage, M., "BASIC to Logo: Computer Talk," Rolling Stone, April 14, 1983

Shapiro, N., "Now Your Kids Can Teach You to Compute," Popular Mechanics, September 1982

Sheffer, Nola, "Boston Logo Users' Group News: diSessa and the Dynatürtle," The National Logo Exchange Newsletter, April 1984

Sheffer, Nola, "Papert Addresses Boston Logo Users' Group," The National Logo Exchange Newsletter, March 1984

Sheffer, Nola, "Reflections on a Logo Project in Cambridge," Hands On!, Spring 1982

Six Lamplighter Teachers, "Learning with Logo at the Lamplighter School," Kilobaud Microcomputing, September 1981

Solomon, Cynthia, "Introducing Logo to Children," Byte, August 1982

Sopp, Nancy Parks, "Logo Mess Ups: Process vs. Product," The National Logo Exchange Newsletter, October 1983

Staples, Betsy, "Turtles and Sprites for the 99/4," Creative Computing, December 1981

Stavely, Tony, "List Processing in Logo," The Computing Teacher, December 1982

Strausberg, Robin, "Helping the Girls Catch Up," Teaching, Learning, Computing, March 1984

Steffin, Sherwin, "A Challenge to Seymour Papert," Educational Computer, July / August 1982

Steffin, Sherwin A., "The Instructional Applicability of Logo," Classroom Computer News, April 1983

Stone, Greg, "Of Cats and Turtles in Mathland," inCider, June 1983

Stone, Greg, "Solving Problems with Logo," inCider, April 1983

Stone, Greg, "Solving Problems with Logo, Part II," inCider, May 1983

Streibel, M., "The Educational Utility of Logo," School Science and Mathematics, October 1983

Sugarman, Jay, "Brookline Students Hunt Logo Bugs," The National Logo Exchange Newsletter, December 1982

Sullivan, Nick, "The Man Behind Logo," Family Computing Magazine, Fabruary 1984

Swaine, Michael, "Young People Race to the Future on the Shoulders of Turtles," Infoworld, November 1981

Sweetnam, G. K., "Turtle Talk (Logo for Children)," Science Digest, November 1982

Swett, Sheila, "Logo Offspring: A Look at Modified Versions of Logo and Turtle Graphics Programs," Electronic Learning, May/June 1983

Tepper, Lois, "The Turtle Goes to College," The National Logo Exchange Newsletter, March 1984

Thomas, D. "Learning the Joys of Logo Language," MacLeans, September 6, 1982

Thornburg, David, "Friends of the Turtle," regular column in Compute! magazine

Thornburg, David, "Logo and Pilot Languages," Compute!, magazine, March 1981

Thorne, M., "Modelling Work," New York Times Educational Supplement, May 27, 1983

Tinker, Robert, "Logo's Limits: Or Which Language Would We Teach?" Hands On!, Spring 1983

Tipps, Steve, "Borders and Frames," The National Logo Exchange Newsletter, September 1983

Tipps, Steve, "Both Sides Now," The National Logo Exchange Newsletter, November 1983

Tipps, Steve, "Commodore Logo: Let's Take a Look," The National Logo Exchange Newsletter, November 1983

Tipps, Steve, "Distance and Direction," The National Logo Exchange Newsletter, September 1982

Tipps, Steve, "From Turtling to Programming," The National Logo Exchange Newsletter, January 1983

Tipps, Steve, "Getting Started," The National Logo Exchange Newsletter, December 1982

Tipps, Steve, "How Now, Output?" The National Logo Exchange Newsletter, May 1984

Tipps, Steve, "The Issue of Instant," The National Logo Exchange Newsletter, December 1983

Tipps, Steve, "Logo for All," The National Logo Exchange Newsletter, May 1983

Tipps, Steve, "The Other Side of Logo," The National Logo Exchange Newsletter, February 1983

