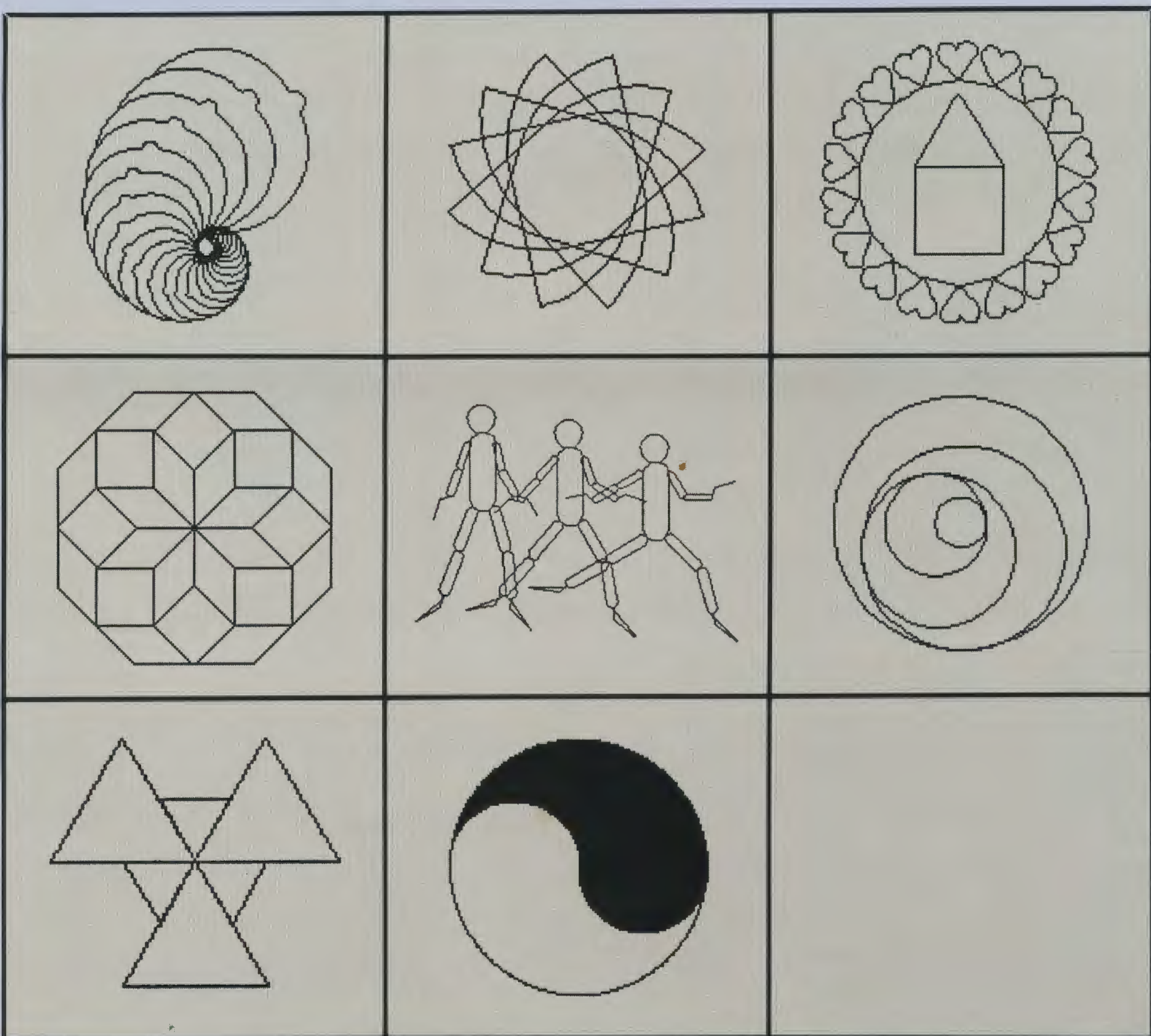
LOGO EXCHANGE

The Magazine for LOGO Activities Worldwide

ISSN 0888-6970 VOLUME 5 NUMBER 8 APRIL 1987



Meckler Publishing Corporation

Second Class
Pending
Westport, CT and
Add'l Mailing Offices

LOGO EXCHANGE

The Magazine for LOGO Activities Worldwide

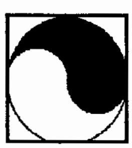
ISSN 0888-6970 | VOLUME 5 NUMBER 8 | APRIL 1987

Logo Exchange (ISSN 0888-6970) combines *National Logo Exchange* and *International Logo Exchange* and is published monthly September through May by Meckler Publishing Corporation, 11 Ferry Lane West, Westport, CT 06880, (203) 226-6967.

Application to mail at Second Class postage rates is pending in Westport, CT and additional mailing offices. POSTMASTER: Send all address changes to *Logo Exchange*, Dept. VV, P.O. Box 3000, Denville, NJ 07834

CONTENTS

International Logo News



Cover: The design added to this month's cover was contributed by Jeremy Franklin-Ross, a 6th grader at Bowen Elementary School in Newton Center, MA. He constructed the design without a FILL command.

Subscription: $29.95 per year. Subscribers outside the USA please add $15.00 for surface mail or $50.00 for airmail. Orders from North and South America should be sent to *Logo Exchange*, Dept. VV, P.O. Box 3000, Denville, NJ 07834; elsewhere to Meckler Publishing, 3 Henrietta Street, London WC2E 8LU, UK, (tel.) 01-240-0856.

From the Editor

by Tom Lough

Good Enough? Part 2

In the February issue, I offered some thoughts on the idea of "good enough." It was a topic about which I had been thinking for a long time. And, if you want the truth, I was somewhat scared as I wrote the editorial. (If you think I'm kidding, just ask my editorial reviewers!) I had never seen anything published about the idea before.

In the editorial, I suggested that we aim for excellence, do our best, keep our balance, and let that be good enough.

Your response to that editorial (and the rest of the February issue) was encouraging. Thank you for your notes and other communications. It is comforting to know that similar thoughts have occurred to you.

Based on this response, I would like to extend the idea of that editorial.

I know a large group of people who have a great many of both educational and personal responsibilities. Sometimes I wonder how they get anything done, the demands on them are so great. Yet, they manage to come through, performing admirably most of the time. Surely, they have a balanced view of what is "good enough."

I'm talking about our students.

It occurred to me that today's young people live in a world much more complex than that of my happy (and naive) adolescence. The demands of society, family, peers, and school are taxing. And yet, by and large, they manage to perform. How do they decide when something they on which they are working is "good enough?"

Since I didn't know, I decided to ask some students in our neighborhood. The response was quite interesting. In nearly all cases, the idea was something they had never really thought about.

As you might expect, I received a wide range of answers.

"It just feels right."

"I have included all the important facts and done all the important things."

"It is what I think my teacher expects."

"When my Mom says it is OK."

"My friends and I decide together."

"I don't ever know what's 'good enough' until I get my grade, and then it's too late."

As I thought about their interesting replies, however, I became aware that I was really taking aim at a related idea. If students go through a thoughtful process of deciding when something on which they are working is "good enough," then what impact do we as teachers have on that process, and, more importantly, on the self concept of the students themselves?

Several implications immediately suggest themselves.

Should students participate in their own grading or evaluation process? Perhaps. After all, if a Logo project which has been sandwiched in between the science fair and soccer practice does not live up to the expectations of the teacher, yet was judged to be "good enough" by the student in the face of other responsibilities, then maybe it merits a second look. How can we judge to what level a set of Logo procedures should be refined? Perhaps once the procedures are "good enough," the student wants to apply what has been learned to a different challenge.

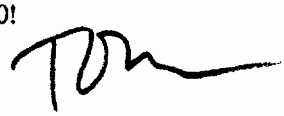
Are teachers viewed as hypocrites? Do we measure our own performance against a standard which we have determined is "good enough," while expecting perfection from our students? I laugh at and easily forgive myself for arithmetic mistakes at the blackboard during a lecture. Yet, I take off 5 points for such mistakes on student tests.

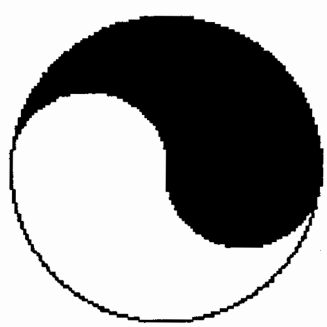
I am not suggesting a relaxation of academic standards. If there is a lesson in this at all, it is a suggestion that we give more thought to our students' situations. If we could become more sensitive to the pressures which they endure daily, the responsibilities which they bear with such grace, then maybe we could appreciate more their accomplishments and celebrate them accordingly.

In a number of instances, when students are allowed to judge their own work (with appropriate orientation), I have seen them be more critical than I expected. I wonder what would happen if this opportunity were extended somewhat?

How do *your* students decide when something is "good enough?" I think I'm ready to ask my own. Why not ask yours?

FD 100!





Logo Comes to Life

by Brian Silverman

Inspiration for Logo projects often comes from quite surprising places. This article describes a project that I worked on over the last couple of years. It's a Logo project in the sense that a Logo microworld is involved. It's a bit unusual, though, in that it didn't involve much geometry, turtles, or list processing. It didn't even involve very much programming.

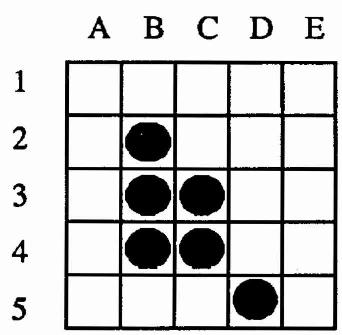
Let's Start at the Beginning

Several years ago while visiting a science museum, I saw an interesting contest. The goal was to build a structure out of toothpicks and glue which would hold up a brick between two supports. The winning design would be the one that used the fewest toothpicks. I thought this contest was particularly interesting because it presented both an engineering challenge and an easy way to select the winner.

A couple of years later, while discussing Logo projects, the idea of a contest came up. I thought it would be fun to have a programming contest based on the brick and toothpicks model, a challenge that had both some intellectual content and a straightforward way of measuring how well one is doing. To give this challenge a Logo flavor, it could be contained in the environment of a specially built Logo microworld.

To Life

In the late sixties, John Conway, a mathematician from Cambridge England, invented what he called a *zero player game*. It is not played competitively. It is not even really a solitaire. The so-called "Life game" [see "Mathematical Games," *Scientific American*, October 1970 and February 1971] was originally played on a checkerboard with checkers of a single color. An initial configuration is established by placing checkers on any number of squares. Here is an example.



A few transformation rules are applied simultaneously to all checkers in the configuration to form the configuration of the next "generation."

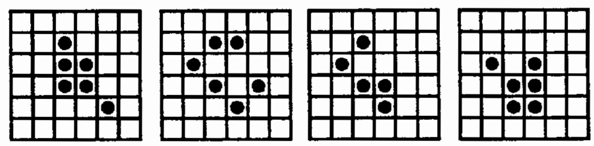
1. A checker with two or three neighbors (either adjacent or diagonal) remains in the next generation. For example, the checker at B2 has two neighbors [B3 and C3] and would remain. Similarly, the checker at B4 has three neighbors [B3, C3, and D4] and would remain.

2. With less than two neighbors, a checker "dies" of loneliness. The checker at D5 has only one neighbor [C4], and would not appear in the next generation.

3. With more than three neighbors, a checker "dies" of overcrowding. The checker at B3 has four neighbors [B2, B4, C3, and C4] and would not appear in the next generation.

4. An empty square with exactly three neighbors will experience the "birth" of a new checker, which appears in the next generation. Squares A3, C2, C5, and D4 each have exactly three neighbors (can you find them?). Checkers would appear in those squares in the next generation.

In transforming from one configuration of checkers to another and another, the design on the board changes in weird and wonderful ways. For example, the first few generations of our earlier configuration would look like this.



Logo and Life

Playing the game with checkers is rather tedious. The Life game is more often than not played on a computer screen with the computer applying the rules of transformation. What results is a kaleidoscope-like show in which "colonies" of checkers (or *cells* as they are often called) grow, shrink, die, give birth, and move in ways which undoubtedly inspired Conway to call his game "Life."

The thought of a contest gave me the excuse that I needed to play with Life. I'd already implemented and played the Life game on a number of different computers. These versions had seemed too self-contained. They didn't allow for any deep exploration. I suspected that a Life game Logo would be the ideal mechanism for exploring Life.

My plan was to think of a contest question in the context of the Life game and then let everybody play. At first, people could try things out at toplevel. As players became more comfortable with the game, they could write procedures to let them automatically try things out and save the results.

At LCSI, I had all the resources I needed. After a morning of assembly language programming, a Life Logo was "born." (This may have the marginal claim to fame of being the only Logo in the 1980's that doesn't have a turtle.) I made up a potential contest question and then went off to see if I could "win" the contest. The challenge was to find a starting position that lasted the greatest number of "moves" before becoming stable.

Life Logo has a 40x40 board which provides for an enormous number of starting positions. (A closer approximation of the actual number of starting positions would be several hundred digits long.) I wasn't able to think of any obvious way to find a position that would last any longer than any others. So, in true Logo style, I began to experiment.

Ready, Set, Grow

Most starting positions led to an evolution that had a wild flurry of growth at the beginning and then settled down into stability after less than a couple hundred generations. Stability in many cases occurred when all the cells had died. In other cases, the cells settled into configurations that either didn't change or that changed in a consistent pattern every few generations. Since Life Logo runs at the rate of about 10 generations per second, a typical starting configuration would last for about ten seconds or so. In rare cases, a position would last for a thousand or, even more rarely, for several thousand generations.

After a couple of days of random experimentation, I decided to automate the process. I wrote some procedures that would run a position until it became stable, others that generated starting positions based on a particular pattern, and still others that keep track of how well a position was doing.

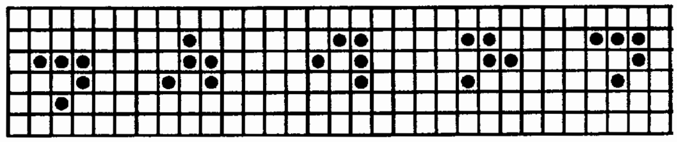
I experimented in this way for the better part of a year, but I had no luck. Even after a year of trying I don't think I found a single clue about how to win the contest. At some level that was encouraging. It meant that in the contest beginners would have as much chance as experienced experts.

Computing and Life

About that time, I got my hands on a copy of a book called *The Recursive Universe* (William Poundstone, New York: William Morrow and Company, 1985). The even numbered chapters are about the Life game and the odd numbered chapters are about thermodynamics, information theory, cosmology and things like that. The book also mentions that Conway proved it was possible to build computers in the Life world.

That notion intrigued me. It was somehow possible to construct a starting position that was a representation of a question, run it through a series of transformations, and that would produce the representation of the answer.

The key thing in building Life game computers is the ability to transfer information from one part of the board to another. In the Life game you often see "gliders." A glider is a five cell configuration that appears to move across the board. They are not mentioned in the rules. It just so happens that when the rules are applied to a certain configuration of five cells, after four generations, the same configuration reappears one cell up and one cell over. Since gliders are able to "move" from one place to another, they can be used to carry information.



Any information processed by a computer is boiled down to a huge collection of bits. A bit is just a choice between one of two possible states, often simply the numbers 1 and 0. Imagine a long stream of gliders moving from one point to another. This stream can be used to carry information. A glider present in the stream can be called a "1." An glider missing from the stream can be called a "0."

Unfortunately, it turned out that Conway's proof was not terribly useful, though theoretically correct. The smallest components required to do anything interesting are huge. A configuration to add two numbers together requires a board thousands of cells across. My Apple's 40x40 board started looking very small. I wasn't too happy about the idea of finding several thousand more Apples to make up the difference.

Breaking All the Rules

At that point I remembered reading that Conway developed the rules of the Life game so that the results would be as unpredictable as possible. If the rules were different and more predictable, it might be easier to find small computational elements. I redesigned the Life Logo, adding a few primitives that allowed a player to change the rules.

I looked for other rules with simpler gliders and simpler ways of manipulating them. One of the first things that I discovered was that Conway's rules, which at the outset appear a bit bizarre, are really not all that surprising. Similar but not identical rules just didn't work as well. If the rules allow for easier birth or easier survival, then the board quickly gets cluttered. If they make it harder for birth or survival then most starting positions quickly die back to near nothingness. Conway's rules represent an appropriate balance that lead to the pyrotechnics that have made Life so popular.

I did manage to find a simpler glider. In fact, I found many different sets of rules that had lots of different species of gliders. Some of the rules let me build the computing "circuits" that I wanted to build. Others were very pretty and just fun to watch. Even others made me take a step back and want to do some more serious reading into topics ranging from philosophy to number theory.

Where was Logo through all of this? It certainly wasn't at the center of attention. As it turned out, Logo proved to be one of the easiest ways in to a complicated problem. Life Logo contains only a few primitives which can be used to construct most of the other "primitives" of the system. Logo provided the "glue for the toothpicks," even though I'm not sure whether what I did would support that "brick." But the important thing is that Logo helped me to grow intellectually and have fun at the same time.

And it all started because of an idea for a contest.

Brian Silverman is the Vice President for Research at Logo Computer Systems, Inc.

Tipps for Teachers

by Steve Tipps

Probably

Probability is probably the most interesting, the most useful, and the most neglected topic in mathematics. If events in the world were certain and definite, we would have little need for studying random and chance occurrences. Excluding death and taxes (as the old saw goes), nothing in life is certain. Everyday life is full of events which are uncertain: whether to buy the extended warranty, precipitation, cars starting on a cold morning, getting the grade you want, whether to move or stay, winning the $2,000,000 Publisher's Clearinghouse sweepstakes. Many things in life must be considered as likely or unlikely rather than certain. Some information about probability helps.

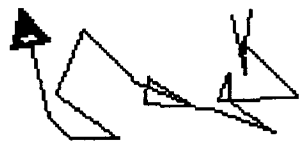
Before beginning a formal study of probability, students at all levels need time and organized experiences with uncertainty. Logo can be very useful in illustrating certain events and speeding their exploration.

Randomly

The RANDOM command in Logo is often introduced by having the turtle move and turn randomly. SCRAMBLE uses RANDOM for both distance and direction.

```
TO SCRAMBLE
RIGHT RANDOM 360
FORWARD 10 + RANDOM 20
END
```

?REPEAT 20 [SCRAMBLE]



Random walks can also be used in other ways. Consider a person walking down a street. At the end of each block, s/he flips a coin to decide whether to continue forward or turn around and walk back a block. What would be the path of the person? How far from the starting place would s/he be after several coin flips?

First act out this situation with a coin and a number line, perhaps with masking tape on the classroom floor. Students need a real sense of the situation before going to the more abstract representation on a number line. Some may prefer to use a number line or ruler on their desk to work through the situation.

