

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 4

FORWARD 100!

December 1984

Functions of OUTPUT

by
Nancy Kovatch

"How can I use Logo to make algebra more meaningful to my students?" This question was my motivation for developing the Logo ideas which follow.

Although my algebra students were able to deal with linear functions in the x- and y-notation, invariably they had difficulty dealing with the function notation. The expression $f(x)=2x+10$ didn't come as easily as $y=2x+10$. When chalkboard diagrams of function machines with input and output failed to help, I realized that the Logo OUTPUT primitive might do the trick.

Here is a set of typical procedures which I typed into the computer before class.

```
TO F :X
OUTPUT 2 * :X + 10
END
```

```
TO G :X
OUTPUT :X + 2
END
```

During class, my students evaluated the functions at several points, and experimented with function operations.

```
PRINT F 3
16
PRINT (F 3) + (G 5)
23
PRINT F (G -2)
10
```

Although these functions are linear, you could use the same technique to explore quadratic, cubic, and trigonometric relationships as well.

HOUSEKEEPING

Here are some additional procedures I developed for housekeeping and to facilitate exploration.

```
TO AXES :YSIZE :XSIZE
PU HOME CS PD
Y.AXIS :YSIZE
RT 90
X.AXIS :XSIZE
END
```

```
TO Y.AXIS :Y
FD :Y BK 2*:Y FD :Y
END
```

```
TO X.AXIS :X
FD : BK 2*:X FD :X
END
```

```
TO SET.TURTLE :X :Y (LCSI versions)
PU SETPOS LIST :X :Y PD
END
```

```
TO SET.TURTLE :X :Y (MIT versions)
PU SETXY :X :Y PD
END
```

After drawing the AXES, my students used SET.TURTLE and the predefined functions for their explorations.

```
SET.TURTLE -50 (F -50)
SET.TURTLE 10 (F 10)
SET.TURTLE 50 (F 50)
```

Sometimes, they used this sequence.

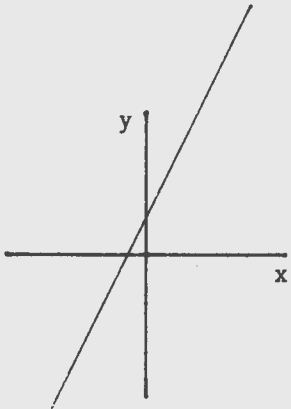
```
PRINT F 40
90
SETPOS [40 90] or SETXY 40 90
```

Once they had a good understanding of the behavior of the function, they had the turtle draw the line.

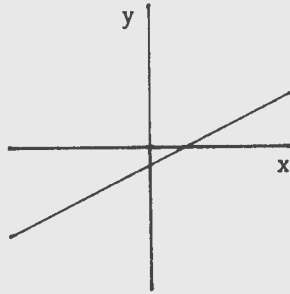
Functions continued

Another project they enjoyed was the investigation of the behavior of inverse functions. That is, if $f(x)=2x+10$, then the inverse function is $x=(f(x)-10)/2$. The inverse of the procedure F could be written

```
TO F.INV :X
OUTPUT (:X-10)/2
END
```



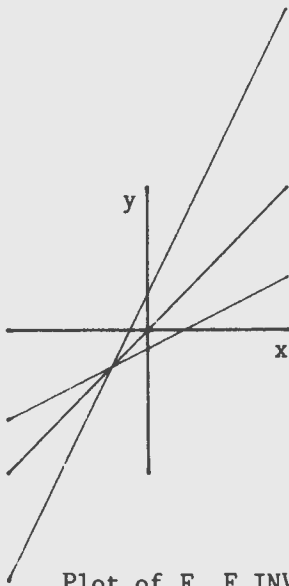
Plot of F



Plot of F.INV

When working with inverse functions, the identity line is useful. This is a line from the lower left to the upper right so that $x=y$ at all points.

Here is what F, F.INV, and the identity line look like all together. The students fold the graphics printout along the identity line to verify the reflexive property of inverse functions.