Tipps, Steve, "Patterns and Repetition," The National Logo Exchange Newsletter, October 1982

Tipps, Steve, "Patterns and Repetition II," The National Logo Exchange Newsletter, November 1982

Tipps, Steve, "Random Thoughts," The National Logo Exchange Newsletter, March 1983

Tipps, Steve, "Right Here in Turtle City," The National Logo Exchange Newsletter, Fabruary 1984

Tipps, Steve, "Sweet P is a Sweetheart," The National Logo Exchange Newsletter, March 1984

Tipps, Steve, " Triangle Thinking :SIDE1," The National Logo Exchange Newsletter, January 1984

Tipps, Steve, " Triangle Thinking :SIDE2," The National Logo Exchange Newsletter, February 1984

Tipps, Steve, " Triangle Thinking :SIDE3," The National Logo Exchange Newsletter, March 1984

Tipps, Steve, "The Truth about Numbers," The National Logo Exchange Newsletter, October 1983

Tipps, Steve, "Unsquare Square: The Story of Scrunch," The National Logo Exchange Newsletter, April 1984

Tipps, Steve, "Variables: More Than One Way to Tame a Turtle," The National Logo Exchange Newsletter, April 1983

Toberman, Sandy, "A Parent Looks at Logo," The National Logo Exchange Newsletter, October 1982

Todd, Nancy, "Hello Logo," Educational Computer, March/April 1983

Torgerson, Shirley, "Classroom Management for Logo," The Computing Teacher, December 1983 / January 1984

Torgerson, Shirley, et al., "Logo in the Classroom, Session 1," The Computing Teacher, December 1983 / January 1984

Upitis, Rena, "Logo and the Primary - Junior Pupil: One Student's First Encounter," The Computing Teacher, November 1983

Upitis, Rena, "Turtle Talk," The Computing Teacher, November 1983

Vidya, S., "Making Logo Accessible to Preschool Children," Educational Technology, (in press)

Vidya, S., "Using Logo to Stimulate Children's Fantasy," Educational Technology, December 1983

Voderberg, Hank, "Use Logo Graphics in Your BASIC Program," The Computing Teacher, March 1984

Watt, Dan, "A Comparison of the Problem Solving Styles of Two Students Learning Logo: A Computer Language for Children," Creative Computing, December 1979

Watt, Dan, "Learning with Logo," Classroom Computer News, April 1983

Watt, Dan, "Logo: What Makes It Exciting?", Popular Computing, Vol 2 No 10, August 1983

Watt, Dan, "Logo in the Schools," Byte, August 1982

Watt, Dan, "Should Children be Computer Programmers?" Popular Computing, September 1982

Watt, Dan, "Teaching Turtles: Logo as an Environment for Learning," Popular Computing, July 1982

Watt, Molly, "Logo Building Blocks," inCider, January 1984

Watt, Molly, "De-Bug Collection," inCider, February 1984

Watt, Molly, "Logo: Where's the Pony?," inCider, March 1984

Watt, Molly, "What is Logo?", Creative Computing, October 1982

Weinreb, William, "Problem Solving with Logo: Using Turtle Graphics to Redraw a Design," Byte, November 1982

Weintraub, Hillel, "Logo for the Sony," The National Logo Exchange Newsletter, April 1984

Weir, Sylvia, "Logo: A Learning Environment for the Severely Handicapped," Journal of Special Education Technology, Vol. 5 No. 1, Winter 1982

Weir, Sylvia, et al., "Logo: An Approach to Educating Disabled Children," Byte, September 1982

Weir, Sylvia, "Logo: An Information Prosthetic for Communication and Control," Proceedings of the International Joint Conference on Artificial Intelligence, 1981

Weir, Sylvia, "Logo and the Exceptional Child," Kilobaud Microcomputing, May 1981

Weir, Sylvia, and Watt, Dan, "Logo: A Computer Environment for Learning Disabled Students," The Computing Teacher, January 1981