Flipping and moving can become laborious. However, the turtle can flip and zip along the number line with ease. You can make a Logo coin to flip.

```
TO FLIP
OUTPUT 1 + RANDOM 2
END

TO FLIP.COIN
TEST FLIP = 1
IFTRUE [OP "H ]
IFFALSE [OP "T]
END
```

Based on the results of the flip of the coin, the turtle moves one block east or one block west. Each block in this case will be 10 turtle steps.

```
TO WALK
TEST FLIP.COIN = "H
IFTRUE [ SETHEADING 90]
IFFALSE [ SETHEADING 270]
FD 10
END
```

A handy tool for determining the distance from HOME is to print the XCOR or the XCOR/10. A BLOCKS? procedure is also useful to get the position.

```
TO BLOCKS?
PRINT XCOR / 10
END

?CS
?REPEAT 10 [ WALK ]
?BLOCKS?
```

Chance determines how far from the starting place (HOME) you would be after taking 10 random walks, or 100 random walks, or 10 sets of 10 random walks. What is the chance that all walks will be headed north or headed south? What is the chance that after 10 walks you would wind up back at home? Although these probabilities can and should be calculated, the purpose of this exploration is to give students a feeling for the phenomenon.

To begin the exploration, ask the students to guess how the turtle could go in either direction. How likely the turtle is to wind up 10 blocks away from the starting place? A chart is useful to record their guesstimates. The first line shows one 7th grade student's guess of the distribution.

BLOCKS WEST										HOME										BLOCKS EAST
10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10-
1	1	1	1	1	2	3	5	5	10	40	10	5	5	3	2	1	1	1	1	1

The last line is for tallying the results of the 100 sets of 10 random walks. When the student or small group is finished collecting data, they can compare their guess to the observations and discover some interesting things about the likely position of the turtle. Write down your guesses before taking the turtle for a walk.

Advanced students can even create a list of the blocks and draw a graph of the results on the screen. Procedures from columns entitled Survey Sense I, II, and III (see the March, April, and May 1985 issues of the *National Logo Exchange* newsletter) can be adapted to this end.

Quarterly

Another random walk involves random direction. A coin is tossed to determine whether the direction of the turn is left or right. A die is tossed to determine the size of the turn. Again, acting out this situation is important to get a feel for the experience. Many schools have a four-square grid painted on the playground. Have a student stand at the center of the four-square. Flip a coin and roll a die.

Have students decide whether heads or tails means right or left. Each number of the die multiplies 30 degrees. So, "heads" and 5 could mean turn right, 5*30 degrees. Take one step in that direction. Continue for 5, 7 or 9 more moves. Record the quadrant or square in which the student is standing. Students should work with this until they discover that not all the 30 degree turns have an equal chance to be chosen: a turn for 0 degrees is left out. They will have to decide how to compensate for this problem.

After working out the problems with the model, they are ready for Logo simulations. Some students will be able to write their own procedures. You may decide to provide all or part of the tools for the exploration. One possible set of procedures follows. FLIP.COIN and FLIP are the same as used in the WALK procedure.

```
TO WALKER
TEST FLIP.COIN = "H
IFTRUE [ TURN1]
IFFALSE [TURN2]
FORWARD 10
END

TO TURN1
SETHEADING 0
RIGHT 30 * DIE
END

TO TURN2
SETHEADING 0
LEFT 30 * (DIE - 1)
END

TO DIE
OUTPUT 1 + RANDOM 6
END

?REPEAT 10 [ WALKER ]
```

Students could experiment with the turtle WALKER and make a record of which quadrant of the screen the turtle is in after 10 moves. Another interesting activity is to record how far the turtle is from HOME. The Pythagorean Theorem is used to find the distance. Some of the students can write the procedure if they know about XCOR and YCOR functions which give the location of the turtle.

```
TO HOW.FAR?
OUTPUT SQRT (XCOR*XCOR) + (YCOR * YCOR)
END
```

This procedure may give a rather cumbersome number such as 43.1499783. If you include the ROUND function in HOW.FAR?, a tidy although slightly inaccurate 43 results.

Now all the tools are ready for the exploration. A chart which includes the trial, the location, and distance can be made to record the results. What is the farthest that the turtle could move? How likely is that to happen? After discussing what is possible, students should guess the number of times the turtle would wind up in each section and the average distance from HOME. Another variation is to take the SETHEADING 0 out of TURN1 and TURN2. Compare the results of the two variations.

Likely

Creating random situations with a coin and a die provides the beginning for many experiments in probability. Next month, it is likely that the discussion of probability will continue with other Logo tools and explorations. Of course, nothing is certain...

(Ed. note: It is also probable that LX readers would enjoy Steve Tipps' previous treatments of the concept of randomness. See "Tipps for Teachers" in the March 1984 and April 1986 issues of the *National Logo Exchange* newsletter.)

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX, and has been involved with Logo since 1982. He conducts Logo workshops for school systems throughout the United States, and is a popular conference speaker. His CompuServe number is 76606,1623.

Teacher Feature

by Rebecca Poplin

Featuring: Theresa Overall

Lighting the Lamp

A child is not a vessel to be filled but a lamp to be lighted.

This motto is the guiding philosophy of Lamplighter School in Dallas, Texas. Theresa Overall is one of the many teachers at Lamplighter who have been involved with Logo since the fall of 1978. Not only does Lamplighter have an impressive Logo history, it is also a fascinating setting for learning.

The school is about 30 years old and serves children from age three through the fourth grade. The philosophy of the school encourages teachers to look for any tool which will enhance learning. A greenhouse and a barn are adjunct classrooms on the school property and work in them is a part of the curriculum.

Erik Jonsson, a former mayor of Dallas and one of the three founders of Texas Instruments, is an active supporter of Lamplighter and honorary chairman of the board of directors. Mr. Jonsson met Seymour Papert through a mutual friend at MIT, saw Logo there, and became excited about Logo's potential for Lamplighter students. He was the thread which wove MIT, TI, and Lamplighter together.

Turning On the Lights

At the time of the first Logo implementation, Theresa was a share teacher in third grade. (A share teacher is a regular teaching team member with a variety of duties.) Theresa was asked if she would mind learning how to turn the machines on and off. She was then asked if she would become coordinator of the computer program. Part of her job was to see that the teachers became involved with Logo and the computers. A core group of five teachers from all different grade levels was initially trained by MIT personnel. Then those teachers trained their teammates.

For the first five years, Theresa had two half-time jobs: coordinator of the computer program and share teacher for third grade. Now she is a full-time classroom teacher in fourth grade and Coleta Lewis serves as coordinator of the program and half-time Kindergarten teacher. Theresa, Coleta, and the rest of the teachers have worked hard at continuing to grow and change with Logo. Each year the program has had to change somewhat because the students have had more and more experience. This has been the first year to have students in the fourth grade who have had computer experience since they were three years old.

Keeping the Light

Lamplighter teachers adhere to the "computer as pencil" philosophy, so all computers are in the classrooms in the shared open spaces. The primary grades have two computers per room while the upper grades have a ratio of one computer for every five students. Nursery and kindergarten have one computer in a room.

Nursery and kindergarten students have teacher-designed microworlds and work mainly with one-key programs. TI Logo sprites are favorite feature to play with. Students do a lot of animation, build things, and create movies. In the lower grades, teachers may have a half hour a week to work on computer skills.

In Theresa's fourth grade, students don't have computer class, but the teachers schedule computer time for whatever projects and experiences seem appropriate. Theresa recently used the turtle as an introduction to x- and y-coordinates. She found that the students already really knew that x comes before y, because in their Logo experience if they used the coordinates incorrectly, pictures would be sideways or garbled.

In another recent project, students each created a Texas notebook. Some requirements were the state flag, the state bird, a favorite Texas city, something else Texan, and a Texas computer project. The computer projects varied from a reenactment of the Battle of the Alamo to a graphic display of the legend of the horned frog. Computer skills are taught as students need them.

In the first years of Logo at Lamplighter, Theresa reports that students had wilder ideas about what they could accomplish with the computer. Now the majority start with a simple idea and build on it, with an increasing sense of accomplishment.

Students have become better at planning and breaking projects into modules. Teachers work with the students on brainstorming project ideas and subjects, getting kids to brainstorm with each other, and planning projects. Often Theresa first has the students draw pictures of what they want it to do. She then has them focus on what part of the picture was drawn first.

Passing the Torch

Coleta and Theresa have written a TI Logo Curriculum guide which is available. The guide explains the concept and describes ways to use Logo with students. There is also a list of bugs students might encounter.

Teachers at each grade level at Lamplighter have a notion of what students should be able to do, but there are no strict guidelines about what should be "covered." The sequence of instruction really depends on the needs of the students.

Both Coleta and Theresa have worked individually with the teachers at Lamplighter, because each teacher has unique needs too. Since the computers have been there so long, there is now a different focus of training new teachers, working on group projects, curriculum development, and creating new ways to develop projects. The librarian, music teacher, and art teacher all use Logo. It is a topic of discussion at staff meetings, and teachers frequently improve their own Logo skills.

Theresa firmly believes that if you took all of the computers out of Lamplighter tomorrow, they would still be better off for having had them. Teachers have become more aware of the learning process, and Logo has made all of them much better teachers. If you are interested in learning more about Theresa Overall, Coleta Lewis, or the program at Lamplighter, write to:

The Lamplighter School
11611 Inwood Road
Dallas, Texas 75229

If you know of anyone who should be featured in this column, please write:

Rebecca L. Poplin
2421 Fain Street
Wichita Falls, Texas 76308

Rebecca Poplin uses Logo to teach computing and mathematics at a junior high school in Wichita Falls, TX.

Logo Publications

Kaleidoscopes is a quarterly Logo newsletter with ideas and suggestions for intermediate and advanced Logo users. The editors, Alison Birch, Larry Davidson, and Phil Lewis, have among them many years of Logo experience and have authored several significant Logo books and other publications.

The content of the newsletter is appropriate for teachers at the junior high and senior high school levels. Topics such as language exploration, mathematical investigations, and computer science activities have been treated in past issues.

Dan Watt has enthusiastically endorsed *Kaleidoscopes* as "... the only source of advanced, esoteric information about the uses and workings of Logo."

Subscriptions cost US$12.00 per year (4 issues) for addresses in the USA and Canada, and US $18.00 elsewhere.

For more information, write to:

Kaleidoscopes
Computers for a New Education
8 Lowell Avenue
Suite 100
Malden, MA 02148

Polyspiral is the newsletter of the Boston Computer Society Logo Users Group. It contains interesting Logo articles as well as Logo User Group news. For information, write to: Polyspiral, Boston Computer Society, One Center Plaza, Boston, MA 02108.

Teaching Tools

by Glen Bull
and Paula Cochran

Plotting Tools: Part II

Microcomputer Based Laboratories

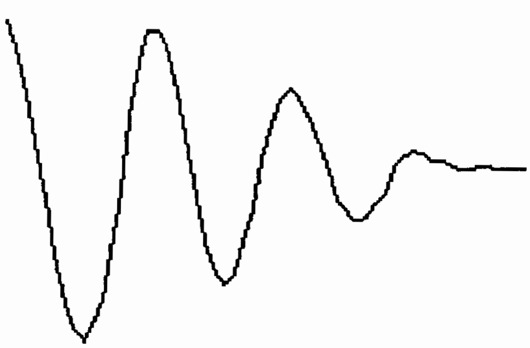
Numbers which are plotted can be typed in by hand, or generated by the computer. There is a third source of numbers which can be plotted with PLOT tools. Two previous "Teaching Tools" columns (January and February 1987 *Logo Exchange*) were about ways of acquiring data from the outside world. The January column (Science and Sensors: Part I) was about conversion of analog data to digital form via the game port, using the motion of a pendulum as an example. The February column (Science and Sensors: Part II) demonstrated how to store data in the form of a list once it is acquired, using temperatures acquired with the temperature probe from a Brøderbund Science Tool Kit as a sample illustration.

It is no coincidence that the PLOT tool is designed to operate on values stored in the form of a list. Any data acquired from sensors measuring the external environment can be displayed with PLOT and similar Logo tools once the data is digitized. For example, the figure of a decaying sine wave in the January column was developed by storing the list of numbers produced by a swinging pendulum. The list of numbers obtained looked like this:

```
?PRINT :PENDULUM.DATA
61.0 50.1 33.1 15 -7 -27.1 -45.4 -54.2 -59 -53.4 -43.8
-26.4 -3 24 45 57 57 51 34.2 16.8 -6 -21.6 -34.2 -37.8 -33.6
-22.8 -9.6 7.8 21.6 30.6 34.8 31.2 23.4 15 4.2 -7.2 -12 -14.4
-13.2 -8.4 -3.6 5.4 10.2 12 11.4 8.4 7.8 6.6 4.8 4.8 6 6 4.8
4.8 4.8 4.8
```

The data acquired in this way was plotted with the PLOT tool.

```
?PLOT :PENDULUM.DATA
```



A pendulum swings too fast to permit manual measurement of its motion; a computer provides the most practical tool for doing this. It is possible to measure temperatures with a thermometer and write them down manually, since changes occur at a relatively slow rate in many cases. However, compare the following two sequences:

DATA OBTAINED MANUALLY

1. Measure temperatures with thermometer.
2. Write down values in column on sheet of paper.
3. Add values manually and obtain average.
4. Plot temperatures with pen and graph paper.

DATA ACQUIRED WITH COMPUTER

1. Monitor data with temperature probe and digitize.
2. Store values in the form of a list.
3. Obtain average with Logo AVERAGE procedure.
4. Plot values with PLOT tools.

Both methods are worthwhile. In fact, it will be easier for students to understand the function of the Logo procedures if the same process is done manually first. However, addition of the microcomputer-based measurement procedures extends the capabilities of the student, and permits experiments which would otherwise be too tedious or time-consuming. Thus, the combination of a computer, Logo, and external sensors permits the student to perform science in the way scientists do.

A Function Generator

Last month a Logo function generator was shown. A function generator takes a mathematical function, and generates a series of numbers associated with the function. In our earlier example, the function $Y = X^2$ was produced.

```
TO GENERATE.FUNCTION
MAKE "DATA [ ]
MAKE "RANGE 10
FUNCTION.LOOP -1 * :RANGE
END

TO FUNCTION.LOOP :NUMBER
IF :NUMBER > :RANGE [STOP]
MAKE "DATA LPUT PARABOLA :NUMBER :DATA
PRINT LAST :DATA
FUNCTION.LOOP :NUMBER + 1
END

TO PARABOLA :NUMBER
OUTPUT :NUMBER * :NUMBER
END
```

This function generates a parabola:

```
?GENERATE.FUNCTION
100
81
etc.
```

The procedure produces a variable containing a list of all the numbers which we might otherwise have typed in by hand:

```
?PRINT :DATA
100 81 64 49 36 25 16 9 4 1 0 1 4 9 16 25 36 49 64
81 100
```

Here is the plotting tool from last month's column which was used to plot the list of numbers. A setup procedure is also needed, to position the turtle at the left-hand side of the screen before plotting begins. (Note: PLOT1, PLOT2, and PLOT3 were earlier versions of the PLOT tool described in the March column.)

```
TO SETUP
CS PU LT 90
FD 140 RT 90
END
```

```
TO PLOT4 :NUMBERS
IF EMPTYP :NUMBERS [STOP]
SETPOS LIST (XCOR + 10) (FIRST :NUMBERS)
PENDOWN
PLOT4 BUTFIRST :NUMBERS
END
```

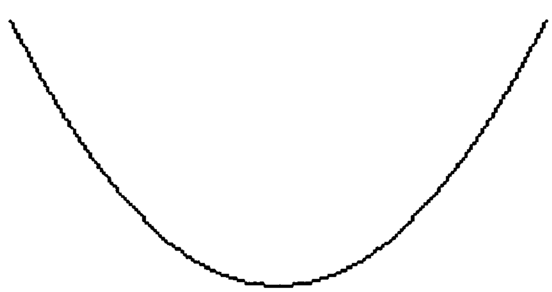
In these and subsequent plot procedures, the following modifications are required for Terrapin Logo:

1. Substitute EMPTY? for EMPTYP
2. Omit brackets around STOP
3. Substitute SETXY for SETPOS LIST

```
TO PLOT.4A :NUMBERS
IF EMPTY? :NUMBERS STOP
SETXY (XCOR + 10) (FIRST :NUMBERS) PD
PLOT.4A BUTFIRST :NUMBERS
END
```

When the data are plotted, using the plotting procedures provided above, the graph of a parabola is produced.

?SETUP
?PLOT4 :DATA



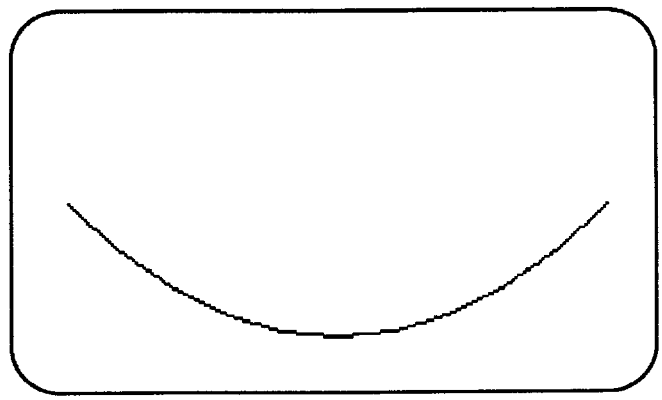
Changing the Vertical Scale

In some cases the figures plotted will go off the top of the Logo screen. In that case, it will be necessary to add a scale factor to the plot procedure. The same method can be used to shrink or flatten the vertical dimension of figures which you wish to make smaller. For example, to make a flatter parabola, each number could be multiplied by 0.5 as it is plotted.

```
TO PLOT5 :NUMBERS
IF EMPTYP :NUMBERS [STOP]
SETPOS LIST (XCO +10) (FIRST :NUMBERS) * 0.5
PENDOWN
PLOT5 BUTFIRST :NUMBERS
END
```

The result is a squashed parabola:

?SETUP
?PLOT5 :DATA



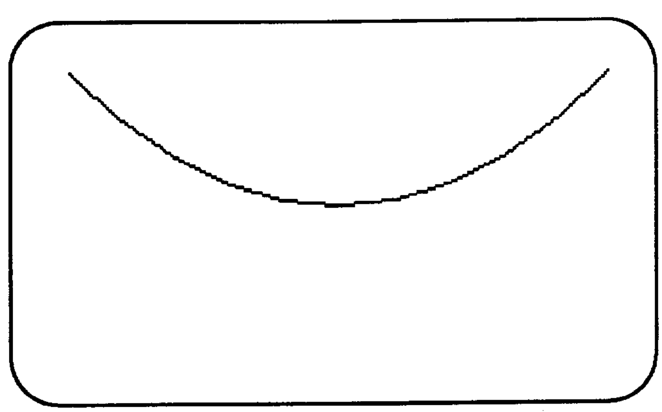
Changing the Vertical Offset

At times it may be desirable to move an entire figure up or down on the screen. This can be accomplished by adding or subtracting the desired shift of the figure. For example, to move the parabola up by 50, the following change would be made to PLOT.

```
TO PLOT6 :NUMBERS
IF EMPTYP :NUMBERS [STOP]
SETPOS LIST (XCOR + 10) (FIRST :NUMBERS) * 0.5 + 50
PENDOWN
PLOT6 BUTFIRST :NUMBERS
END
```

When the parabola is plotted with PLOT6, an offset of 50 will be added in, shifting the figure higher on the screen.