Plot of F, F.INV, and the identity line

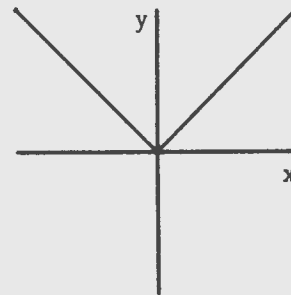
Several students experimented with ways to estimate the y-intercept. They moved the turtle along either the function line or the y-axis until the turtle was near the intercept. Then, with PRINT POS (or PRINT XCOR PRINT YCOR) and instructions to print various values of F or F.INV, they zeroed in on the intercept point.

The slope of a linear function could be investigated in a similar manner, using SETHEADING and HEADING.

WEIRDO FUNCTIONS

Many students became fascinated with "strange" linear functions. For example, the absolute value function is easy to program in Logo, but difficult for some students to understand. Exploration with the SET.TURTLE procedure gives them a personal encounter with the "beast."

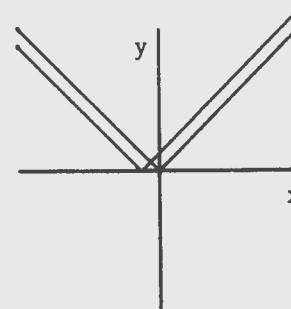
```
TO ABS :X
IF :X < 0 [OUTPUT -:X]
OUTPUT :X
END
(MIT Logo: omit the brackets above.)
```



Plot of ABS

Later, just for fun, I gave them related functions to compare, such as

```
TO B :X
OUTPUT ABS :X + 5
END
```



Plot of ABS and B

Functions continued

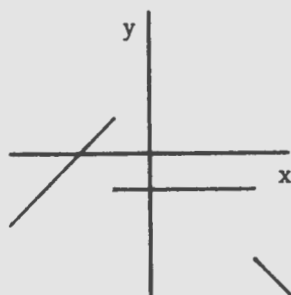
MORE WEIRDO FUNCTIONS

Here is an "exotic" linear function which gave my students quite a challenge.

For $x \leq -10$ $f(x)=x+20$
 For $-10 < x < 30$ $f(x)=-10$
 For $x \geq 30$ $f(x)=-x$

This is the Logo procedure.

```
TO H :X
IF (OR :X<-10 :X=-10 ) [OUTPUT :X+20]
IF (AND :X>-10 :X<30 ) [OUTPUT -10]
OUTPUT -:X
END
(MIT Logo: omit the brackets above.)
```



Plot of H

By this time, the students were enthusiastically making up their own functions and challenging each other to sketch them. Logo proved to be an able arbiter of many heated discussions!

With these few beginning ideas and procedures, my students explored many second year algebra concepts in a way never before possible. Further, they were able to create, test, and modify their own predictions in an encouraging atmosphere.

The OUTPUT operation produced an output for my class: that of experienced mathematical thinkers and enthusiastic problem solvers. It can do the same for you!

Nancy Kovatch is a teacher and computer coordinator at St. Anne's Belfield School and an adjunct faculty member at Piedmont Virginia Community College in Charlottesville, VA. She is currently studying applications of Logo in second year algebra classes.

From the Editor

Looking and Seeing

"Look, Jane, look and see." What memories that phrase brought back to me! Dim visions of my wonderful first grade teacher, Miss Eloise Miller, flooded my mind as I thumbed through the worn pages of the classic primary book.

I began thinking about "look and see." I wondered why the author chose that phrase. Look and see.

My memory recreated the first class of a photography course taken years ago. The instructor was speaking.

"...and your first homework assignment is to learn how to see. You each are to select a vantage point and remain one full hour, doing nothing but looking quietly at what is there. Record your observations. After a while, start thinking about what you are looking at. Try to imagine things you could photograph. Next, imagine what small changes you could make to improve your imaginary photographs. Record these observations. At first, you will be looking. Later, you will be seeing."

Look and see. What's the difference? Thinking, imagining, ... making something of what you are looking at?

This simple but important concept has great potential for Logo. There are things happening all around us which make wonderful Logo explorations. But we have to think while we are looking at them in order to see them.