Weston, D., "Hi-Res Characters for Logo," Micro - The 6502/6809 Journal, No 64, September 1983

Wierzbicki, Barbara, "The Success of Logo Hinges on Teacher Training," Infoworld, January 23, 1984

Wigley, Griff, "Apple Sprite Logo," The National Logo Exchange Newsletter, April 1984

Wigley, Griff, "Logo," a monthly column, Closing the Gap

Wigley, Griff, "NLXionary: A Lectionary of Selected Logo Readings," a monthly column, The National Logo Exchange Newsletter

Williams, Gregg, "Logo for the Apple II, the TI- 99/4A, and the TRS-80 Color Computer," Byte, August 1982

Wills, Sandra, "Turtle Talk," regular column in COM3 Magazine

Wise, Deborah, "1982 to be the Year of the Turtle?" Infoworld, January 1982

Yee, Duane, "Of Sushi and Mushi," The National Logo Exchange Newsletter, January 1984

Young, George P., "Children, Computers, and Logo," Monitor, July / August 1983

Young, Kendall, and Bull, Glen, "Speaking of Logo, Now Tracy Can," The National Logo Exchange Newsletter, May 1984

Zemke, Suzanne, "Microcomputers and Education: Choices and Consequences," Educational Computer, March/April 1983

*Author unknown*, "A Home Computer Primer," Natural History, November 1982

*Author unknown*, "Apple Logo Spoken Here," Microcomputing, Vol 6 No 6, June 1982

*Author unknown*, "Computers are Objects to Think with: An Interview with Seymour Papert," Instructor, March 1981

*Author unknown*, "It's 9 A. M. - Do You Know Where Your Turtles Are?" TERC Staff, Hands On!, Spring 1982

*Author unknown*, "Learning with Computers: Are the Schools Ready for the Challenge?" Kilobaud Microcomputing, September 1981

*Author unknown*, "Logo for Personal Computers," Byte, June 1981

*Author unknown*, "Logo: Pratique Active de l'Informatique par l'Enfant," Recherches Pedagogiques, No. 111 (INRP SEVPEN), 1981

*Author unknown*, "Pilot's Turtle Graphics for Atari," Recreational Computing, May / June 1981

*Author unknown*, "Progressive Learning by Computer," Personal Computing, December 1981

*Author unknown*, "Teaching the Turtle New Tricks: Logo Language Used to Instruct School Children," Time, October 11, 1982

*Author unknown*, "The Computer: A Learning Friend," Early Years, 1983

*Author unknown*, "TI Logo," Creative Computing, December 1981

*Author unknown*, "Trapped Intelligence," Science, July / August 1980

*Author unknown*, "What is the Logo Language?" Consumer Reports, November 1983

## 4. RESEARCH GROUP PUBLICATIONS

Massachusetts Institute of Technology

The Educational Computing Group at MIT has over sixty Memos on Logo as it was developed through the 1970s. Of particular interest is "The Final Report of the Brookline Project, Parts II and III," by Dan Watt, Seymour Papert, Andrea diSessa, and Sylvia Weir, September 1979 (Logo Memos 53 and 54). For a bibliography of all the MIT Logo Memos, send $1.00 (checks payable to M.I.T.) to Logo Memo Bibliography, Educational Computing Group, MIT, 545 Technology Square, Cambridge, MA 02139.

National Science Foundation

A series of reports is available from the National Science Foundation. Many of these reports are also published at the Massachusetts Institute of Technology as Logo Memos.