?PLOT6 :DATA



Using Scaling Techniques

Scale and offset capabilities make it possible to plot a range of numbers which otherwise would not fit on the screen. For example, the function generator was initially set up to generate the range of numbers between -10 and +10, so that all the numbers would fit on the screen. With scaling techniques, it is now possible to plot a larger range of numbers.

First change the range of numbers generated by the function generator by replacing the number 10 with 20:

```
TO GENERATE.FUNCTION
MAKE "DATA [ ]
MAKE "RANGE 20
FUNCTION.LOOP -1 * :RANGE
END
```

Next, run the procedure to produce a new series of numbers:

```
?GENERATE.FUNCTION
400
361
324
etc.
```

When you print out the data generated, you will see that the range of numbers is greater than it was previously.

```
?PRINT :DATA
400 361 324 289 256 225 196 169 144 121 100 81 64
49 36 25 16 9 etc.
```

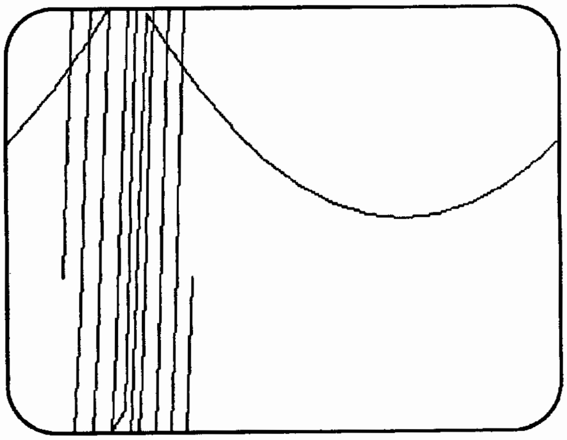
Since the number 400 will not fit on the Logo graphics screen, it will be necessary to scale the data as it is plotted. Two changes to PLOT5 will be required to accomplish this scaling. The horizontal scaling must be changed from "XCOR + 10" to "XCOR + 5" and the vertical scaling must be changed from "0.5" (fifty percent reduction) to "0.2" (twenty percent reduction).

```
TO PLOT7 :NUMBERS
IF EMPTYP :NUMBERS [STOP]
SETPOS LIST (XCOR+5) (FIRST :NUMBERS) * 0.2
PENDOWN
PLOT7 BUTFIRST :NUMBERS
END
```

Now let's examine the effects of plotting the data with and without proper scaling. First plot the data without appropriate scaling.

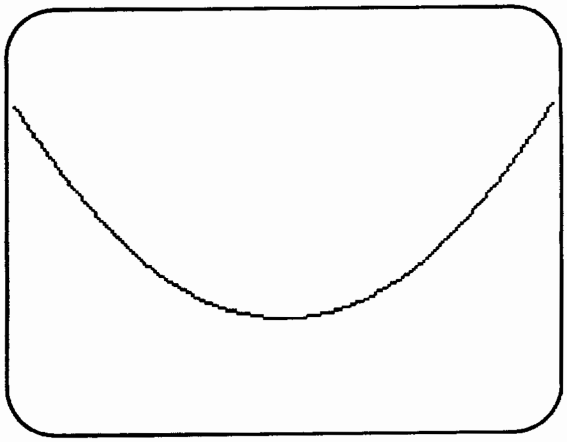
```
?SETUP
?PLOT5 :DATA
```

The graph will shoot off the top of the graphics screen, creating an indecipherable mess.



Next try plotting the data with proper scaling. This time the numbers all fit on the screen. Scaling makes it possible to handle a wider range of numbers than could be displayed otherwise. The issue of scaling can serve as introduction to a wide range (no pun intended) of subjects, such as logarithmic functions.

```
?SETUP
?PLOT7 :DATA
```



The Algebra in PLOT

The PLOT procedure itself may be of interest to students studying algebra. Consider the slightly rearranged PLOT6 procedure:

```
TO PLOT6 :NUMBERS
IF EMPTYP :NUMBERS [STOP]
SETPOS LIST (XCOR + 10) (0.5 * (FIRST
:NUMBERS) + 50)
PENDOWN
PLOT6 BUTFIRST :NUMBERS
END
```

In the foregoing example, the second line of the procedure itself contains the following algebraic function to determine the y-coordinate:

Y = AX + B

This may be easier to see if the elements of the Logo command are rearranged beneath their equivalents in the algebraic expression.

Algebra: A * X + B

Logo: 0.5 * (FIRST :NUMBERS) + 50

Summary

Logo plotting tools can enhance other Logo procedures and applications. This is particularly the case when large quantities of data have been generated by the computer. The plotting tools are also particularly useful when large quantities of data have been acquired from external sensors by Logo procedures. In these cases, it may be tedious to re-enter the data by hand so that it can be plotted by another software package. Control over the plotting procedures makes it possible to control the exact appearance of the output. Finally, the composition of the plot procedures may be of interest to an algebra class. Thus, PLOT procedures provide a tool for study and a tool to be studied.

Glen Bull is a professor in the University of Virginia's Curry School of Education, and teaches Logo courses at both the graduate and undergraduate level. His CompuServe number is 72477,1637. Paula Cochran is an assistant professor in the Communication Disorders Program of the University of Virginia's Curry School of Education. She is interested in Logo applications for language arts and special populations.

New LogoWriter Prices Announced

Logo Computer Systems Inc. (LCSI) has announced a series of new purchasing options for LogoWriter. In addition to the site license agreement previously offered, LogoWriter can now be purchased in single-disk starter sets ($129) and six-disk lab packs ($299), both with accompanying support materials.

LCSI has also restructured its LogoWriter site license agreement by eliminating the annual renewal fee. For a one-time fee of $450, the licensed school can make unlimited copies of LogoWriter (currently available for the IBM PC, IBM PCjr, Apple II family, and the Commodore 64) for in-school use. For an additional $150 fee, all students attending the licensed school can use LogoWriter in their homes.

For more information about LogoWriter or the various licensing or purchasing options, contact LCSI, 121 Mount Vernon Street, Boston, MA 02108, or call (617) 742-2990.

TO BEGIN :PROJECTS

by Elaine Blitman and Barbara Jamile

Previous columns have each presented a single theme for exploration, along with ideas for both on and off-computer activities. This month we've collected a few examples of projects that children have done at our school, along with descriptions of how they did them. As you'll notice, there's no single way to teach or to learn Logo, even in the same school environment. Some of the best learning experiences occur unexpectedly when you take advantage of a teachable moment, or when you observe one child helping another...and "aha" happens!

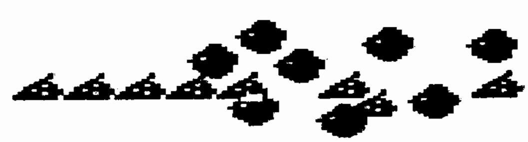
One to One Introduction

Ariel is a kindergartener in Nancy Kulp's class who enjoys writing, and who already had some experience with a typewriter as well as with a computer. This is what he produced about an hour and a half after his first encounter with LogoWriter, with an adult sitting nearby to answer (and ask) questions. He asked for help to spell some of the words; otherwise the vocabulary is his own. He drew the fish and waves as sprites and placed them carefully with the STAMP function.

Ariel's focus and attention span are remarkable for his age. He will probably be interested in helping other children while they're learning to do projects with LogoWriter, for he has patience, mature social skills, and an inquiring mind. At one point, while he was revising the spelling of a word, he mentioned that his father's word processor could overstrike letters. Pleased that Ariel recognized overstrike and insert, and that he had transferred that understanding to this experience, we talked about how computer programs are different and alike.

The sea is full of lots of living animals and plants. Some of the animals living in the sea are very small. The fish are very beautiful and they swim in schools. Some of the animals swimming in the ocean are very dangerous and big.

Ariel Goldberg



The children in this kindergarten classroom ordinarily use MECC's EZ Logo (MECC, 3490 Lexington Avenue North, St. Paul, MN 55112). They have had some off-computer experiences with the concepts that they explore with this simplified version of Logo. Two Apple II+ computers are available for the children to use on a rotation schedule. An Apple IIe is shared with the class next door.

Teacher / Student Language Experience

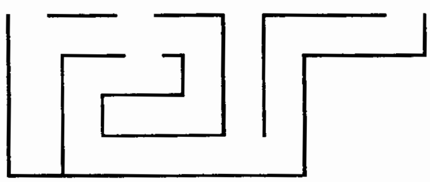
First grade teacher Betsy Hata introduced beginning LogoWriter concepts to her class at the beginning of the school year. Then she used LogoWriter word processing and a language experience technique to write about their activities with the class. A 19" monitor, connected with a Y cable to the small computer monitor, made it easier for all of the children to read and suggest revisions for their joint composition. Each child has a copy for his or her computer notebook.

Home / School Cooperation

Because of changing hardware and software, the second graders in Vickie Bisho's class used TI Logo in kindergarten and LCSI Logo in first grade. Since the beginning of this school year, they've been working with LogoWriter. Under the provisions of the site license agreement with LCSI, the school may loan LogoWriter disks to students for use at home.

Shawn Lee is one of the children whose parents have requested participation in our "LogoWriter at Home" program. Shawn is a self-starter, one of those individuals who likes to find out about things and go his own way. He uses the activity cards and reference manual in the classroom to learn new techniques for some of his projects. His teacher says that he's very helpful to other children when they're doing LogoWriter projects. This is one that he did during Christmas vacation, inspired by his teacher's gift:

Dear Mrs. Bisho,
Thank you for the Christmas maze book.
I love mazes!



Love,
Shawn Lee

P. S. I'm writing to you on LOGO WRITER
Here is a maze I made using LOGO WRITER

After Shawn shared his work, the first grade teachers made transparencies from his maze and now the younger children are trying to guide the turtle around it in six different classrooms!

Elaine Blitman and Barbara Jamile are the K-2 and 3-4 supervisors at the Punahou School in Honolulu, HI. They have been using Logo with young children since 1982. Their CompuServe number is 76067,211.

Logo LinX

by Judi Harris

DisSPELLing a Myth

My students took spelling tests on words that they hadn't studied or memorized.

No, it wasn't cruel and unusual punishment. It was a different way to learn to spell. And, considering that they spelled most of the words (that they hadn't studied) correctly, I'd be willing to wager that either they learned to spell without memorization, or they were excellent spellers already.

If you had seen their writing in September, you would have known for sure that they were *not* national spelling champions.

A Curricular Catalyst

"So what's so Logo-like about spelling?" you ask.

The answer is simple: *patterns.* Students can learn to spell better by recognizing and applying letter patterns, instead of memorizing 20 new words each week.

AIRS (Andover's Integrated Reading System), a mastery learning language arts program, contains a spelling module called "Structural Skills" that helps children to spell in this way. It is based upon the generalizations of eight structural skills: plurals, derived words, possessives, contractions, root words, hearing syllables, syllabication, and prefixes / suffixes.

With the AIRS program, students first learn a spelling *rule* by examining and discussing words that follow the same spelling *pattern.* For example, what pattern do you see in the spelling of these root words?

START BUZZ CRISP BLINK LOUGH

All of them end with two consonant letters. If we wish to add a suffix, such as -s, -es, -est, -ed, or -ing to any of them, we can do so without altering their spelling.

STARTED BUZZES CRISPS BLINKED LOUGHING

The same is true for these words,

APPEAR SEAM COOL TOOT NAIL BROAD

although they follow a different pattern. Can you see it?

Once students can recognize a spelling pattern and remember the rule that accompanies it, they can spell most words that they hear. There are a total of 39 structural skill rules in the AIRS spelling materials used in grades 3-8. That's fewer than two weeks' worth of things to remember, according to traditional spelling instructional methods.

DisSPELLing can be fun!

Paula Cochran and Glen Bull's "spelling fuzzies" ("SpecialTalk," *LX*, October 1986) are a wonderful way to reinforce correct spelling using Logo. The AIRS materials inspired thoughts about some interesting ways to help insure correct spelling by attending to letter patterns.

Each of the words in the first sample set above ended with two consonants. Let's see how to tell Logo to check a word for that pattern.

First, we need a procedure to tell if a letter is a consonant.

```
TO CONSONANT? :LETTER
OUTPUT NOT (MEMBERP :LETTER [A E I O U])
END
```

(Terrapin Logo: change MEMBERP to MEMBER?)

To see how the procedure works, type

```
?PRINT CONSONANT? "L
TRUE
```

The following procedure will check for consonants as the last two letters of a word.

```
TO TWO.END.CONSONANTS? :WORD
OUTPUT AND CONSONANT? (LAST BL :WORD) CONSONANT? (LAST :WORD)
END
```

(Terrapin Logo: change AND to ALLOF)

Now you can type

```
?PRINT TWO.END.CONSONANTS? "TURTLE
FALSE
```

Impelled to Spell

Did you figure out the second sample letter pattern above? All of those root words have two *vowels* together. Here is a procedure that determines if a single letter is a vowel:

```
TO VOWEL? :LETTER
OUTPUT MEMBERP :LETTER [A E I O U]
END
```

(Terrapin Logo: change MEMBERP to MEMBER?)

The next procedure will check recursively a word of any length to see if it has two vowels together:

```
TO TWO.VOWELS? :WORD
IF (COUNT :WORD) < 2 [OP "FALSE]
IF AND (VOWEL? FIRST :WORD) (VOWEL? FIRST BF :WORD) [OUTPUT "TRUE]
OUTPUT TWO.VOWELS? BF :WORD
END
```

(For Terrapin Logo, change AND to ALLOF and eliminate the square brackets.)

Now, let's make the computer generate some root word and ending combinations that follow the "two vowels" pattern.

```
TO TWO.VOWEL.WORDS
OUTPUT [COOL MAIL NAIL WEED TOAST BREAD LEAP GROAN SEED LOAD]
END

TO ENDINGS
OUTPUT [S ED ER ING]
END

TO PICK :OBJECT
OP ITEM (1 + RANDOM (COUNT :OBJECT)) :OBJECT
END

TO DERIVED.WORD
OUTPUT WORD (PICK TWO.VOWEL.WORDS) (PICK ENDINGS)
END
```

Type PRINT DERIVED.WORD, and the computer might return NAILED or BREADING or GROANING. If you enter REPEAT 5 [PRINT DERIVED.WORD], what do you think will happen?

Now, let's combine these procedures into a superprocedure that will help students to recognize and extend spelling patterns.

```
TO PATTERNS
CLEARTEXT
PRINT [HERE ARE SOME WORDS:]
PRINT TWO.VOWEL.WORDS
PR [ ]
PRINT [PLEASE TYPE A ROOT WORD THAT]
PRINT [FOLLOWS THE SAME PATTERN.]
PRINT [(IF YOU NEED A HINT, TYPE HINT)]
MAKE "NEW.WORD READWORD
IF :NEW.WORD = "HINT [PR [V V] MAKE "NEW.WORD READWORD]
IF TWO.VOWELS? :NEW.WORD [PR [GREAT! SAME PATTERN!]] [TRY.AGAIN]
END
```

(In Terrapin Logo, change READWORD to FIRST REQUEST, take out the outer brackets in the last two lines, and insert ELSE before TRY.AGAIN.)

```
TO TRY.AGAIN
PR [ ]
PRINT [THAT'S ANOTHER PATTERN.]
PRINT [TRY A DIFFERENT WORD.]
PRINT [ ]
PRINT [PRESS ANY KEY....]
IGNORE RC
PATTERNS
END

TO IGNORE :KEY
END
```

I encourage you to write a similar set of procedures that utilize the two-final-consonants pattern. And then, how about plurals? Words that end in silent e? Contractions?

Yes, 'dis spelling *can* be fun.

Judi Harris was an elementary school computer use facilitator, graduate education instructor, and computer consultant for a number of public and private schools in Pennsylvania. She is now a doctoral student in education at the University of Virginia. Her CompuServe electronic mail address is 75116,1207.

MathWorlds

edited by
A. J. (Sandy) Dawson

David Smith teaches mathematics at Duke University and has done so for many years. Recently, however, he spent some time at Benedict College where he worked on the development of an experimental course in Logical Problem Solving for students with very weak backgrounds in mathematics. David has written else where that "...I find my students have had all or most of their natural curiosity trained (beaten?) out of them, especially with regard to anything called *mathematics.*" [1]

Thus, David was obviously pleased when he found that Logo and Turtle Geometry were vehicles through which students could express themselves. Now David uses Logo even in his Duke University calculus courses.

Indeed, he described below how he spontaneously used Logo to illustrate what might otherwise have been impossible; namely, a graphic representation of Archimedes approach to the calculation of π. David reports that his students were impressed by this illustration, but I have the feeling that David was pretty excited about the session as well. Read on, and see if you agree.

Logo Serendipity in the Calculus Lab

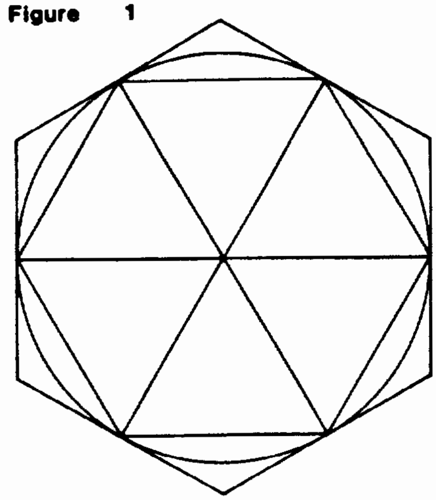
by David A. Smith

I teach a calculus course supported by a "computer lab," which means my section of calculus meets an extra hour each week to discuss computer applications related to the current topic. The students are organized in teams of two and they have a week to work on the assigned lab project. We meet in a room equipped with an IBM PC and overhead monitors. The students do their computer work in open labs at various locations on campus. The classroom computer has a wide variety of software available on a hard disk, and three of the calculus packages have been licensed for loan to the students on floppy disks. The number crunching workhorse is a "general problem solver" called MATHPROGRAM which is essentially a specialized spreadsheet that tabulates recursively defined sequences. It also has some primitive graphics capabilities, and the other two packages (EPIC from Prentice-Hall and Calculus-Pad from Queen's University) provide the students with some very flashy graphics for specific topics in calculus.