In the review section of this issue, David Thornburg's magnificent books feature explorations of patterns which can be seen in nature. The NLXual Challenge is based on an observation of Kyser as he toddled toward his shadow. Many more suggestions come if we look and see more often.

Why not encourage students to try this idea? Make Logo part of a homework assignment. What things can students find in their daily lives which could be explored with Logo?

But, be prepared. The results may astonish you!

May you see a special joy during the holiday season!

FD 100!

Tipps for Teachers

by Steve Tipps

Numbers in Between

Last month, the topic was development of a number guessing game. TEST was used to determine whether the guess was right or not. The game responded whether the guess was higher or lower than the target. However, the game did not tell the players if they were getting close to the right number.

The idea of getting close to a number may lead to another game. An improvement of the guessing game would tell when the guess is within five of the right answer. One way of expressing this would be: "Subtract the guess from the number and TEST the difference."

This simple idea can be a real mind bender because the difference can be 10, 5, 3, 1, -5, -20, and the like. You really want to tell if the guess is five too high or five too low.

How do you TEST for a difference less than 5 but not less than -5? One way is to use ALLOF in the TEST.

```
TEST ALLOF (:NUMBER-77<5)(:NUMBER-77>-5)
IFTRUE PRINT [PRETTY CLOSE.]
```

This is read: "Find out if the difference is less than 5 and more than -5. If that's true, you are getting close."

BETWIXT AND BETWEEN

The same high/low idea could be used in a starter game of BETWEEN?. The guesser would give a low number and a high number. The game would tell whether the target number was between them.

```
TO BETWEEN? :HIGH :LOW
TEST ALLOF (:LOW < 54 ) ( 54 < :HIGH )
IFTRUE PRINT [ IT IS BETWEEN ]
IFFALSE PRINT [ IT IS NOT BETWEEN ]
PRINT1 :LOW PRINT1 [ ' AND ' ] PRINT :HIGH
END
```

The guessing game could begin by finding the range of high and low numbers. Then use GUESS.IT to get it exactly right.

```
BETWEEN? 10 40
IT IS NOT BETWEEN
10 AND 40
BETWEEN? 40 70
IT IS BETWEEN
40 AND 70
BETWEEN? 50 60
IT IS BETWEEN
50 AND 60
BETWEEN? 53 55
IT IS BETWEEN
53 AND 55
```

After narrowing the range, GUESS.IT could be used to pinpoint the number. Of course, the value of "IT needs to be the same for both the BETWEEN? and the GUESS.IT procedures. The target in BETWEEN? must be changed from a fixed number to a variable IT. The value of IT is used instead of the fixed numbers in the TEST.

LINE OF NUMBERS

Showing the range of numbers on the screen might be a way to help students picture which numbers are too high and which are too low. The turtle could mark the guesses or show the high and low numbers.

In the SPLITSCREEN mode, draw a number line and label it. The number line is drawn at the bottom of the graphics screen. The number labels appear on the TEXTSCREEN just under the line. Pseudocode for the number line might be:

1. Draw a number line using graphics and text.
2. Move the turtle to the bottom of the split graphics screen.
3. The screen is about 280 wide. I want 100 numbers on it. I can use $280/10 = 28$ turtle steps for each 10 numbers.
4. Draw a line with marks every 28 turtle steps.
5. Put 10 20 30 ... on the text line.
6. Move the cursor to line 22 to work out problems.

Three procedures are needed to make and label the number line.

```
TO NUMBER.LINE
DRAW HIDETURTLE
PU BK 75 RT 90 BK 140 PD
REPEAT 10 [MARK FD 28]
BK 1 MARK
PRINT.LABEL
END
```

Tipps continued

```
TO MARK
LT 90
FD 2 BK 2
RT 90
END
```

```
TO PRINT.LABEL
CURSOR 0 20 PRINT [0]
CURSOR 2 20 PRINT [10]
CURSOR 6 20 PRINT [20]
CURSOR 10 20 PRINT [30]
CURSOR 14 20 PRINT [40]
CURSOR 18 20 PRINT [50]
CURSOR 22 20 PRINT [60]
CURSOR 26 20 PRINT [70]
CURSOR 30 20 PRINT [80]
CURSOR 34 20 PRINT [90]
CURSOR 37 20 PRINT [100]
CURSOR 0 22 PRINT [ ] CURSOR 0 22
END
```

The number line can be used in many ways. Guesses can be put on it. For example, each time a number is guessed, a STAR is drawn at that number on the line.