Abelson, Harold, and Adams, Jim, A Glossary of Logo Primitives

Abelson, Harold, et al., Logo Progress Report 1973-1975

Austin, Howard, Teaching Teachers Logo, The Lesley Experiments

Bamberger, Jeanne, Development of Musical Intelligence 1: Strategies for Representing Simple Rhythms

Bamberger, Jeanne, The Luxury of Necessity

diSessa, Andy, The Turtle Escapes the Plane: Some Advanced Turtle Geometry

Feurzeig, Wallace, et al., An Introductory Logo Teaching Sequence: Logo Teaching Sequence on Logic, Logo Reference Manual

Feurzeig, Wallace, et al., Programming Languages as a Conceptual Framework for Teaching Mathematics. Final Report on the First Fifteen Months of the Logo Project

Feurzeig, Wallace, et al., The Logo-S Language and the Portable Logo System. Volume I: Language and System Descriptions. Volume II: Program Listings

Goldenberg, E. Paul, A Glossary of PDP11 Logo Primitives

Goldstein, Ira, et al., Llogo: An Implementation of Logo in LISP

Grant, Richard, et al., Logo Teaching Sequences on Numbers and Functions and Equations. Teacher's Text and Problems

Lieberman, Henry, The TV Turtle: A Logo Graphics System for Raster Displays

Lukas, George, et al., Logo Teaching Sequences on Strategy in Problem Solving and Story Problems in Algebra. Teacher's Text and Problems

Papert, Seymour, et al., Final Report of the Brookline Project, Part II: Project Summary and Data Analysis

Papert, Seymour, et al., Interim Report of the Logo Project in the Brookline Public Schools: An Assessment and Documentation of a Children's Computer Laboratory

Perlman, Radia, TORTIS (Toddler's Own Recursive Turtle Interpreter System)

Solomon, Cynthia, and Papert, Seymour, A Case Study of a Young Child Doing Turtle Graphics in Logo

Watt, Dan, Final Report of the Brookline Logo Project, Part III: Profiles of Individual Students' Work

Weiner, Walter B., et al., The Logo Processor: A Guide for System Programmers

*Author unknown*, Turtle Geometry, Teacher's Guide

*Author unknown*, Turtle Symmetry, Teacher's Guide

Bank Street College of Education, Center for Children and Technology

Recent research at the Bank Street College of Education has produced a number of technical reports, many of which are related to Logo. To obtain acquisition information, write L. Bryant, Bank Street College of Education, Center for Children and Technology, 610 West 112th Street, New York, NY 10025. (212) 663-7200

Hawkins, Jan, Learning Logo Together: The Social Context, Technical Report #13, April 1983

Kurland, D. Midian, and Pea, Roy., D., Children's Mental Models of Recursive Loop Program, Techincal Report #10, February 1983

Pea, Roy D., Logo Programming and Problem Solving, Technical Report #12, April 1983

Pea,Roy D., and Kurland, D. Midian, Logo Programming and the Devleopment of Planning Skills, Techinical Report #16, December 1983

Pea, Roy D., et al., Developmental Studies on Learning Logo Computer Programming, summarized in Abstracts from the Biennial Meeting of the Society for Research in Child Development, Vol. 4, April 1983

*Author Unknown*, Structured Interviews on Children's Conceptions of Computers, Techinical Report #19, December 1983

University of Edinburgh

The Department of Artificial Intelligence of the University of Edinburgh has produced over 100 publications documenting research on the use of computers and Logo in education, from 1970 to the present. For a complete listing of these publications, write Department of Artificial Intelligence, University of Edinburgh, Forest Hill, Edinburgh EH1 2QL, Scotland.

Chiltern Logo Project

The Advisory Unit for Computer Based Education has produced a series of articles and reports on their Logo research project. For more information, write Chiltern Logo Project, Endymion Road, Hatfield, Hertfordshire, AL10 8AU, United Kingdom.