In a recent lab session I was presenting the approach of Archimedes to the calculation of π by measuring the perimeters of regular polygons inscribed in and circumscribed about a circle of diameter one (1). This is a lovely visual and numerical example of a limiting process. It is also an example for which recursive definition of the sequences is challenging, but possible, and the explicit definition of the n-th term is essentially impossible. The number of sides of each polygon is doubled at each step. Then, the new inscribed side length is computed from the old, and the circumscribed length from the inscribed one.

The recursive process starts with hexagons, because the side length for the inscribed hexagon is clearly the same as the radius, i.e., 0.5. [See Figure 1: this and the other figures are taken from my book, *Interface: Calculus and the Computer*, Saunders, 2nd ed., 1984.] I had no sooner drawn Figure 1 on the board than I found myself wishing out loud that the computer could draw these pictures for me, because I knew the 12-gon was already beyond my blackboard abilities.

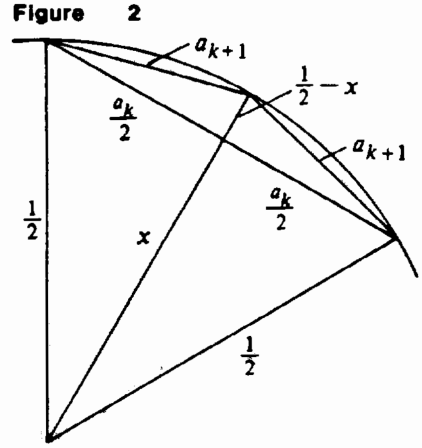
Figure 1



As soon as I said it, I remembered that we had just installed IBM Logo on the hard disk. Clearly, Logo is the right tool for drawing such pictures, not to mention its ability to calculate the approximations to π recursively.

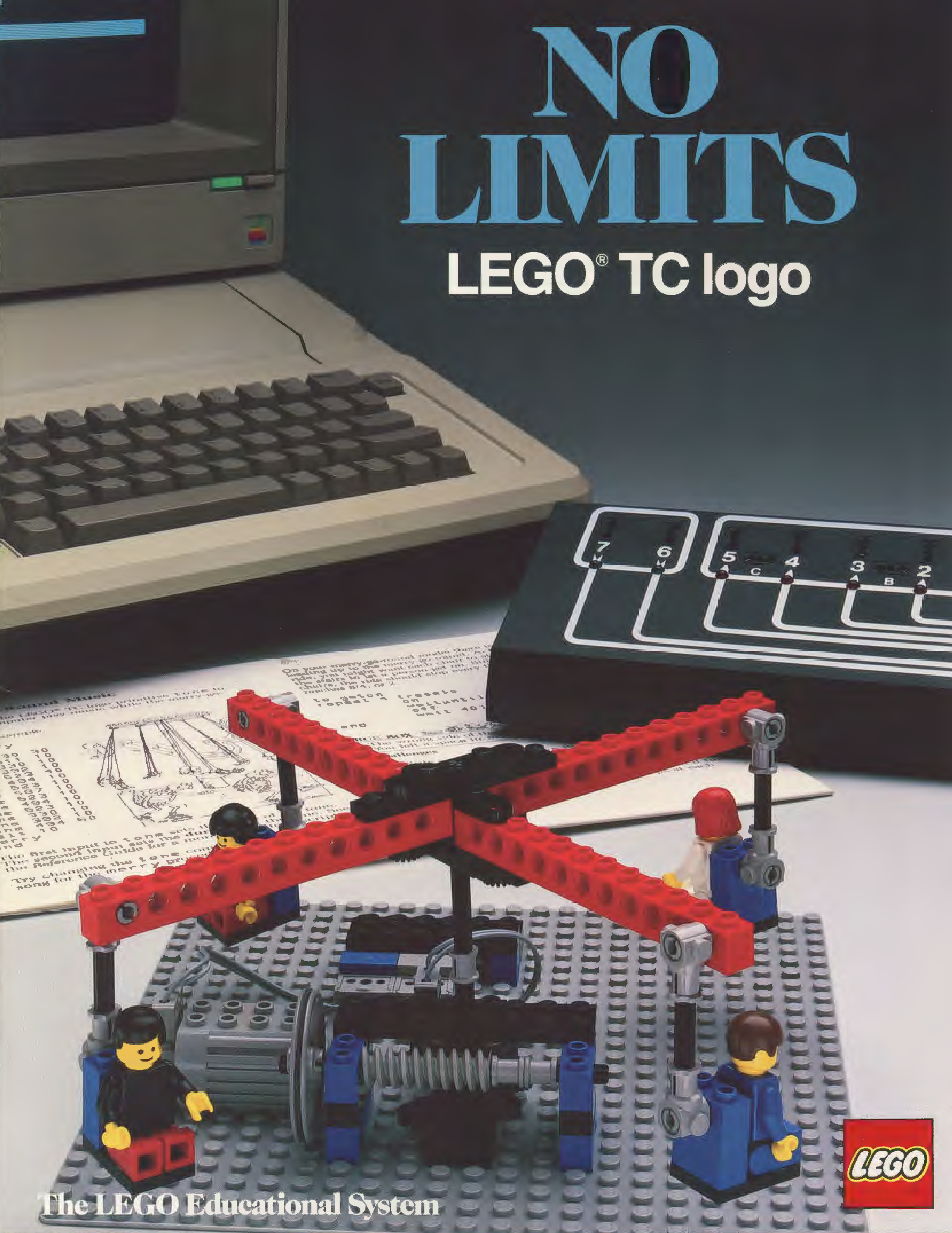
In a matter of minutes I had written the handful of lines required, debugged my mistakes, drawn the pictures I wanted, and had done all the calculations of π for some 1800 years beyond Archimedes. My students are not easily impressed, but I think I impressed them that day. To make my short story even shorter, I will leave out the mistakes I made, the details for the circumscribed polygons, and the algebraic details that may be found in my book. On the other hand, I will add a few refinements that occurred to me right after class.

Figure 2



NO LIMITS

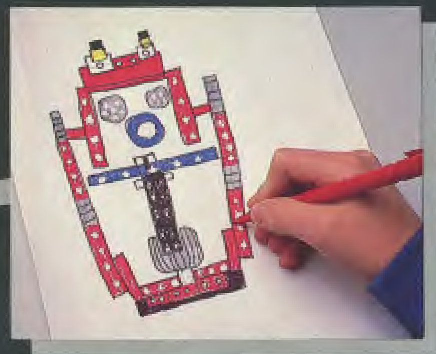
LEGO® TC logo



The LEGO Educational System

LEGO® TC logo

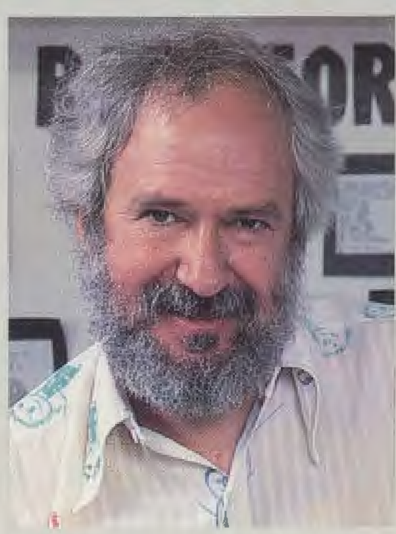
The problem is to design, build and program a machine that will move forward and backward. It will reverse direction every time it runs into an object.



Many different models could be built to carry out this objective. William chooses to build a robot. The first step is designing his model. He draws a rough sketch.

The next step is building his model. How should the gearing and transmission be arranged? Where should he place the sensors so that the robot can "feel" an object it has run into?

PHILOSOPHY



LEGO and Logo are alike in many ways. Of course, there is the obvious: The two names sound almost the same. But there are much deeper connections. In the hands of a child, LEGO and Logo *feel* almost the same. Each is a playful construction set that gives children freedom to explore and create.

Watch a group of children playing with LEGO or Logo and you will see many similar things. In either case, the children are building things. More important, they are building things that they care about. In one case, the building blocks are LEGO bricks; in the other case, the building blocks are Logo procedures. But the results are the same: Children actively involved in building, creating, inventing.

What do children learn from these building experiences? For one thing, they learn certain general skills one might call "project management." That is, they develop a sense of how to plan and execute a complex project. Beyond these general skills, children can learn quite specific principles of science, design, mathematics, and engineering. In working with LEGO, for example, children can gain an understanding of concepts like acceleration, friction, and mechanical advantage.

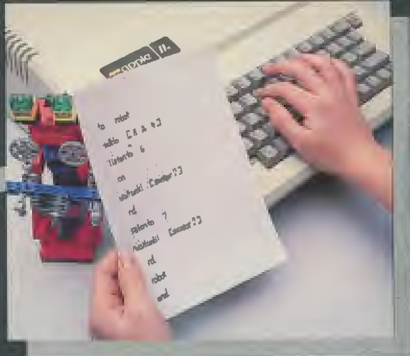
Perhaps most important, LEGO and Logo serve as carriers of some very deep intellectual ideas. Consider, for example, the idea that complexity can emerge from a small collection of simple elements. People have been fascinated with this idea ever since ancient philosophers proposed fire, earth, water, and air as the fundamental elements of the universe. This theme of complexity-from-simplicity plays an important role in the history of physics, mathematics, moral philosophy, and many other fields.

How can children learn such an important idea? Of course, you could try to explain the idea directly. But my feeling is that children must actually experience the thrill of building complex systems from simple elements—as they do in working with LEGO and Logo. It is only through such firsthand experience that children will feel an "ownership" of such important ideas.

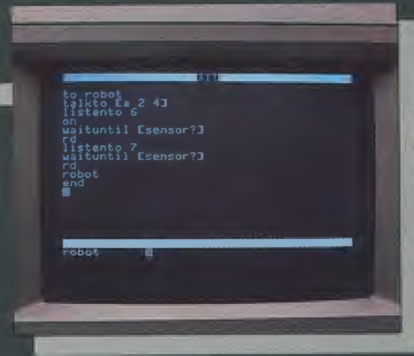
As powerful as LEGO and Logo are individually, they become much more powerful when joined together. As children write Logo programs to control their LEGO constructions, they come in contact with new ideas such as sensors and feedback. Moreover, the two systems reinforce one another: When children have *two* good examples of a general principle, they are much more likely to recognize that there *is* a general principle.

Joined together, LEGO and Logo create one of the richest environments yet for children to explore. Children learn, without ever being told in words, that if you possess powerful elements, the only limit to where you can go is your own desire.

Seymour Papert
Professor of Mathematics and Education
Massachusetts Institute of Technology



The third step is programming. The LEGO® TC logo language makes it clear and easy for William to program his robot.



It will be exciting to see if his robot goes as planned.

If the robot does go as planned, William will write a report giving the details of each step he took in designing, building and programming his model.

If the robot does not go as planned, maybe the model should be redesigned. Or maybe the program should be changed.

Linking the Abstract To the Concrete

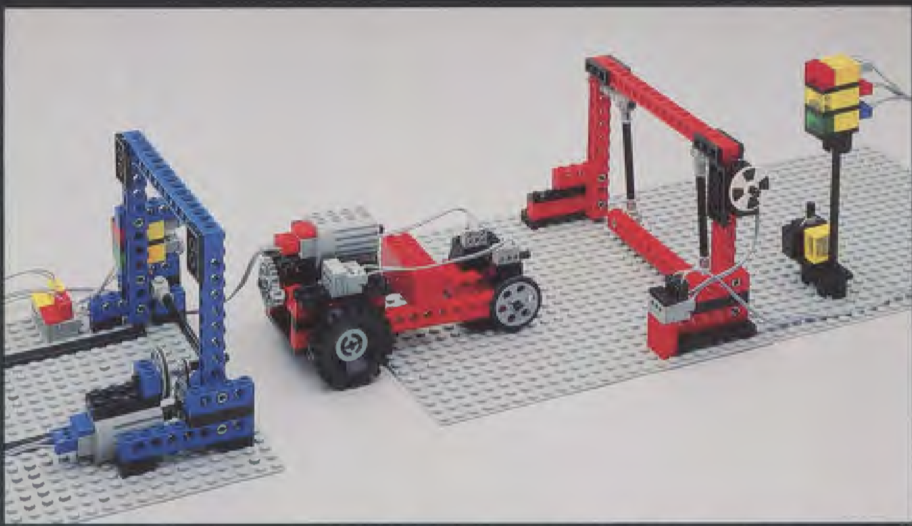
LEGO® TC logo

MODELS TO BUILD

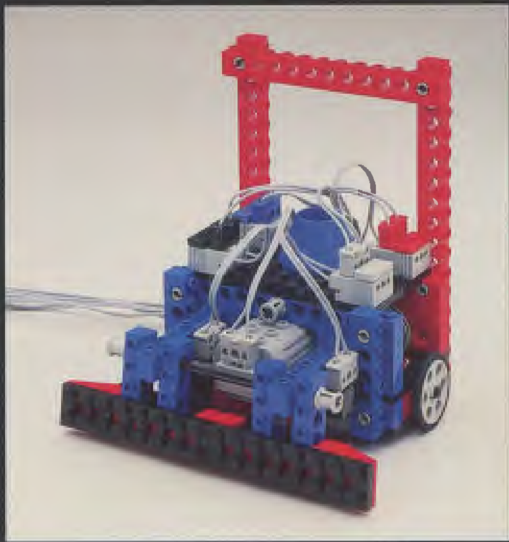
LEGO® TC logo is an interdisciplinary activity involving important ideas from Science, Mathematics and Engineering. While working on their projects, students experience what it is like to actually work as a designer, inventor and engineer. They create new ideas, build prototypes, test them out and modify them to improve performance.

The system is based upon a special interface box through which students can send signals to LEGO motors and receive information from LEGO touch sensors and optosensors.

To program their machines, students use a special version of Logo, using primitives, such as ON, OFF, RD and SENSOR?, so that they can program their machines using natural, intuitive language.