HOME ON THE RANGE

Small stars or marks may be hard to read on the screen. A better way would be to have the range marked on the number line. A RANGE procedure draws a red line from 0 to the :LOW number, then a green line from the :LOW to the :HIGH, and finally a red line from :HIGH to 100 on the number line.

```
TO RANGE :LOW :HIGH
HT PU HOME
BK 40 RT 90 BK 140 PD
PC 4 FD :LOW / 10 * 28
PC 2 FD (:HIGH - :LOW) / 10 * 28
PC 4 FD (100 - :HIGH) / 10 * 28
END
```

Try out the number line and RANGE several times to see how they work. Get a clear picture of how many numbers are between the low and high numbers.

RETURN OF BETWEEN

The old BETWEEN procedure can be elaborated upon with the visual display in RANGE. The IS.IT.BETWEEN? procedure has more information and more feedback about the range of numbers. One refinement is a filter at the very beginning to check for :LOW being lower than :HIGH.

```
TO IS.IT.BETWEEN? :LOW :HIGH
FILTER :LOW :HIGH
TEST ALLOF (:LOW < :IT)(:IT < :HIGH)
IFTRUE PRINT [ YES, IT IS.]
IFFALSE PRINT [ NO, IT IS NOT.]
IFTRUE RANGE :LOW :HIGH
DELAY 3000
PRINT.LABEL
END
```

```
TO FILTER :LOW :HIGH
TEST :LOW < :HIGH
IFFALSE PRINT [ YOU ARE TRYING TO
TRICK ME.]
IFFALSE PRINT [TRY AGAIN.]
IFFALSE TOPLEVEL
END
```

```
TO DELAY :NUMBER
REPEAT :NUMBER [ ]
END
```

A START procedure calls all the other procedures necessary to begin the game of IS.IT.BETWEEN?. The RANDOM.SET.IT procedure could assign a random number as the target. For ideas on how to do that, see Patrick Salsich's article in this issue. A complete set of printed instructions would be a good addition.

```
TO START
RANDOM.SET.IT
INSTRUCTIONS
NUMBER.LINE
END
```

By asking IS.IT.BETWEEN? several times, you get a smaller and smaller range. If the random number was 72, the game would look like this.

```
IS.IT.BETWEEN? 0 50
NO, IT IS NOT.
IS.IT.BETWEEN? 50 100
YES, IT IS.
IS.IT.BETWEEN? 50 75
YES, IT IS.
IS.IT.BETWEEN? 65 75
YES, IT IS.
IS.IT.BETWEEN? 70 75
YES, IT IS.
```

Each time a number line is redrawn to show the green range. An exploration focuses on the range of right numbers regardless of whether the exploration is

Tipps continued

systematic or haphazard. At the end of the exploration, a direct guess may be wanted. GUESS is changed to an IS.IT? procedure which also puts a star at the right place on the number line.

```
TO IS.IT? :NUMBER
TEST :IT = :NUMBER
IFTRUE PUT.STAR :NUMBER
PRINT.LABEL
IFTRUE PRINT1 :NUMBER PRINT1 [' IS THE
RIGHT NUMBER.']
IFFALSE PRINT1 :NUMBER PRINT1 [' IS
NOT THE NUMBER.']
END
```

```
TO PUT.STAR :NUMBER
HT PU HOME
BK 40 RT 90 BK 140 PD
PC 4 FD :NUMBER/ 10 * 28
PC 3 LT 90 FD 35
PC 2 LT 15 STAR RT 15
PC 3 BK 50 FD 15 RT 90
PC 4 FD (100 - :NUMBER )/10 * 28
END
```

```
TO STAR
REPEAT 5 [ FD 30 RT 144 ]
END
```

Only three or four procedures are needed to create the range game. But each one of the procedures should challenge you and the students to think about the position of numbers and how to show that position with Logo. The modular structure of Logo is a great aid in solving this problem. Each part can be worked out, defined, debugged, and combined.