## 5. KNOWN DISSERTATIONS AND THESES REPORTING LOGO RELATED RESEARCH

Austin, Howard, "A Computational Theory of Physical Skill," unpublished doctoral thesis, Massachusetts Institute of Technology, 1976

Evans, Harold Ray, "Some Effects of Logo Programming Instruction with Fourth Grade Children," University of Virginia, 1984

Folk, Michael, "Influence of Development Level on a Child's Ability to Learn Concepts of Computer Programming." unpublished doctoral dissertation, Syracuse University, 1972, Dissertation Abstracts, 34/03 A p. 1125

Keiser, Linda, "Logo Software Package for Elementary School Teachers," unpublished bachelors thesis, University of Virginia School of Engineering, 1983

Lawler, Robert W., "One Child's Learning: An Intimate Study," unpublished doctoral dissertation, Massachusetts Institute of Technology, 1979

Milner, Stuart Dennis, "The Effects of Teaching Computer Programming on Performance in Mathematics," unpublished doctoral dissertation, University of Pittsburgh, 1972, Dissertation Abstracts, 38/08 A p. 4183

Milojkovic, James D., "Children Learning Computer Programming: Cognitive and Motivational Consequences," unpublished doctoral dissertation, Stanford University Psychology Department, December 1983

Rampy, Leah, "The Programming Styles of Fifth Grade Students: An Exploratory Study Using Logo," unpublished doctoral dissertation, University of Indiana, 1983

Reiber, Lloyd Paul, "The Effect of Logo on Increasing Systematic and Procedural Thinking According to Piaget's Theory of Intellectual Development and on its Ability to Teach Geometric Concepts to Young Children," unpublished masters thesis, University of New Mexico, 1983

Seidman, Robert Howard, "The Effects of Learning the Logo Computer Programming Language on Conditional Reasoning in School Children," unpublished doctoral dissertation, Syracuse University, 1980, Dissertation Abstracts, 41/06 B p. 2249

Statz, Joyce A., "The Development of Computer Programming Concepts and Problem Solving Abilities among Ten Year Olds Learning Logo," unpublished doctoral dissertation, Syracuse University, 1973, Dissertation Abstracts number 34/11 B p. 5418

Valente, Jose Armando, "Creating a Computer-based Learning Environment for Physically Handicapped Children," unpublished doctoral dissertation, Massachusetts Institute of Technology, MIT/LCS/TR-301,1983

Young, Lucy Mirisciotti, "An Analysis of the Effect of the Logo Computer Programming Environment upon the Reflective and Impulsive Cognitive Styles of Second Grade Students," unpublished doctoral dissertation, University of Pittsburgh, 1982

## 6. LOGO AIDS

Bailey, Harold et al., Aple Logo: Activities for Turtle Graphics, R.J. Bradley Co.

Beairsto, Bruce et al., Problem Solving with Logo: A Computer Unit Suitable for Grades 6 - 9, British Columbia Provincial Educational Media Center

Blitman, Elaine, and Jamile, Barbara, Logocards, Scott, Foresman and Company

Cron, Mary, Logo Guided Discovery Kits, Microcomputer Resources

Davis, Joan, and Schenker, Joan, Logo: A Problem Solving Approach, Turtle Enterprises

Davis, Joan, and Schenker, Joan, Logo: Problem Solving for Primary, Turtle Enterprises

Torgerson, Shirley, "Logo Lessons: Ideas for the Classroom," International Council for Computers in Education

Zinn, Karl, Exploring Logo: Apple, Sunburst Communications

Zinn, Karl, Exploring Logo: Commodore, Sunburst Communications

*Author unknown*, Apple Logo in the Classroom, MECC

*Author unknown*, EZ Logo, MECC

*Author unknown*, Getting Started with Logo, Developmental Learning Materials

*Author unknown*, Introduction to Logo for Teachers, MECC

*Author unknown*, Logo Reference Flip Chart, Scott, Foresman and Company

*Author unknown*, Logo Task Cards, Computer Skill Builders

*Author unknown*, Sketch a Picture, Houghton Mifflin Canada

*Author unknown*, Visual Masters for Teaching Logo, Computer Skill Builders

## 7. LOGO GROUPS

Asociacion Amigos de Logo, Salguero 2969, 1425 Buenos Aires, Argentina

Boston Computer Society, Logo Users Group, Three Center Plaza, Boston, MA 02108

British Logo User Group, 33 Croft Gardens, Old Dalby, Melton Mowbray, Leicestershire LE14 3LE, England