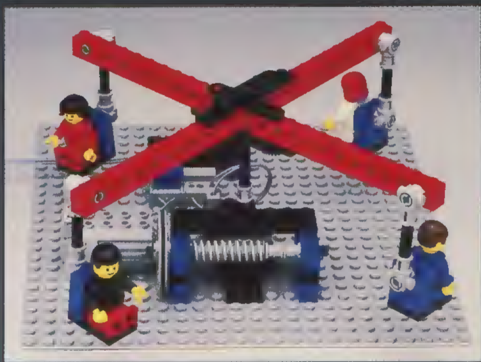


Starting Gate Car Finish Line Traffic Light

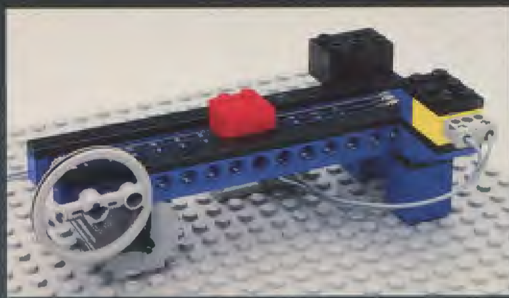


Turtle

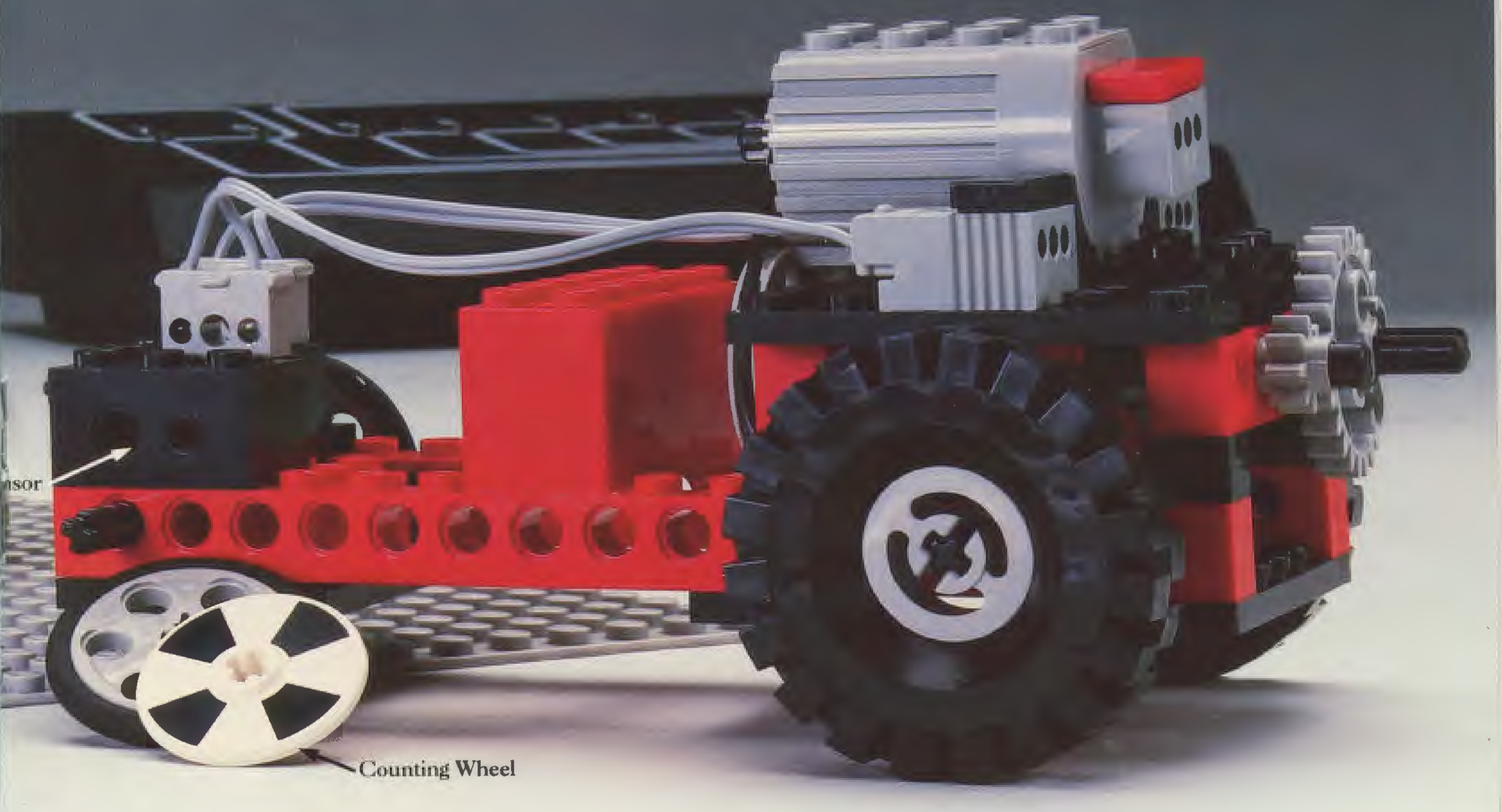
Washing Machine



Merry-Go-Round



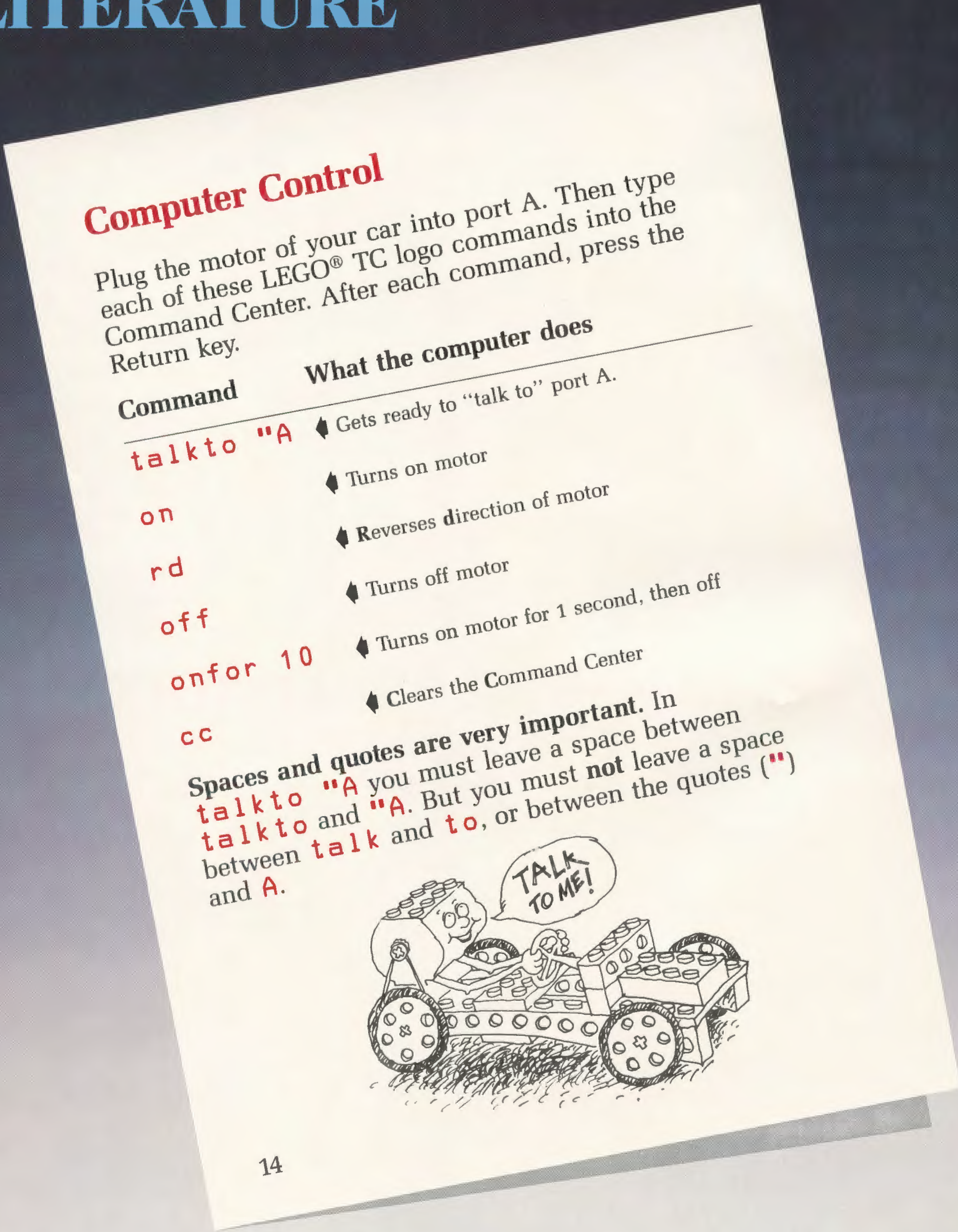
Conveyor Belt



See back cover for description of these special elements.

LEGO® TC logo

LITERATURE



Computer Control

Plug the motor of your car into port A. Then type each of these LEGO® TC logo commands into the Command Center. After each command, press the Return key.

Command	What the computer does
talkto "A	Gets ready to "talk to" port A.
on	Turns on motor
rd	Reverses direction of motor
off	Turns off motor
onfor 10	Turns on motor for 1 second, then off
cc	Clears the Command Center

Spaces and quotes are very important. In talkto "A you must leave a space between talkto and "A. But you must **not** leave a space between talk and to, or between the quotes (") and A.

14

The literature package includes three different Student Guides, one comprehensive Teacher Guide and a Reference Guide.

LEGO® TC logo

ORDERING INFORMATION

9700 Technic Control-0 .... $142.00

Contains over 450 elements and a strong molded storage box. Special elements include 2-4.5 volt motors, 1 optosensor and 1 counting disk, 2 touch sensors, 4 light bricks and 7 sets of building instructions for different models.

9750 Technic Control Interface and Transformer $150.00

Specially designed in conjunction with our research team at M.I.T. This box has 6 outputs from which students can send signals to LEGO motors and 2 input sensors which receive information from the LEGO touch and optic sensors.

950051 Starter Pack $450.00

1 TC-0 Set
Interface and Transformer
Reference Guide
1 Master disk and 1 Backup disk
Quick Reference Guide
Apple IIe® and Apple IIgs® slot card and cable

2x *Getting Started* Student Guide
2x *Making Machines* Student Guide
2x *Teaching the Turtle* Student Guide
1x *Teacher's Guide*

SHIP TO:
Name ______
Title ______
School ______
Street ______
City ______ State ______ Zip ______
Customer's Purchase Order No.: ______

BILL TO:
Name ______
Title ______
School ______
Street ______
City ______ State ______ Zip ______
Order Date: ______ Tel.. ______

10% DISCOUNT FOR ORDERS OVER $1,000

(Offer ends June 15, 1987)

Set No.	DESCRIPTION	Order Qty.	Educ. Price	Ext.
	LEGO TC logo			
950051	Starter Pack		450.00	
9700	Technic Control-O		142.00	
950052	Extra Literature Pack			
950053	Teacher's Guide			
950054	2 Sets of Each (3) Student Guide			
950055	2 Disks, Slot Card, Ref. Guide			
9750	Interface, Transformer		150.00	
1343	2 Optosensors, 2 Counting Wheels		49.00	
1346	2 Touch Sensors		15.00	
1347	Extra Plugs and Leads		13.50	
1348	3 Building Plates		4.25	
1334	2 Technic Motors		19.00	
1335	2 Battery Boxes		7.50	
1336	2 Battery Box Switches		4.00	
1072	300 Extra Bricks		16.29	
1085	1-15″ × 15″ Building Plates		7.32	
1087	6 Small Building Plates		6.00	
	TECHNIC CONTROL			
1090	Technic Control—1		138.00	
1092	Technic Control—2		195.00	

Set No.	DESCRIPTION	Order Qty.	Educ. Price	Ext.
	TECHNIC			
1030	Technic-1, Simple Machines		37.00	
1035	Technic-1, Teacher's Guide		6.00	
1032	Technic-2, Technical Applications		49.99	
1036	Technic-2, Teachers Guide		6.00	
999	Simple Machines Curriculum		14.00	
1034	Technic Teacher's Components Set		150.00	
	Sub Total	$		
	Less 10% Discount, If Applicable	$		
	In Connecticut Add 7½% Sales Tax	$		
	+5% Freight and Handling	$		
	TOTAL PAYMENT	$		

Thank You

Authorized Signature ______
Title ______
Telephone No: ______
Method of Payment: check enclosed ☐ VISA ☐ Mastercard ☐ P.O. ☐
Card No/P.O. No.: ______
Exp. Date:

LEGO® TC logo

SPECIFICATIONS

The 4.5 volt DC motor is actually a 4.0 volt motor, but as it works well on 4.5 volt power supply, it is referred to as 4.5 volt. It uses a current of appr. 1/10 amps without a load and runs at appr. 6.000 r.p.m. When it stalls it uses appr. 1 amp.

The LEGO optosensor is specially developed for use with the LEGO Interface A. It functions as a digital sensor i.e. it measures **changes** between dark and light. With the above feature the sensor has a multitude of applications. It can count revolutions by means of the black/white LEGO counting wheel. It can be placed in front of a LEGO light brick (included in the set) and responds if the light beam is broken. This ability allows the sensor to function as a switch and it is built into a LEGO brick.

The LEGO touch sensor is also specially developed for use with LEGO Interface A. The touch sensor consists of a touch-sensor brick and a small axle. The axle acts as a pushbutton. If something is pressing the button, the sensor reports "true". If nothing is pressing it, it reports "false".

The LEGO Interface A connects to a special slot card that fits the Apple IIe® and Apple IIgs® computers. A power supply transformer is included in the set. The interface and computer are fully protected against wrong connections of motors or sensors. The interface has a stop button which cuts off power to all output sockets. It also withstands normal wear and tear from fingers and dust.

IN-SERVICE TRAINING

For product information, up-to-date availability and to find out about our Teacher Training In-Service Program, please write or call:

Sarah McDonald
Education Dept.
LEGO Systems, Inc.
555 Taylor Road
Enfield, CT 06082

203/749-2291
800/243-4870 (Ex. AK, CT, HI)

Please route this brochure to ALL teachers of Computer Studies, Science and Math in your school. **Thank You.**

Teacher's Name	Room Number	Check Off
		☐
		☐
		☐
		☐
		☐

The key formula is derived from Figure 2 in which a_k represents the inscribed side length at step k, and a_{k+1} represents the next term of the sequence. Application of the Pythagorean Theorem to both large and small triangles (to eliminate x) suffices to establish the formula:

$$(a_{k+1})^2 = \frac{(1 - \sqrt{1 - a_k^2})}{2}$$

from which a_{k+1} can be expressed in terms of a_k by taking the square root of each side. However, this formula turns out to be a computational disaster beyond very small values of k. When a_k gets small, $1 - a_k^2$ is very close to 1, and its square root is still closer to 1. Subtraction from 1 then leads to heavy loss of significant digits, i.e., a large relative error. That error (in a number much less than 1) is then magnified by the subsequent square root step and further magnified by multiplication by a large number of sides. The upshot is that there is no point in going beyond the sixth step; the approximation to π never gets better than four correct decimal places.

While the formula may be unfamiliar, the loss-of-significance problem is very common. The solution is simple, but unfortunately it is not commonly taught--if anything, our secondary curriculum treats it as *heresy*. I refer to rationalization of the *numerator* (which is also the solution of a wide range of problems in calculus, from differentiation of the square root function to evaluation of various limits and integrals). When we apply it to the formula above, we get:

$$(a_{k+1})^2 = \frac{a_k^2}{(2 + 2\sqrt{1 - a_k^2})}$$

Now the only subtraction is of quantities very different in size, and the side lengths can be computed quite accurately.

To write the graphical procedures we need one more formula. We have to decide how to turn the turtle after each polygon is drawn so the next one will be inscribed in the same circle. Figure 2 shows us what to do here. At the k-th step the turtle will end up pointing along a side of length a_k. To start the next polygon, we have to turn the turtle to the left by an angle whose cosine is $a_k / 2a_{k+1}$.

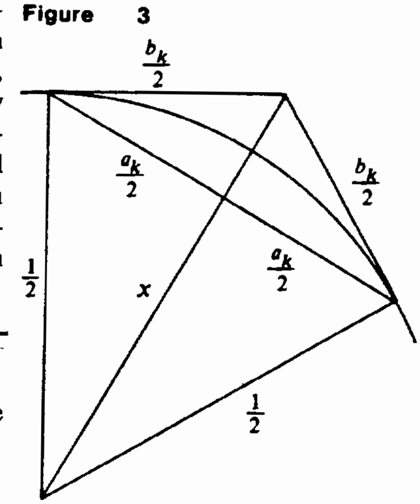
Here is what appeared on my edit screen:

```
TO SEQUENCE :N :A
MAKE "NEWA :A / SQRT 2 + 2 * SQRT 1 - :A * :A
POLYGON :N 200 * :A
LEFT ARCCOS :A / (2 * :NEWA )
PRINT SENTENCE :N * :A :N
SEQUENCE 2 * :N :NEWA
END

TO POLYGON :N :S
REPEAT :N [ FORWARD :S RIGHT 360 / :N]
END

TO ARCCOS :X
OUTPUT ARCTAN (SQRT 1 - :X * :X ) / :X
END

TO START
CS PU SETPOS [ -100 -30] PD
SEQUENCE 6 0.5
END
```

The factor of 200 converts numerical side lengths to turtle steps. The print line gives the perimeter of the polygon just drawn and the number of its sides. The need to define ARCCOS provides an excellent opportunity to point out that all the inverse trigonometric functions can easily be defined from a single one. (IBM Logo provides ARCTAN.) The START procedure positions the turtle to use as much of the screen as possible and starts the recursive SEQUENCE. Only the polygons of 6, 12, and 24 sides are distinguishable on the screen. Thereafter, the impression of doubling the number of sides is created by the time it takes the turtle to draw what looks an awful lot like a circle. If you are willing to wait long enough, at the 24th step (12,582,912 sides), Logo reports the perimeter as 3.141592654, which is the correct (rounded) value to the ten digits shown.

Actually, there is no point in waiting for all those very slow, visually identical circles. Once the geometric point is made, the graphics commands in START and SEQUENCE can be deleted, and Logo can quickly calculate the perimeters for as many sides as you like. Beyond the 24th step you quickly discover the effect of the (default) 10 place accuracy. Subsequent output values creep slowly away from π. However, this is a good opportunity to experiment with IBM Logo's SETPRECISION command which permits up to 1000 significant figures. If you set the precision at 40 places, you find that the same SEQUENCE (without the graphics) will produce 40 accurate places of π with about two minutes of calculation time (and about 1.6×10^{35} sides). Newer and more powerful implementations of Logo are laying to rest the criticism that it's not a scientific number crunching language.

Figure 3



I leave the circumscribed polygons as an exercise for the reader, with a hint provided by Figure 3. Using the similarity of large and small triangles in the figure, you can show that the circumscribed side length is given by

$$b_k = \frac{a_k}{\sqrt{1 - a_k^2}}$$

From there you are on your own.

David Smith
Department of Mathematics
Duke University
Durham, NC 27706

1. David A. Smith (1986) "It's Showtime." *Collegiate Microcomputer* , August, Vol. IV, No. 3.

A. J. (Sandy) Dawson is a member of the Faculty of Education at Simon Fraser University in Vancouver, British Columbia. His CompuServe number is 76475,1315.

SpecialTalk

by Paula Cochran and Glen Bull

Type, Look, and Listen: Talking Word Processors

> Once upon a time . . . there was a talking computer. His name was TC. TC learned to read out loud whenever Sally or the other children typed on his keyboard. Each time Sally pressed a letter, TC would say the letter's name. When Sally pressed the spacebar TC would know that the word was finished, so he would read the whole word out loud. He would read a whole page, too, if Sally wanted him to. He moved the cursor from left to right as he read, so that Sally could follow along. Sally liked it when TC talked to her, and he didn't seem to mind reading her writing and over and over again . . .

For some readers, this scenario may seem a bit far-fetched -- a talking computer under the control of a child? Well, if you haven't experienced a talking word processor yet, you are in for a treat. You may recall that in the January "Special Talk" column we provided some general information about making Logo talk, and sample procedures in several Logo dialects. If you have tried using speech output with your students since then, chances are you are already pretty enthusiastic about the effect of a "talking computer" in the classroom. If you haven't had a chance to try it yet, or are hesitating because you are a Logo beginner, a talking word processor may be a good way to introduce synthetic speech into your classroom or clinic.

Alternative to Drill and Practice

Our personal experience is confirmed by recent surveys that suggest that special educators and clinicians tend to use a computer more for drill-and-practice applications than for other kinds of instructional activities (Mokros and Russell, 1986). Although word processing is also a frequent application with older children, teachers seem to fall back on pre-programmed lessons for younger children with special needs.

We think that the current availability of good talking word processor software for the major brands of computers used in education presents an appealing alternative. Talking word processing seems Logo-like in its flexibility and in the sense of control which is conveyed to the user. Unlike silent word processing, it also seems to lend itself to activities involving more than one child at a time.

For example, the talking word processor can be used as the basis for a show-and-tell period in which the emphasis is on the telling. Sitting in a small group around the computer, children are encouraged to tell about a recent experience or special event that happened in their home or school. Each contribution is typed into the computer either by the student or a teacher. Later in the day, children can return to the computer, press a key, and hear their sentences read over and over again. Their stories can easily be printed out and sent home with them.

One of the teachers who participated in the UVA/IBM Institute in Charlottesville last summer is Bonnie Nelson. Bonnie teaches a non-categorical (multiply-handicapped) pre-school class at Mountain View School in Roanoke, Virginia. Here are excerpts from the Morning News in her class, written with a talking word processor called Listen to Learn (we have changed the names of her students).

> Morning News
>
> Timmy: I played with a box. My granddad got it for me.
>
> Sally: That cat is back. She wants to borrow our hamster.
>
> Susan: For Christmas I want a real baby with diapers. Also I want another one too that doesn't make so much mess.
>
> Ben: I want a Teddy Ruxpin . . .
>
> Peter does not want a Teddy Ruxpin. Peter was sick but is feeling better.
>
> Ben (typing his own news): rrrrrrrrrrrrrrrrrrrr
> BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
> NNNNNNNNNNNNNNNNNNNNNNNN
> eeeeeeeeeeeeeeeeeeeee BEN

Using Talking Word Processors

The special populations who may benefit from the use of a talking word processor include beginning and pre-readers, learning disabled, adult brain-injured, and speech/language impaired. In education settings, goals may be related to developing literacy skills (Rosegrant, 1984, 1987). For example, early sound-symbol relationships can be acquired as the computer repeats the name of each letter the child types (over and over, if desired).

There are many other academic and communication skills that can be facilitated when speech output is added to a program under the user's control (Cochran & Bull, 1987; Meyers, 1984; Rosegrant, 1984) Older children can write and read stories for each other that contain target speech sounds, or young language delayed students could plan messages for each other with the help of the teacher or clinician. An adult aphasic may need to regain reading or writing skills after a stroke. Using a talking word processor, he can receive aural and visual feedback as he writes a shopping list. Nick Lape, director of speech and hearing services for the Southside Virginia Training Center for the Mentally Retarded in Petersburg, Virginia, used a talking word processor to help a mentally retarded, non-oral adult write a personal statement for his annual review conference.

It Sounds Good, But . . .