The procedures for guessing and estimating and playing with numbers should look very different by the time students have worked on them. GUESS and BETWEEN? are starters because they open up many other possibilities.

Working out how things look on the screen may be as challenging a problem as the TEST and consequences. But whether the number is too high, too low, in between, or just right, Logo is a marvelous tool for logical thinking and problem solving.

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

NLXionary

A Lectionary of Selected Logo Readings

by Griff Wigley

The following selection is from a paper written for Logo 84, the National Logo Conference held at MIT this past June, and is included in the preproceedings.

(Ed. note: To obtain a copy of the Logo 84 preproceedings, send \$15 US to MIT Logo Group / Publications, NE 43-438, 545 Technology Square, Cambridge, MA 02139.)

"Logo as an Empirical Window," by Sylvia Weir (an excerpt from her forthcoming book, "Untrapping Intelligence").

>Although I don't know her personally, Sylvia Weir seems to be the Salvador Minuchin of Logo teachers and researchers. Family therapists amongst you, listen up. Therapist A asks the family what the problem is, and then suggests a solution based on what they say. When this doesn't work, Therapist A typically ponders, "How can I make my suggestion clearer?"

>Therapist B asks the family to demonstrate the problem and then prescribes a task that seeks both to change and further diagnose the structure (pattern of interaction) of the family. When this doesn't work, Therapist B typically ponders, "What else could I try to change the structure so that they can solve this and future problems themselves?"

>Yes, therapists as well as teachers frequently have their own agenda to impart to their clients, and tend to strive for greater clarity when hitting a brick wall. (Need I mention blaming as an alternate response?)

>Sylvia Weir and Salvador Minuchin go deeper. In this excerpt, Weir challenges teachers to ask, "How can I set up situations so as to enable my students to find their way to an understanding?" Likewise, she challenges researchers to ask, "What must I provide so that what my subjects know already, as well as what they can come to know, will surface in a clear way for me to observe and probe?"

>What could possibly be beneficial to observing and analyzing something so insignificant as family members seating them-

NLXionary continued

selves for a session, or a student's first attempt at choosing a series of numbers to go with an initial exploration of FD and BK? The true professionals among you will want to know.

>"Logo as a window" is a phrase that evokes memories of the one-way glass and videotape of my family therapy training. Few therapists really use them as diagnostic tools. They take too much time.

>"Got to get on with solving the problem." "These students should be learning REPEAT by now." Curriculum, curriculum, curriculum.

>Like most of the really good stuff about Logo, its potential as a window is available only through the skill of a teacher. I leave it to you to decide whether that is good news or bad news. In either case, watch for her book!

"SmartLogo," a review by David Wilson, InfoWorld, November 12, 1984.

>A glowing 4-disk rating review of LCSi's Logo for the Coleco Adam home computer. Among the positives: a speedy, turtle-like turtle; 16 colors for ink or screen; 30 turtles with a shape editor; commands for fill, speed, sound, and music; great documentation; and "more list processing capabilities than we have ever seen in any prior version of Logo for microcomputers." The only fault the reviewer finds is with the computer - no disk software (the Adam uses a tape drive), and no graphics printer.

>The reviewer writes as if he has never seen any of these capabilities in Logo versions before, and mentions other versions only to compare turtle drawing speed. He says SETSPEED is a totally new primitive when, in fact, it has been around since 1981 when TI Logo was released.

>Actually, none of the features he mentions are brand spanking new. (However, SmartLogo does sound like a top contender with LCSi's Sprite Logo for the most complete Logo version to date.) Clearly, a more thorough review of SmartLogo is needed quickly, or the Adam may never make it out of the Cabbage Patch. (And send the man some NLX back issues!)