Friends of LISP, Logo, and Kids, PO Box 22094, San Francisco, CA 94122

Friends of the Turtle, PO Box 1317, Los Altos, CA 94022

Groupe de Reflexion en Didactique des Mathematiques de la Societe de Neuchateloise des Maiters de Mathematique, de Physique et de Chemie

Japanese Logo Research Group, Hiroyoshi Goto, Uny Co, BYNAS Div., 3F Dainagoya Building, 28-12 3Chome Meieki, Nakamura- ku, Nagoya, Japan 450

Logo Centrum Nederland, Postbus 1408, 6501 BK Nijmegen, Netherlands

National Logo Exchange, PO Box 5341, Charlottesville, VA 22905

Young Peoples' Logo Association, PO Box 855067, Richardson, TX 75085

## 8. PUBLISHERS' ADDRESSES

Addison Wesley, 2725 Sand Hill Road, Menlo Park, CA 94025

Alfred Publishing Co., Inc., 15335 Morrison St., PO Box 5964, Sherman Oaks, CA 91413

Basic Books, 10 East 53rd Street, New York, NY 10022

British Columbia Provincial Educational Media Center, 7351 Elmridge Way, Richmond, British Columbia, Canada V6X 1B8

Brooks / Cole Publishing Company, Monterey, CA 93940

Computer Skill Builders, 3130 N. Dodge Blvd., PO Box 42050, Tucson, AZ 85733

Creative Computing Press, 39 EAst Hanover Avenue, Morris Plains, NJ 07950

Creative Publications, 3977 East Bayshore Road, PO Box 10328, Palo Alto, CA 94303