In our enthusiasm, we are in danger of hyping talking word processors the way that Logo was hyped a few years ago. It goes without saying, we hope, that this genre of software has not been perfected, and that it will not be appropriate or successful in every situation. BUT, it has also been our experience, and the experience of the teachers and clinicians with whom we work, that many initial questions and concerns about talking software are easily resolved.

For example, we are often asked why we are not worried that children will imitate the less-than-perfect speech of a low-cost speech synthesizer. While it is sometimes the case that students do so in jest, so far we know of no instance in which a child seriously adopted the computer as a speech model.

It is true that many speech synthesizers are difficult to understand, especially at first. It has been fairly unusual, however, to find a student or client who had persistent difficulty comprehending the speech of the synthesizers commonly used with educational software. This does not mean that we should be satisfied with the quality of low-cost speech output that is currently available -- just that teachers and clinicians may find that intelligibility is less of a barrier than they supposed.

What about distraction to other students in the classroom? This is a legitimate concern -- in most instances, a system with a headphone jack is available, so that students working independently need not disturb others.

We think that you will find speech output a welcome addition to some of the computer activities you already use with your students. Having a computer that says whatever he types seems to give the user a sense of control, and a sense of audience. He decides when his message is correct and complete -- and he gets to hear it the way another listener would. In some instances, students can use talking word processors independently, completing self-initiated projects or assignments suggested by the teacher. In other cases, a talking word processor can be used as the basis of an interactive learning activity, between the computer, the teacher, and one or more students together.

For More Information

Talking word processors have various strengths and weaknesses, depending on the population and the computer for which they were developed. Features to consider include ability to choose size and color of text on the screen, brand of speech synthesizer, speed and mode (letter, word, page) of speech output, and ease of use for a particular population. For more information about particular software and hardware, you may want to contact the following:

Key Talk (Apple)
PEAL Software
3200 Wilshire Blvd.
1207 South Tower
Los Angeles, CA 90010

Kidtalk (Macintosh)
First Byte
2845 Temple Ave.
Long Beach, CA 90806

Listen to Learn (IBM)
IBM Corporation
P.O. Box 1328-W
Boca Raton, FL 33429

Talking Text Writer (Apple)
Scholastic, Inc.
730 Broadway
New York, NY 10003

Word Talk (IBM or Apple)
Computer Aids Corporation
124 West Washington, Lower Arcade
Fort Wayne, IN 46802

References

Cochran, P., & Bull, G. (January, 1987). SpecialTalk: Untrapping intelligence. *Logo Exchange*, pp. 15-17.

Meyers, L.F. (1984). Unique contributions of microcomputers to language intervention with handicapped children. *Seminars in Speech and Language*, 5 (1), 23-33.

Mokros, J.R., & Russell, S.J. (1986). Learner-centered software: A survey of microcomputer use with special needs students. *Journal of Learning Disabilities*, 19(3), 185-190.

Rosegrant, T.J. (1984). Use of microprocessors to remediate speech through literacy. In W.H. Perkins (Ed.), *Current therapy of communication disorders: Language handicaps in children* (pp. 57-60). New York: Thieme-Stratton.

Rosegrant, T.J. (1987, April). *Exploring the algorithms of written English.* A paper presented at the East Coast Logo Conference, Alexandria, VA.

Paula Cochran is an assistant professor in the Communication Disorders Program of the University of Virginia's Curry School of Education. She is interested in Logo applications for language arts and special populations. Glen Bull is a professor in the University of Virginia's Curry School of Education, and teaches Logo courses at both the graduate and undergraduate level. His CompuServe number is 72477,1637.

Life Logo Available

The Life Logo mentioned in Brian Silverman's article in this issue (see page 3) is available for purchase. Called the "Phantom Fishtank," the program with manual costs $29.95 plus $1.50 shipping and handling. Order from Logo Computer Systems Inc., 121 Mount Vernon Street, Boston, MA 02108.

Testudinal Testimony

by Douglas H. Clements

Knowing All the Angles: Part 1 Research on Angle Concepts

If It's Not Achievement, What Is It?

Last month we saw that students appear to use mathematical concepts in Logo programming. However, results of studies examining the effects of working with Logo on standardized mathematics tests were inconsistent. We discussed possible reasons, such as limited exposure to Logo, limited number of students in some studies, and a lack of correspondence between what appears to being learned in Logo and items on standardized achievement tests. Closer examination of this final reason reveals a fundamental misconception that is held by many researchers and teachers.

Logo was originally developed to serve as a conceptual framework for learning subject-matter content. Much of the literature on Logo has (implicitly or explicitly) presumed that straightforward *practice* with such notions within the context of Logo programming would increase achievement. As we know, this cannot be guaranteed.

It may be that Logo does provide practice within limited topical areas and in constrained contexts. If this were "the whole story," direct provision of practice through CAI drill would probably be more effective and efficient. However, unlike CAI, Logo—used as a *framework*—does not just teach the mathematics curriculum directly. It provides students with opportunities for directly manipulating certain mathematical objects such as angles, numbers, and variables. They can reflect upon those manipulations and their effects. This may permit them to build the mental structures that represent their knowledge about those objects. Most students need to experiment with and reflect on mathematical objects to construct their own mathematical structures, or schemas (Davis, 1984).

But wait a minute. Disregarding the fancier language, what is the difference between "achievement," as measured by traditional tests, and these "mental structures"? On the one hand, nothing. Existing structures do determine how students respond on tests. On the other hand, however, these tests usually measure only the surface layer of the structures — basically, rote knowledge. We all know students who can respond correctly to items on such tests but whose ability to understand and apply the material is weak (i.e., their structures are "shallow"). But how then can we ever tap these structures? It's difficult, but possible. Variations on Piaget's clinical interviews represent one method, as we shall see. But first, one more pertinent word from Piaget.

One of Piaget's strongest findings was that the abstraction of geometric form from real objects is *not* based on perception. It is based on the *actions students perform on the objects* (Piaget & Inhelder, 1948/1967). Thus, in forming paths by walking and then directing the Logo turtle, children are performing actions that may build knowledge about the figures formed by these paths. Because turns play a central role in forming shapes such as polygons, forming such figures in a Logo environment may focus students' attention on this aspect of their actions in constructing the shapes. This may enhance their understanding of the Euclidean notions of angles and rotations. In this spirit, this month's column and the next will take an in-depth look at students' ideas about angles and angle measurement.

How Many Angles in a Triangle, Again?

Postulating that the effects of Logo as a framework may be delayed rather than immediate, I sought out children to whom a colleague and I had taught Logo several years before. After working with Logo, these children had made significant gains in such areas as creativity and problem solving compared to a control group (Clements & Gullo, 1984).

Did the experience affect their subsequent learning of mathematics? To find out, I first looked at their achievement scores (note that only 7 of the control and 9 of the Logo children were still in the school system). The Logo group scored somewhat higher than the control group, but not overwhelmingly so (Clements, 1987b). I thought that the difference might have stemmed from certain types of items. Therefore, I examined any items on which the groups scores differed by more than 20%. To my delight, both geometry items fell in that category. To my consternation, it was the *control* group that scored more than 20% higher! I interviewed the children on these items to figure out what was happening.

One of these items asked, "How many angles does a triangle have?" Both on the test and in the interview, six of the seven control children answered, "Three." Their explanations were similar: "I just knew." However, five of them, when asked to "show me the angles," indicated the sides. When told, "Those are the sides," three pointed to the intersection of two sides and asked "Here?" The other two stated that they did not know what an angle was. The sixth child identified the vertices. When asked, "What are angles?" she queried in return, "Points?"

The seventh child demonstrated a slightly different conception. She responded, "Two. No, one."

Examiner: "Where would they be?" (drawing a triangle).

"This is the same as the others" (pointing to the side nearest to her).

E: "So how many would there be?"

"Three."

E: "Is that what an angle would be, this (pointing to the sides), this, and this?"

"If they're not straight, like."

E: "So if we drew it like this..." (notices that his first triangle was drawn with a horizontal base from his perspective; draws another triangle with the horizontal base from the child's perspective) "it would be...."

"This one and this one."

E: "How about this one?" (pointing to the base).

"No."

Thus, her concept of angle was of a "slanted" line segment, "tilted" from her egocentric frame of reference.

Three of the Logo students similarly answered "three," yet indicated the sides when asked to "Show me the angles." When told those were sides, they identified the intersections as the angles. Two children answered "two" and identified the "slanted" sides which were not parallel to any side of the paper. When asked to "draw an angle," they both drew a single, slanted line segment.

The sixth child originally said "Easy. Three sides."

E: "Not sides. Angles."

"Oh. They're little parts."

E: "Little parts? Show me."

"Here." (He draws two small line segments at an intersection.) "The 'V' part."

E: How many of those?

"Six" (points to the line in each "V" angle as he draws it).

Three Logo children asked a question immediately. One (SC) asked, "What do you mean...corners?"

E: "What do you think it means?"

"This" (pointing to a vertex angle).

E: "Then how many would a triangle have?"

"Three."

E: "What else could it mean?"

"The sides."

E: "How many would there be?"

"Three."

Two girls asked, "What are angles?" When asked, "What do you think angles are?," the first said, "I was thinking of corners"; the second, "It could be where two parts come together." In response to "How many would there be in a triangle, then?," they each said, "Three"

At the close of the interview, the Logo children were asked, "Do you remember when you used to make the turtle draw triangles? How did you do it?" Six out of nine proffered an "adequate" program. Not surprisingly, however, the number of degrees given as input to the rotation commands was often incorrect. Responses ranged from RT 45 FD 50 RT 90 FD 50 RT 90 FD 50, to a written REPEAT 3 [RT 45 FD 50], to a verbally delivered, "Repeat three, bracket, forward fifty, 'R' 'T' three-sixty, divided by three, bracket." (This last child stated that "I try to get Logo when I can" -- apparently utilizing a neighbor as his supplier.) Two children gave appropriate, but less specific, process descriptions, such as "You would turn right 50, then forward, then left, then forward, then left, then forward, then that's it". The last child could not remember.

The children were then asked, "How does the turtle make an angle?" Responses included: "The turn, like right or left," "'R' 'T' or 'L' 'T'," "Turns degrees," "By going straight, then a turn," and "The rights." Asked, "How many angles would be in the turtle's triangle?" seven of the Logo children stated the same answer they gave to the original question. Examples included: "Uh, turtle triangle would have two slanty sides," and "Six" (the "V" angles), as well as the expected "three". The girl who was "thinking of corners" changed her answer for the turtle's triangle: "Four. That (RT 45) when I turned before going forward" (i.e., an initial rotation so that the first side drawn is a "slanted" line segment). The final child's (SC) response illustrates again how the "turns" in Logo may be confused with the "angles in a triangle." It also illustrates how a child's work in Logo may remain disconnected with other aspects of his or her mathematical knowledge. SC had asked, "What do you mean 'angle'...corners?" Later, when asked how he had the turtle draw a triangle, he said:

"We started and then I put left and I made it turn so it would be like that (rotating hand). Then I made it go forward so it would go like that and then I made it turn on an angle. An angle (shouting and laughing)! A turn! A turn...the same thing." (Pause).

E: "That's neat, huh? That's the same thing! So how many angles would the turtle have to make to do it?"

"Well, if he started sideways, does that count? Or should I..."

E: "Start with the turtle straight up, but you could make the triangle like this (drawing first side vertically) or like this (drawing first side at a 30 to 45 degree angle). But how many angles?"

"Four?"

E: "Show me." (SC indicates the original rotation, then the three vertex angles.)

Certainly, the "angle" schema was not well established for most of these third graders. Most interpreted the term as a synonym for "side." Other half-formed concepts included notions of "slanted lines," "points" or "where the two parts meet," and "two short intersecting lines" (the "V" angle). Just as important, there is evidence that work with Logo influenced children's construction of angle schema, but possibly to the *detriment* of their test performance. On the test, two children answered "four." One, SC, answered "three" to the original question in the interview, but stated that the turtle would have to make four angles.

Proportionally more control than Logo children stated the correct answer. This indicates more awareness, albeit imperfect awareness, of the angle concept for the Logo children. There was no evidence, however, that the Logo children's concept of angle included a dynamic, or process, component. They identified the rotation commands as responsible for the construction of angles, but did not initially connect the angle concept with notions of rotation or measure. Prompting was needed to help them establish such a connection.

The child who said that "the rights" make angles was asked, "What are angles?" He replied, "I don't know. I guess they're like the turns...the rights." SC, of course, came upon the connection in the context of recalling turtle commands. Although this realization appeared satisfying, it is clear that his previous use of the term "angle" within the Logo context had been limited to that context. Even given the realization, the notions of "turn" and "angles in a triangle" remained undifferentiated.

Different Angles on the Angle Concept

To determine whether such findings were reliable, I worked with 24 third grade students for 26 weeks with Logo, three sessions per week (Clements, 1987a). Then I interviewed them, along with 24 students who formed a control group (random assignment was used and beginning year mathematics achievement scores were virtually identical). One set of interview questions involved concepts of angle.

Responses to an open-ended question, "What is an angle?" were categorized but not scored. More control children demonstrated either little knowledge of angle (i.e., "I don't know" or "it's a line") or the influence of common usage (e.g., "a diagonal line"). Logo experience appeared to affect children's notion of angle as rotation, with 9 Logo children (compared to 0 control children) reflecting mainly that conceptualization (e.g., "An angle is a turn. Sort of a line from a turn").

More control students said that a angle is a "corner." More Logo children identified an angle as the union of two segments (e.g., "When two lines meet each other and they come from two different ways"). To the extent that these two categories are similar, there was little difference between the groups. The former, however, may be a more constrained context. Often, children's responses indicated that their "corner" must be part of a figure. In addition, for many, "corner" implied perpendicularity. Thus, Logo students appeared more likely to generalize the concept of angle from the more limited "corner" context to that of the union or intersection of two segments (no student mentioned rays). One Logo child explicitly distinguished between several conceptualizations; "Do you mean like an angle in math or like the other kind, like at an angle?" Several reflected a process orientation; for example, "It's like a straight line, and then another line that goes off like this after a turn" (accompanied by gestures).

All other angle items were scored. The Logo group performed significantly better than the control group on the total score. Students responses on individual items were revealing. Logo students performed well selecting "those figures that are angles." However, they did show some misconceptions.[1] For example, two students only selected right angles. One indicated the diagonal segments and non-obtuse angles (obtuse angles were, for her, not "corners" which had been her definition for angle). Two additional misconceptions helped by Logo children were: angles included (a) lines "slanted from straight up" (i.e., both diagonal and horizontal lines; note that drawing these in Logo would require a rotation given the turtle in its home state); and (b) "curved angles" (the union of two arcs).

1. Possibly a more appropriate phrase would be alternate conceptualizations. Children's ideas were often quite valid in their own terms. The word "misconceptions" will be used (a) to highlight differences between children's ideas and traditional mathematical concepts and (b) for ease of expression.

Performance for the control students was lower than that of the Logo group. Misconceptions including choosing: *all* lines, slanted lines only, angles and slanted lines, non-obtuse angles only, and right angles only.

One responded according to the following rules: Perpendicular segments are "corners"; horizontal and vertical segments are "up and down" and "sideways" lines; all other straight lines, alone or in combination, were "angles." One girl contemplated a long time before deciding that the obtuse angle should be selected ("I'm debating this one"). Regarding the right angle (whose horizontal segment was to the left of its vertical segment), she asked, "Can they be left angles or right angles?"

Another was asked why she did not select the two non-intersecting segments. She explained, "Because you can't fit a square piece of paper in it." She then looked at the other non-right angles that she had already selected and said, "Oh, no, it's because they don't close." A third, whose previous definition of an angle was, "When two lines come together," similarly indicated that the curved lines were an angle only after an internal debate. In justification, she said, "It looks like a semicircle almost, but there are two lines coming together." Thus, her own definition may have been in conflict with her intuition, and finally was more influential.

There were several items that asked, "How many angles in this figure." On one such item (the figure was an "X"), students made such errors as indicating the intersection only , one set of vertical angles only, and that there was no angle in the figure. In each case, the Logo group made fewer of errors. The response of two Logo children who did respond "zero" was interesting. One girl said, "They're all straight lines. None of them turn." I asked, "What about if you made them by turning this way" (tracing an angle). She responded, "Two." Another child agreed that, "When you turn when coming down on a straight line, *that's* an angle (therefore, the figure had no angles, but would have two "if you had turned"). One control child had a different notion: If it were "two halves of two triangles, it would be two angles. If it was an 'X', it wouldn't be any angles." Thus, the *way* a figure was constructed was important to these children in determining its nature—in deciding whether or not it was an "angle."

Some Conclusions

Several conclusions emerged from these and other items: (a) children's conceptualizations of angle are complex, reflecting the influence of societal usage, school experiences, and effects of the Logo experience (with possibly the least effect attributable to textbook definitions); (b) these conceptualizations are not well discriminated from related concepts such as orientation, corner, rotation, left, right, etc.; (c) children's responses frequently vary with the task (sometimes taking a completely different "angle" on angle); and (d) while the Logo experience had by no means eliminated "errors," it appeared to have had significantly affected children's ideas about angles (i.e., their "angle schemas").

Several other researchers report that Logo impacts students' conceptualizations. For example, Lehrer & Smith (1986) asked their Logo students, "What do we mean by the word angle?" Most (75%) of the students' responses reflected a Logo-based conceptualization, despite their classroom experiences that defined angle as the intersection of two rays. The authors maintain that Logo's procedurally-based definition was the more salient of the two.

None of Kieran's (1986a, 1986b) intermediate grade students knew what an angle was when first interviewed. After informal work with Logo, about half thought that they knew what an angle was. Half of those referred to corners of objects, the other half involved some reference to surfaces and slanted lines. Most of Kieran's excellent work (as well as that of several other researchers) has involved students' notions of angle *measure*. This will be our "jumping off place" next month. We'll also summarize all the results at that time. For now, think about what kind of multifaceted and divergent ideas about angles your students have. How might this influence your teaching?

References

Clements, D. H. (1987a). *Learning of geometric concepts in a Logo environment.* Manuscript submitted for publication.

Clements, D. H. (1987b). Longitudinal study of the effects of Logo programming on cognitive abilities and achievement. *Journal of Educational Computing Research*, 3, 77-98.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76, 1051-1058.

Davis, R. B. (1984). *Learning mathematics: The cognitive science approach to mathematics education.* Norwood, NJ: Ablex Publishing Co.

Kieran, C. (1986a). Logo and the notion of angle among fourth and sixth grade children. In the *Proceedings of PME* 10 (pp. 99-104). London, England: City University.

Kieran, C. (1986b). Turns and angles: What develops in Logo? In G. Lappan (Ed.), *Proceedings of the Eighth Annual PME-NA.* Lansing, MI: Michigan State University.

Lehrer, R., & Smith, P. C. (1986, April). Logo learning: Are two heads better than one? Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

Piaget, J., & Inhelder, B. (1948/1967). *The child's conception of space.* New York: W. W. Norton.

Douglas H. Clements is an associate professor in the College of Education at Kent State University in Kent, OH. He teaches Logo courses and conducts research on the use of Logo to develop cognitive, mathematical, and social-emotional competencies. His CompuServe number is 76136,2027.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

The results of a recent study suggest that Logo experiences may serve as a beneficial prerequisite for learning to program in Pascal. R. D. Sweetland of the University of Nebraska investigated the extent to which a class of sixth grade students with two years of Logo experience could learn to program in Pascal. These students, five girls and six boys, made up the entire sixth grade of a K-12 school system with 200 students. The school is located in a small rural village in the Midwestern United States.

Each student had studied Logo as fourth and fifth graders. They had learned how to draw designs, write procedures, save files, and print copies of their work. They had been introduced to variables, list processing, arithmetical operations, REPEAT, and IF-THEN-ELSE statements. As fifth graders, they were required to use superprocedures in all assignments. As sixth graders, they were assigned to the computer for one hour three times each week. About half of this time was spent in instruction and planning and about half working alone on the computer.

Learning activities were designed for active involvement by students with opportunities to think about what they were doing. A learning cycle of exploration, invention, and discovery preceded interaction with the computer. Exploration activities were provided so that students could learn the material and be able to use it during invention and discovery. Invention was accomplished when the student could completely describe the use of the computer statements or algorithms. During the discovery phase, the student had to use what had been invented to discover new applications and solve problems. Questions patterned after the Polya model (Polya, G. (1957). *How to Solve It.* Princeton: Princeton University Press.) were used to help students during this phase.

Data collected over the one year period included records from students' notebooks, diaries, attitude surveys, projects, worksheets, quizzes, and tests. Teacher observation of students' interaction with the computer, diary notes, software choices, and frequency of student work before class were the primary means of measuring affective changes. The evaluation of this study includes a case study of each student, a discussion of the programming achievements of the class, an external evaluation of the students, and recommendations for the future use of Pascal.

The external evaluator administered three problems from the final exam for the Pascal course for non-science freshmen students enrolled in a university physics course. The sixth graders' responses to the three questions were compared with those of 32 college students.

The results indicate that all students were able to program in Pascal. These sixth graders programmed as well as college freshmen on two of the three exam questions and better than the freshmen on the third question. Students were able to master 73% of their course objectives and their attitude toward programming improved. A specific breakdown of the mastery of objectives includes the following:

Use the operating system	87%
Write simple Pascal programs	80%
Develop debugging strategies	78%
Use variables	83%
Use iteration	89%
Input / Output data	100%
Use branching	100%
Write programs with two or more procedures	100%
Develop advanced programming techniques	55%
Use mathematical operations	100%
Write an animation program	91%

The researcher recommends additional research to determine if these results can be replicated with larger groups and to fill the void of studies in this area. Curriculum planners may now take a closer look at a computer curriculum which begins with Logo and moves to Pascal. Each case study describes the student, presents a selected profile of the student's work, and a summary of the student's learning style. In terms of the achievement of objectives for the year, students performed at the level of mastery on 73% of the course objectives.

>>>Sweetland, Robert D. (1986). *Learning Pascal in a Sixth Grade Classroom: After Logo Experiences.* (Doctoral Dissertation), University of Nebraska, 1986.

Barbara Elias is an assistant professor in the School of Education of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

LXionary

A Lectionary of Selected Logo Readings

with Commentary and Opinion

by Bill Craig

>>>"The Right Way to Use Computers" by Beth Lowd, *The Computing Teacher,* October, 1986

Beth Lowd has identified five goals that advocates of the use of computers in education are trying to achieve:

1) Make education "teacher proof," make sure each student has mastered each step towards the achievement of an objective, provide immediate feedback.

2) Help to individualize and differentiate instruction and remediate weaknesses of students.

3)Provide an "intellectual jungle gym" through problem solving experiences and instruction in programming.

4) Allow students to practice higher level thinking skills through open ended software.

5) "Put kids in charge of their own learning, give them unlimited access to feedback and information and let their curiosity do the rest."

Lowd identifies these goals to introduce a crucial question, "Should computers be used to do better that which we have always done or should computers be used to change our goals and methods?" She, as well as most members of the Logo community, advocates changing goals and methods through scenarios four and five. She argues for the importance of teaching skills such as organizing and analyzing data and the processes of decision making. While I do not agree with the urgency of one of her statements, ("...the survival...of the planet may depend on students mastering these new skills"), Lowd makes a strong case for curriculum reform.

But she also offers a warning:

" ...those of us who would like to see changes happen quickly (i.e. revolution) too often denigrate computer uses which are fairly traditional... We must stop confusing the 'right' goals for using computers with our unpopular idea that education must change. Computers ...can support more commonly accepted philosophies as well. "

Lowd suggests "evolution, rather than revolution," by creating scenarios four and five in our own classrooms, sharing methods with colleagues who show interest, and committee and council work. She believes that we can use the computer to work towards our goals while accepting its utility to others who may not share those same goals.

I wish I had said that.

>>>"A Nodal Land Investigation" by Tim Barclay, Kathleen Martin, and Tim Riordon, *The Computing Teacher,* March 1986.

I include this article because I think it illustrates a use of Logo which fits into scenarios four and five described above. The article outlines an activity using a variation of the INSPI procedure. The authors describe it as appropriate for middle school mathematics and science classes, but it could be used by any teacher who wants to involve students in skills such as testing hypotheses, detecting patterns, designing experiments, and controlling variables.

The article is based on a short procedure which students run many times with different inputs for a variety of purposes. Investigations are suggested that will interest most students and also give them a chance to practice important thinking skills.

The authors claim that the nodal land investigation is an example of what Papert talked about in his "Teaching Children to be Mathematicians Vs. Teaching Children About Mathematics" memo. I agree. Teachers who use the activity may not teach their students much Logo but there will likely be an awful lot of learning going on.

Bill Craig is the Computer Education Program Specialist for the Chesterfield County Schools, Chesterfield, VA.

LIFT's

Teacher to Teacher

by James Fry

Adventures with Logo and *More Adventures with Logo* [MIT (Terrapin/Krell) Versions] by Joyce Tobias and Carolyn Markuson, 1986, published by McGraw-Hill Book Company, 1221 Avenue of the Americas, New York, NY, 10020. [Teacher Guide also available.]

Target Audience: Logo students in grades four to six for *Adventures in Logo* and grades five to eight for *More Adventures in Logo.*

Content: *Adventures in Logo* and *More Adventures in Logo* are a two book set of Logo student textbooks designed to be used for a one or two semester Logo program. The authors recommend that *Adventures with Logo* (Book 1) be used with fourth through sixth graders as an introductory text for students who have had little or no Logo experience. Book 2, *More Adventures with Logo,* would be recommended for fifth through eight grades, for students with previous Logo experience. The authors suggest that if you are teaching a Logo course of four or five periods a week with 30 minutes computer time for the students each week, then each book should be able to be covered in a semester. If your students do not have that much computer time or instructional time with Logo then the books may take a year to cover.

The emphasis of *Adventures in Logo* is on turtle graphics. It covers the basic turtle commands for drawing pictures and designs. Halfway through the book, the students are introduced to procedures, how to write and edit them, and how to save them as a file on disk. The latter part of the book introduces some more advanced ideas, such as recursion and variables.

More Adventures with Logo starts with a review of turtle graphics and continues to expand upon what was introduced in Book 1. List processing is explored and Logo projects involving text is introduced. The idea of Inputs and output is covered so an interactive type of programming can be explored by the students. There are projects where the computer can be programmed to ask questions and respond on the screen with answers or messages. The later chapters cover more sophisticated Logo ideas such as global variables and conditional statements.

The accompanying teacher guides are an excellent resource of ideas and information to go along with the student books. They contain detailed instructions for each lesson, example solutions for the student projects, information on the language, and ideas on classroom management. There are also blackline masters for use with each of the lessons. Appendices are included on file management, common error messages, other useful commands, a glossary of terms, progress charts, and a list of primitives.

Strengths: Icons are used to help organize visually the information in the books. Books 1 and 2 use a series of three turtle icons to represent Remember Boxes, Projects to Try, and Ideas to Explore. This is an extremely helpful feature.

The chapters are set up with some instruction, some projects to try, then more instructions and suggested projects. The chapters follow the same type of pattern that teachers would use in teaching students at a computer, one of presenting some information, and then letting the students practice.

At the end of each chapter is a concise review page summarizing with "Now you can..." what the chapter presented, a list of the concepts or ideas learned, "Logo Words" (a list of commands that were introduced or practiced), and "New Vocabulary" (a list of new terms that were used.) Following is a page on Ideas to Explore, extensions to the ideas that were introduced.

The teacher guides are also well organized. Each lesson has the objectives stated, along with new vocabulary and materials needed for the lessons. Teaching strategies are presented along with questions to ask and expected answers. I feel this is important to have in any Logo teaching manual as a help to teachers who have not had a lot of Logo experience. Technical Tips are given to assist with some of the ideas or areas of Logo that inexperienced Logo teachers might not fully understand.

I feel the books are organized extremely well. Its easy to see the authors' experiences of being the Curriculum Coordinator for Computer Education and the Supervisor for Library and Media services has influenced the development and organization of the books.

James Fry uses Logo with his Chapter 1 remedial mathematics students at Novi Community School, Novi, MI, and is a co-founder of the LIFT group. His CompuServe number is 76317,565.

Logo LinX Postscript

For more information about AIRS Reading and Language Arts materials (see page 14), write to:

Theresa G. Murphy, Executive Director
AIRS
Andover Public Schools
Bartlett Street
Andover, MA 01810

For a printout of a Logo plural pattern microworld, check Data Library 13 on CompuServe's LOGOFORUM, or send a stamped, self-addressed long envelope to:

Judi Harris
Curry School of Education
Dept. of Ed. Studies
405 Emmet Street
Charlottesville, VA 22903

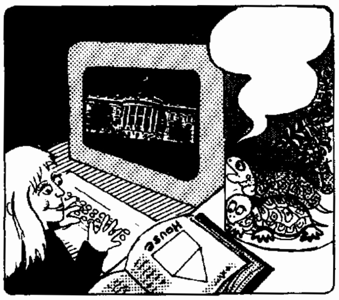
The Adventures of Jacques and Elsie

drawings by Linda Sherman

In our February cartoon, Jacques and Elsie were witnessing a somewhat unusual extension of the HOUSE procedure. Jacques was offering a comment on the situation.

Our favorite caption was submitted by Jennifer Hare, a 3rd grader at J. M. Moriarty School in Norwich, CT.

Mr. Reagan, just because I'm a turtle doesn't mean you have to hide in *your* house!



In the April cartoon, Jacques is definitely getting into the swing of things. His harmonizing (?) with the computer has prompted Elsie to offer a comment. What is she saying?

Please send your caption suggestions for the above cartoon no later than June 1, 1987, to: Jacques and Elsie, Logo Exchange, PO Box 5341, Charlottesville, VA 22905. Please include your name and address so that we can give proper credit should your caption be chosen.

All captions become the property of Meckler Publishing. None can be returned.

Linda Sherman is a freelance author and artist living in Shipman, VA, with her husband and two-year-old son.

Q and A

by Jim McCauley

Q. We are considering different versions of Logo for use as an introductory language for our computer science students at the college level. Are there any material advantages to the new dialect called Object Logo?

A. I hope that you will consider Object Logo (from Coral Software of Cambridge, MA) seriously and carefully. While its surface syntax is essentially identical to earlier versions of Logo, its underlying structure offers many advantages to computer science educators.

The object-oriented approach to computer programming is receiving a great deal of attention from computer scientists. Its most important advantage over traditional approaches is that it allows very clear definition of *modules*. A module is a program element that is intended to carry out a particular function or task without interfering with other functions of the same program. This is very important when designing large or complicated programs, especially if many programmers are involved in the same project. In conventional languages, if two programmers write different procedures but accidentally give them the same names, serious errors can result. The object-oriented approach offers methods of avoiding these kinds of errors.

In Object Logo, programmers can build environments in which program functions may be carried out with assurance that the effects of computation within a special environment will not "escape" to create trouble elsewhere in a large program.

Like the latest versions of Lisp, Object Logo uses *lexical scope* for its variables. In other Logo implementations, it's possible to to something like this:

```
TO BOX :LENGTH
REPEAT 4 [ SIDE ]
END

TO SIDE
FORWARD :LENGTH RIGHT 90
END
```

Notice that even though the value of :LENGTH is not *explicitly* passed to SIDE, the command FORWARD can still use the value bound to :LENGTH in SIDE's superprocedure, BOX. This ability to access the value of a variable bound in another procedure is called *dynamic scope*. It can create serious problems even in simple programs, since procedures like SIDE depend on special conditions (in this case, a binding of a value to :LENGTH) in order to run reliably. This violates the principle of modularity, and it is not allowed in lexically-scoped languages like Object Logo. Values must be explicitly passed, or special provisions must be made to allow a "family" of procedures to share values, thus reinforcing the concept of modularity.

Object Logo may offer some other advantages for university-level programming instruction. It compiles most procedures and command lines into machine code for fast execution, so it may be faster than some alternative languages and implementations. Also, through its "MacTypes," Object Logo offers direct and readily comprehensible access to the window, menu, and QuickDraw facilities of the Macintosh computer, so student programmers may experiment with the Macintosh user interface.

Finally, in addition to the "workspace" environment well-known to Logo programmers, Object Logo supports the sort of file-oriented programming that is more appropriate for use in the very large programming projects that are possible with this implementation.

One caveat: Object Logo differs from other object-oriented languages in that it does not distinguish sharply between classes and instances (as does Smalltalk, for example). Instead, it uses a more readily system of subclasses. For further information, contact Coral Software Corporation, PO Box 307, Cambridge MA 02142, or call (800) 521-1027.

Jim McCauley is a graduate student at the University of Oregon, studying with David Moursund, and has written Logo articles for many national publications. His CompuServe number is 70014,1136.

Improved TOWARDS Procedure Contributed

In the March issue of *Logo Exchange*, Sharon Burrowes published the third of her series of articles on LogoWriter. At the end of the column, she listed a set of procedures necessary for the TOWARDS and ARCTAN operations in LogoWriter.

She has sent an improved version of TOWARDS.1, one of the procedures in that set. Substitute this listing for the TOWARDS.1 procedure on page 26 of the March 1987 LX.

```
TO TOWARDS.1 :DX :DY
IF AND (:DY = 0) (:DX > 0) [OUTPUT 90]
IF AND (:DY = 0) (:DX < 0) [OUTPUT 270]
IF :DY = 0 [OUTPUT HEADING]
OUTPUT TOWARDS.2 :DY ARCTAN (:DX / :DY)
END
```

Ideas and suggestions about the use of TOWARDS, ARCTAN, and other LogoWriter tools may be sent to:

Sharon Burrowes
807 College Avenue
Wooster, OH 44691

CompuServe 73007,1645

IntLXual Challenges

by Robs Muir

GOTO: Curses and Recurses

As I stepped off the manky pin, I wondered about the 10a face ahead. My second was holding the cord, leaving me on the sharp end; I had little choice. The line of thin holds led to a thin-hand crack which started as a layback in a right-facing book and gradually became an offwidth. Fortunately, I had a rack of Friends. Finally at the stance, I clipped into the bolt, tied off and I yelled, "Off belay."

Clear as mud, right? You could probably make sense of that paragraph only if you were a technical rock climber. Some of us hear a lot of that talk here in California; Yosemite Valley, with its 3,000-feet granite cliffs, is crowded with such language.

All of which got me to thinking...

Language can help serve as a medium of expression between unlike groups, or it can help identify and insulate users from "outsiders." Many activities spawn such a specialized vocabulary; the sports pages are filled with similar slang. Novel activities encourage new language.

All of which brings me back to Logo. Are we engaged in a new activity that has built a new language expressly to help communicate our collective ideas? Or, is the novel language encouraging new activities? Well, I'm not sure either. But it certainly is fun!

One of best kept secrets of the Logo community has been Michael Friendly's long-overdue book, *Advanced Logo: A Language for Learning*. Some of us have had bits and pieces of Michael's manuscript since mid-1984. Unfortunately, when the bottom fell out of the Logo publishing market, so did Michael's original publisher. I'm pleased to announce that *Advanced Logo* has been picked up by L. Erlbaum Associates of Hillsdale, NJ. As I write this, the book is going to phototypsetting; its expected publication date is April or May. You will be amazed at its scope!

Last month, I received some electronic mail from Michael about an ongoing conversation he had been having with Brian Harvey. (Incidently, you know we've a long way to go when we get duplicate email referred to as cc. (carbon copy(??)) Michael suggested several possible challenges for this column.

Here's the gist of a good one for you language buffs (activity buffs?):

A BASIC Problem

One of the universal sentiments of modern programming languages is that everybody trashes the GOTO statement. Yet there are some cases where using the Logo equivalent of a 'computed GOTO' is quite elegant.

A while ago, I needed a general GLUE procedure, which took two inputs, either or both could be words or lists.

TO GLUE :PREFIX :SUFFIX works like this:

```
?SHOW GLUE "SOME "BODY
SOMEBODY

?SHOW GLUE "SOME [BODY TIMES]
[SOMEBODY SOMETIMES]

?SHOW GLUE "THANK [FUL YOU]
[THANKFUL THANKYOU]

?SHOW GLUE [AWKWARD STUPID] [LY NESS]
[AWKWARDLY AWKWARDNESS STUPIDLY STUPIDNESS]
```

My solution, which I was quite pleased with, handled the various combinations with the line

```
GO WORD (WORDP :PREFIX) (WORDP :SUFFIX)
```

So, I thought it might be fun to have a challenge to come up with the most compelling example(s) for retaining GO in Logo.

(Author's note: I have even seen a ruthless *Logo Exchange* editor reject Logo examples using GO on the pretext that it could corrupt our readership!)

GOTO It!

CHALLENGE #1: Can you recreate the GLUE procedure?

CHALLENGE #2: Can you come up with a compelling example or two? Here is a worthless example of how you can (usually) throw your Logo interpreter for a loop.

```
TO RECURSE :NUM
IF :NUM = 1000 [OP [ALL DONE!]]
OP RECURSE (:NUM + 1)
END
```

(Terrapin Logo: omit outside set of brackets in IF line.)

Observe the following error statements when PRINT RECURSE 1 is invoked. Just to make life more interesting, you might TRACE the process so you can watch the action.

On a Commodore 64:
NO STORAGE LEFT!, IN LINE
IF :NUM = 1000 OP [ALL DONE!]
AT LEVEL 66 OF RECURSE.