Developmental Learning Materials, 1 DLM Park, PO Box 4000, Allen, TX 75002

EduComp Publications, 14242 Wyeth Avenue, Irvine, CA 92714

Emece Distribuidora, Alsina 2062, 1090 Buenos Aires, Argentina

Entelek, Ward Whidden House, The Hill, PO Box 1303, Portsmouth, NH 03801

Eyrolles, 61 Boulevard Saint-Germain, 75005 Paris, France

Geodesic Inc., 401 Park Heights, 1091 Kumagawa, Fussa-shi, Tokyo, Japan

Harper and Row Publishers, Inc., 10 East 53rd Street, New York, NY 10022

Hayden Book Co. Inc., 50 Essex Street, Rochelle Park, NJ 07662

Holt, Rinehart and Winston, 383 Madison Avenue, New York, NY 10017

Houghton Mifflin Co., 1 Beacon Street, Boston, MA 02108

Houghton Mifflin Canada, 150 Steelcase Road, Markham, Ontario, L3R 1B2, Canada

Martin - Bearden, Inc., 1908 Sandy Lane, Irving, TX 75060

McGraw-Hill Book Company, 1221 Avenue of the Americas, New York, NY 10020

MECC, 3490 Lexington Avenue North, St. Paul, MN 55112

Microcomputer Resources, 2845 Temple Avenue, Long Beach, CA 90806

Milton Bradley, 443 Shaker Road, East Longmeadow, MA 01028

MIT Press, Cambridge, MA 02139

Prentice Hall Inc., Englewood Cliffs, NJ 07632

Prentice Hall International, Englewood Cliffs, NJ 07632

R.J. Bradley, Routes 197 and 450, Bowie, MD 20715

Reston Publishing Co., 11480 Sunset Hills Road, Reston, VA 22090

Scholastic Inc., 730 Broadway, New York, NY 10003

Scott, Foresman and Co., 1900 East Lake Avenue, Glenview, IL 60025

Sterling Swift, 7901 South IH-35, Austin, TX 78744

Stokes Publishing Company, 1125 Robin Way, Suite C, Sunnyvale, CA 94087

Sunburst Communications, 39 Washington Avenue, Pleasantville, NY 10570

Teachers College Press, 1234 Amsterdam Avenue, New York, NY 10027

Trillium Press, Box 921, Madison Square Station, New York, NY 10159

Turtle Publishing, 4810 Mariette Avenue, Montreal, Canada, H4V 2G1

Turtle Enterprises, N. 11515 Kathy Drive, Spokane, WA 99218

University Park Press, 300 North Charles Street, Baltimore, MD 21210

Wiley, John, & Sons Inc., 605 Third Avenue, New York, NY 10158

## CORPORATE SPONSORS

The National Logo Conference would like to thank
the following corporations for their generous support:

ATARI INC.

You are cordially invited to attend presentations by the Corporate Sponsors in Little Kresge on Thursday, June 28 at times to be announced at the conference.

## CONTRIBUTORS & EXHIBITORS

Contributor and exhibitor displays will be located in the Athletic Center. Representatives from the listed firms will be available in the exhibit area on Wednesday, June 27 (11:00 a.m. to 5:00 p.m.) and Thursday, June 28 (9:00 a.m. to 9:00 p.m.).

### Major Contributors

In addition to an exhibit, each of the following firms will be making a presentation in Little Kresge on Thursday, June 28 at times to be announced at the conference:

Acorn Computers Corporation

Coleco Inc.

Commodore International

IBM Corporation

Logo Computer Systems Inc.

Tandy Corporation

Terrapin Inc.

## Exhibitors

Acorn Publishing

Developmental Learning Materials

J.L. Hammett, Company Inc.

Holt Reinhart and Winston

Houghton Mifflin Company

McGraw-Hill Book Company

Microcomputer Resources Inc.

Scott, Foresman Electronic Publishing

Sunburst

Tuttle Products

Unicom

## Contributors

National Logo Exchange

## MEMBERS OF THE STEERING COMMITTEE

## MEMBERS OF THE ORGANIZING COMMITTEE

Gayle Fitzgerald
Special Events Office
Massachusetts Institute of Technology
Cambridge, Massachusetts

Richard Carter
Lesley College
Cambridge, Massachusetts

Sarah Clere
Special Events Office
Massachusetts Institute of Technology
Cambridge, Massachusetts

Jacqueline Karaaslanian
Logo Computer Systems Inc.
Paris, France

Pamela Davis
Massachusetts Institute of Technology
Cambridge, Massachusetts

Elizabeth Lawry
Massachusetts Institute of Technology
Cambridge, Massachusetts

## Acknowledgments

**The National Logo Conference would like to thank the following:**

Linda Bragman for assistance in the preparation of the Pre-Proceedings

Elizabeth Hacker of MIT Design Services for the graphic design of the conference invitation, the cover of the Pre-Proceedings, and the Logo 84 T-shirt

Peter McCann of Atari Cambridge Research for his assistance as Engineering Consultant

Joyce Tobias and Carolyn Markuson, Co-Chairpersons of the Exhibits Committee, and their staff: Jerry Short, Bob Bates, Terry Nowak, Peter Lavelle, Cindy Stephens, Robert Fardy, Bruce Wellman, Betty Bjorck, Michael Fung, and Frances Morse, for technical support during the conference

The MIT Special Events Office for their organizational support and assistance prior to and during the conference

Tom Lough, Barbara Elias, Carolyn Markuson, Regina Sapona, Joyce Tobias and Griff Wigley for compilation of the Logo bibliography found in the Pre-Proceedings

The MIT Physical Plant Department, particularly the Building Operations and Electrical Services personnel, for technical support during the conference

MIT Housing and Food Services, and Graphic Arts Service

Gten Fischer

Turtles' Anonymous.

Andricacos, ~~Peter~~ (Panos)
(Maria Antipa)

"Family Computing" mag.

IBM Turtle Power