On a Macintosh 128 using LCSI's Logo:
Stack overflowStack overflow

Here is an equivalent RECURSE that *really* throws Logo for a LOOP:

```
TO GOTO :NUM
LABEL "TO!
IF :NUM = 1000 [OP [ALL DONE!]]
MAKE "NUM (:NUM + 1)
GO "TO!
END
```

(Terrapin Logo: omit outside set of brackets in IF line.)

Incidently, on the Macintosh even when I extend the workspace (the stack allocation) to a stack space of 20,000 and RECURSE works, the GOTO is still 12% faster!

It is of curious interest that Coral Software's Object Logo has completely eliminated GO from their primitive set. That's one way to squelch an argument!

Robs Muir is a physics and computer science teacher in Claremont, CA, and an instructor at the Claremont Graduate School. His CompuServe number is 70357,3403, and his Bitnet address is MUIR@CLARGRAD.

Conference Information

May 7, 8, and 9, 1987. *Great Lakes Logo Conference*, hosted by the Educational Computing Consortium of Ohio (ECCO), held in Cleveland, OH. The conference will feature preconference workshops on May 7, and a postconference LogoWriter workshop on May 9, with the conference program being presented on May 8. For more information, write to Alice Fredman, ECCO, 4777 Farnhurst Road, Cleveland, OH 44124, or call (216) 461-0800.

May 19 - 23, 1987. *Second International Conference and Exhibition: Children in an Information Age: Opportunities for creativity, INnovation, and New Activities*, sponsored by the Bulgarian Government and the Lyudmila Zhivkova International Foundation. For information, write to: Mr. Branimir Handjiev, State Committee for Research and Technology, 55 Chapaev Street, Sofia 1574, Bulgaria.

June 24 - 26, 1987. *National Educational Computing Conference*, Philadelphia, PA. For information, contact Laurie Shteir, Department of Computer and Information Science, Room 303, Computer Activity Building, Temple University 038-24, Philadelphia, PA 19122, or call (215) 787-1681.

July 5 - 9, 1987. *Second International Congress on Early Childhood Education: Childhood in the Technological Era*, sponsored by the Israel Ministry of Education and Culture and the Israel Association for Computers in Education. For information, write to: International Ltd., PO Box 29313, 61292 Tel-Aviv, Israel.

International Logo News

Dennis Harper
Logo Exchange International Editor

Global Comments

by Dennis O. Harper
Institute of Education
469 Bukit Timah Road
Singapore 1025, Republic of Singapore

This issue's international selections describe Logo efforts that are indicative of the sophistication some countries are experiencing. The birthplace of Logo can learn much from studying the techniques used in other cultures and witnessing the different ways Logo can progress in a country. The Latin American column describes a Logo group in Brazil that has been active since 1973. Their efforts have resulted in the infusion of Logo into the public schools of a large Brazilian state. The Asian article reports on a Chinese Logo publication that is initiating some very innovative Logo materials.

Throughout the past few months, the LX has seen many countries taking off with Logo. But a far greater number of countries have no Logo (or computers) in any of their classrooms. What is it that sparks schools to use Logo? In many cases it is the presence of a Logo enthusiast whose passionate fervor stirs others to use the language. Examples of this type of person are Reggini in Argentina, Leron in Israel, Sendov in Bulgaria, Kristiansdottir in Iceland, Ershov in the Soviet Union, Seye Sylla in Senegal, Goto and Weintraub in Japan, and Ling Qiyu in the People's Republic of China. It can be argued that these knowledgeable Logoites are necessary to spread Logo throughout a region of the world but their existence does not guarantee implementation on a wide scale. Although these people have met with varying degrees of success another important factor is necessary: high quality teacher training in Logo.

Logo and Teacher Education

Very little research exists in the area of Logo use in teacher education. This is unfortunate as teacher education is probably the biggest obstacle on the road to realizing the full potential of Logo as a learning tool throughout the world. Although some universities are adding a Logo component to their preservice programs and some literature is appearing on the subject, the proportion of such efforts compared to their importance is indeed low.

In many ways, preparing teachers to use Logo is not as easy as teaching it to children. For one, computers and Logo are unfamiliar to most teachers or student teachers (especially in developing nations); they have not grown up in a computer culture. Also, teachers already have a demanding schedule and the time to learn the language often is simply not available. A 10 to 20 hour rush course may leave the teachers overwhelmed and frustrated.

With these problems in mind, many teacher training centers feel that the inclusion of Logo is unwarranted. This doesn't imply, however, that there are no schools doing quite adequate jobs in teaching Logo. The fact of the matter remains that, regardless of grade level, most teachers using Logo have learned the language pretty much on their own and not in official classes. I sense that conferences, books, discussions with colleagues, and journals such as the *Logo Exchange* have had more influence than coursework and classes.

The coursework that is given is often a short 3-hour to two-day workshop for beginners. I used to be hesitant when asked to deliver a three-hour workshop on Logo or a one-hour workshop aimed at "enthusing teachers about Logo." Is this type of "scratching the surface of Logo" workshop doing more harm than good?

Although the answer to this question is debatable, one shouldn't turn away from the one-day or even half-day Logo workshop. Planned carefully and enthusiastically delivered, teachers can become excited and spurred on to investigate Logo further. Part of the workshop must be a discussion of where both human and written Logo resources can be found. Some present-day "power user" Logo teachers originally gained their spark from a 3-hour well-thought-out Logo workshop. Most teachers are bright enough to know that it takes more than the one-day workshop to develop a new approach to learning and to become a capable Logo teacher. They should treat the workshop for what it is -- an introduction.

Ideally, the teachers will have the chance to reflect on what they have learned and also be able to use the ideas and concepts they have gathered with children; availability of computers and Logo after the class is crucial to success. Around the world I find teachers who have access to computers but who don't know what to do with them. These are the type of teachers who can potentially benefit the most from the short workshop described above.

FD 100!

Dennis

Asia

by Hillel Weintraub
Doshisha International High School
Tatara, Tanabe-cho, Tsuzuki-gun
Kyoto-fu, Japan 610-03

Logo in China

We were fortunate to receive a letter from Ling Qiyu, the editor of *Children's Computerworld* telling us of some of the interesting computer-related events in China recently. This newspaper, published every two weeks, is the only one of its kind in China and is well received by children who are interested in computer activities. Apparently, the number of such children is growing rapidly these days.

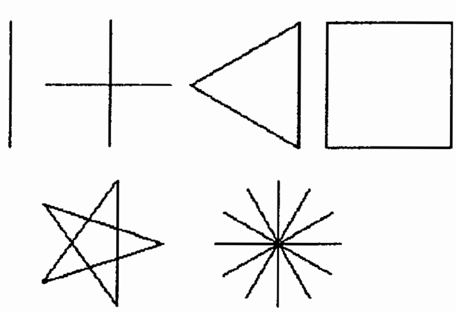
In October last year, at the ABC (American, Britain, China) Conference on Secondary School Computer Education, Ling Qiyu met Roy Pea (perhaps best known in the Logo community for an early rather negative evaluation of Logo's effects on children) and John Wood of the British Council and an editor of *MicroMath*, a new British publication with much Logo related material. Both Roy and John were so impressed with the Logo Challenge idea that it was decided to have an ABC version in 1987. And, as any user of the NLX alphabet procedures knows, there are more letters than ABC; why not add one more country each year, the three men wondered: Denmark, Egypt, France, Ghana ... So, perhaps your country can join this project in the near future, unless, you happen to live in Zambia, in which case it will take a while.

Now to give you a flavor of the Logo Challenges which were posed to the children of China in *Children's Computerworld*, the following list shows the tremendous range of challenges.

#1) Simple Graphics - Find as many ways as you can to draw a dotted line like this:

\- - - - -

#2) Complex Graphics - Write as *few* procedures as possible to create all these 6 shapes. Different variable values may be used as input.



#3) Words and Lists - Write a procedure which asks you to enter a list of three words, none of which have more than six letters. When you enter the words, the procedure displays them vertically in three columns. For example, if your list is [WE LIKE APPLES] the screen display will be:

```
W     L     A
E     I     P
      K     P
      E     L
            E
            S
```

#4) Using defined procedures - Some of the problems look simple but are beautifully constructed because they allow for total freedom in their response, and by the very nature of their instruction, encourage creativity and good programming habits. For example: Write three procedures which will draw the following figures.

Then write one or more main procedures which call them to create your own picture. Try to make your picture beautiful and meaningful. Give it a title. The fewer lines in your picture which are not drawn by these procedures, the better!

Ling Qiyu is the quintessential Logo teacher: not only does he ask questions which encourage thinking and allow for many levels of successful answers, but also he asks the learners to pose their own questions for others to think about. In other words, the learners become the teachers, a process which we know can, with some guidance, enable us to confront the strengths and weaknesses of a particular part of our knowledge system. Furthermore, in his *Children's Computerworld*, he even invites the children to redesign Logo. "Can you make any constructive suggestions about improving the Logo language?" he asks each person responding to the challenges.

These are revolutionary happenings in many educational systems in the world, the questions that will change the way children look at learning and teaching. I know that in *very* few classrooms in Japan are learners encouraged to make questions for each other, much less invited to challenge the very structure of the learning tools. It must be equally as rare in China and most other countries. Yet this is the very dream of Logo-using educators: to put the power of learning in the hands of the learner.

We are extremely grateful for Ling Qiyu for sharing his ideas with us and look forward to contact with him in the future.

Ling Qiyu, Executive Editor
Children's Computerworld
157 Changshu Road
Shanghai 200031 China

Other Asian News

In Japan, an arranged marriage is called an "Omiai." Who would have thought Logo might someday be involved in two people getting together? But it has! One of Japan's most eloquent Logo proponents, Takito Totsuka, has married a Logo-using kindergarten teacher, Kyoko Nishimiya. Takito, who spoke at Logo '85, is finishing his 1000 page (!) book on Logo's effect on children's behavior. Since Takito is one of the most sensitive, careful observers of children (and life) that I have ever met, I'm sure his book will be a valuable one. The book's dedication is from "Imagine," one of John Lennon's songs. "You may say I'm a dreamer, but I'm not the only one."

The International Council for Computers in Education (publishers of *The Computing Teacher*) have formed a new International Committee. Dennis Harper and I were invited to be members. Unfortunately, neither of us could attend the first meeting, held in Israel in February. At that meeting, the committee explored plans to apply new learning approaches to help individuals and societies solve the serious problems confronting the world.

I was the Japanese coordinator recently for an international project called "Kentucky to Japan: Bridge to Understanding." A high school here in Kyoto and one in Kentucky played Tom Snyder's "The Other Side" using telecommunication. The telegame was one of the keynote events at a large educational conference held in Lexington, KY. It was also an event highly publicized in Japan as an example of a new style of learning.

Latin America

by Eduardo Cavallo & Patricia Dowling
Instituto Bayard
Salguero 2969
1425 Buenos Aires, Argentina

The UNICAMP Logo Project - University of Campinas - Brazil

The Interdisciplinary Group of Informatics Applied to Education was created at the beginning of 1983 at the School of Education of the State University of Campinas (UNICAMP), with the Logo project as its central activity. This project was actually begun in 1973 after the visit of professor Afira Vianna Ripper to the Logo Laboratory at MIT, where she had had the opportunity of learning about the work of professor Seymour Papert and Marvin Minsky. They both visited the UNICAMP in 1975 and 1976.

As a result of these visits a first interdisciplinary group began to work. Its members were professors Fernando Curado from the Informatics Department, Maria Fausta Campos and Claudia Lemos from the Linguistics Department, Raymond Shepard and Marcia de Brito from the Department of Educational Psychology, and professor Afira Ripper, specially engaged to participate in this project. Later, several post graduate students joined the project, among them Jose A. Valente, who was sent to MIT for his post graduate study at the Logo Laboratory.

In 1978 work with children began, and as the activities increased it became necessary to train new instructors; this was done systematically from 1981.

The UNICAMP Logo project was the first of its kind to be undertaken in Brazil when very few people, even in other countries, were involved in the subject. It was initially based on a pilot study with a few children; the intention of the study was to verify the ways in which a Logo environment influences learning. A new philosophy for the use of computers in the areas of mathematics, physics, chemistry, biology and language was being searched for. The group's philosophy was that the computer should be basically used as a tool for learning, not a teaching machine.

The project is creating Logo learning environments with different socioeconomic characteristics in public schools. The aims of project are to:

1) Shape the Logo philosophy and language to the country's needs.

2) Develop teaching tools and promote the training of teachers who might introduce Logo principles in the classroom and within the curriculum.

3) Evaluate the teaching / learning process in a Logo environment within the context of regional public schools.

4) Study the learning process in children of different socioeconomic backgrounds but with similar stimulation.

During this period of nearly seven years, a "know how" has accumulated, and a tradition of investigation has developed which, with external support, will continue to grow.

Students for Students in Buenos Aires

Having verified that children's activities and needs never find a place in traditional congresses (conferences), we organized a Logo congress where adults were "forbidden" - only primary and secondary students would participate. The Bayard Institute was the host and its students worked actively in the preparation and development of this event.

More than 300 children from Buenos Aires, other regions of Argentina and the Crandon Institute of Montevideo, Uruguay, attended the two-day gathering which provided the opportunity to walk about the premises and exchange experiences. There was much to observe and participate in; one joyful presentation showed the activities which go on inside a computer from the point when it is first powered up through loading the language, on to the orders it receives, and ending with the storing of programs.

This was a good introduction to the film TRON, which describes a spectacular adventure inside a computer. Short documentaries from the United Nations along with cartoons produced by primary school children were also available for viewing. Logo games, workshops, demonstrations, debates, and running and jumping in the playground, eating hamburgers and hot dogs, were some of the other activities which were enjoyed at this very special event.

The teachers present also had a meeting place, and we didn't miss the opportunity of exchanging addresses and telephone numbers so replication of this event would be possible. Considering how the children really profitted from this idea, we hope it will be repeated; we all agreed on the necessity of providing children with this sort of opportunity.

LogoPals

by Barbara Randolph

LogoPals from coast to coast! I have had the delightful task of reading the letters of students from California to Wisconsin, from Nebraska to Ohio, Pennsylvania and New York. I call them "wishes-in-envelopes."

LogoPals are being requested from places both near and far:

Daniel Pasquarelli (McKees Rocks, Pennsylvania, USA): I would like a LogoPal from Italy, France, or Pakistan. I like to play sports, ride my bike and do math. I'm 10-1/2 years old.

Thomas Janci (Pittsburgh, Pennsylvania, USA): I am in the 4th grade and I am 9 years old. I love math, English, and Logo. Please send me a penpal from Russia. We can exchange Logo programs.

Neil Prakash (McKees Rocks, Pennsylvania, USA): Hi! My name is Neil. I'm in the 4th grade. I'm nine years old. I collect coins and I am a first degree black belt in Karate. Also I *love* computers. I would like a penpal from Greenland.

Alison Baehr (Wauwatosa, Wisconsin, USA): I love animals. I am planning to be a veterinarian someday. I am 10 years old and in the fifth grade. I would like a girl LogoPal from Spain, France, Romania, or Germany.

Jason Haas (Toledo, Ohio, USA): I like to play football and baseball. I like to swim, and I like to use Logo. My age is ten and I'm in fourth grade. I would like a penpal from the Congo.

Kori Kawczynski (Sylvania, Ohio, USA): I am a nine year old girl. My favorite hobby is ballet dancing. Logo is fun. I would like a girl penpal from Australia. This is going to be fun!

Jill Romanyshyn (Orchard Park, New York, USA): I am a computer student and I love Logo. I also like skiing. I am 8 years old. Please send me a LogoPal from California.

Elizabeth Grucella (Orchard Park, New York, USA): I am interested in science and drawing. I like skiing. I love computers and Logo. I'm in third grade. Send me a LogoPal from Australia or Florida.

Jill Lowrey (Lincoln, Nebraska, USA): I love to collect stickers and I like to ski. I love Logo and have fun with it in school. I'm 10 years old and in the 5th grade. Send me a LogoPal, please.

John Welsh (Coraopolis, Pennsylvania, USA): I am in the sixth grade and love Logo. Some of my favorite things to do are reading and singing in the choir. I would like a penpal from Ireland.

Jolynn Woehrer (Wauwatosa, Wisconsin, USA): I like to rollerskate. I also love to dance. I like working on the computer. I would like my LogoPal to be from New Jersey or Utah.

As you can see, our network of LogoPals is growing. These and many others I have heard from would *love* to write to Logo students from countries all over the world and I'd like to grant their wishes.

Teachers everywhere, share these ads with your students. They can request one of our featured LogoPals, if they like, or I will send them one of the many others available. Once a match is made, the letters and the sharing of Logo ideas begins.

Each new LogoPal needs to send a brief description about themselves along with a stamped, self-addressed envelope. Children writing from outside the USA need to enclose international postal coupons (purchased at the post office) for a 1-ounce or 28-gram reply. Write to: Barbara Randolph, LogoPals, 1455 East 56th Street, Chicago, IL 60637 USA.

Classified

** Teacher's Pet **

"...the best robot I've ever seen..."

–Seymour Papert, M.I.T. LOGO '85 Conference, Cambridge, Massachusetts.

"We have found the Valiant Turtle's presence has increased the interest level of students and teachers in Logo, facilitated the understanding of Logo commands, and changed personal interaction during Logo work, providing students more time to share and discuss their observations."

–June 1986 review in *The Computing Teacher* by Jim Ellickson-Brown, Mountain View Elementary School and Doris Mimnaugh, 4th grade teacher, Wilson Elementary School, Corvallis, Oregon.

Meet Valiant Turtle, state of the art in Logo programming! Controlled from your Apple II+, IIe , IIc, IBM PC, PCjr or Commodore 64 computer by an infra-red beam, the Valiant Turtle requires no cords or wires to move. Create a design on the screen and watch Valiant draw it on paper! Watch as Valiant executes your commands in three dimensions. Plan some fancy footwork and choreograph a dance in Logo!

Valiant Turtle is the ideal learning tool for students learning Logo as well as a great introduction to the world of robotics. Valiant is simple to use and easy for even the very youngest Logo learner to understand. Complicated and sophisticated ideas are presented in simple, graphic form. Students build artificial intelligence concepts and learn to think about space and spatial relationships.

Valiant Turtle operates with most popular versions of the Logo language. Watch through Valiant's transparent plexiglass dome as the Valiant Turtle draws with extreme accuracy with its built-in pen. Valiant comes completely assembled with easy-to-read instructions and control software. Because its so easy to set up and use, the Valiant generates much enthusiasm in the classroom! Young imaginations fly whenever Valiant Turtle enters the room!

The Valiant Turtle is available from Harvard Associates, Inc., 260 Beacon Street, Somerville, Massachusetts 02143. Harvard Associates provides full technical support for the Valiant. For more information, or to order your Valiant Turtle, please call (617) 492-0660.