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

Turtle Tips

by
Donna Lanyi
and Jane Toth

So how do we get to Michael Jackson's house? Students may need additional practice in guiding the turtle. An excellent gamelike activity to practice this concept makes use of mazes.

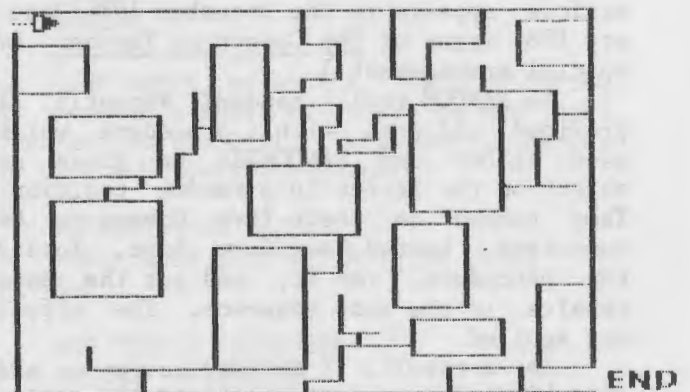
TEACHER HINTS

We trace commercially prepared mazes onto transparency material. There are some excellent mazes in the MECC Logo publications. (For a current catalog, write MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.) The students tape the mazes to their monitors and practice guiding the turtle through.

If the transparency material is light enough, static electricity will hold it to the screen. Turning the monitor off and on sometimes seems to make more static "cling."

Encourage your students to create their own mazes, such as how to get to MJ's house, to Santa's Workshop, and the like. They really enjoy practicing with mazes. Our students think of the activity more as a video game than turtle practice!

Thanks to Elizabeth Moody, former Computer Coordinator at Trinity School in New York City, who sent us a procedure to draw the following maze her students used for practice.



TurtleTips continued

PROCEDURE OF THE MONTH

Happy Holidays!! Encourage your students to create graphics for the holiday season. How about a picture of one of Santa's reindeer? Or all eight?



If you are interested in a free listing of this procedure or Elizabeth's maze procedure, please send us a self addressed stamped envelope.

We invite elementary teachers and students to send their procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Really Random

by
Patrick Salsich

(Ed. note: A similar version of this article appears in the December 1984/January 1985 issue of The Computing Teacher by special arrangement.)

Is RANDOM really random? Recently, I provided children with a procedure which used RANDOM and RANDOMIZE to place an object on the screen in a random position. They turned on their five Commodore 64 computers, booted Commodore Logo, loaded the procedure, ran it, and got the same results in the same sequence. The effect was spoiled.

As a result, I decided to try to add some kind of unpredictability to the procedure. Here is what I learned.

A LOOK AT RANDOM

Have you ever tried to generate a random number with Commodore Logo? It is not as easy as it seems. RANDOM is supposed to output a random number within certain limits, but, in fact, it outputs the first number in a predetermined sequence designed to simulate randomness. This approach is standard with many computers.

The RANDOMIZE command is used to select the starting point in the sequence RANDOM uses. Unfortunately, if RANDOMIZE is used with no input, it starts at the same number in the sequence each time the computer is turned on. Try the following to see what I mean.

Turn on your Commodore 64 and boot Commodore Logo. Then define and run the following procedure.

```
TO TRY.RANDOM
RANDOMIZE
REPEAT 10 [PRINT RANDOM 5]
END
```

Write down the sequence of numbers generated. Now turn the computer off, then on, and reboot Logo. Type and run the same procedure. The same sequence of numbers is generated.

We can change the sequence. RANDOMIZE can accept an input which determines the point (called the "seed") in the random number sequence at which RANDOM begins. For example, if you replace RANDOMIZE with (RANDOMIZE 56) in the TRY.RANDOM procedure (the parentheses are required), a different sequence is produced, but the sequence of numbers does not change each time the procedure is run.

If we could arbitrarily choose the input for RANDOMIZE, we could produce an unpredictable result with TRY.RANDOM each time the computer is turned on. Here is one approach to the task.

HOW TO RANDOMIZE

GROW.NUMBER is a procedure which counts up from an initial number until it is interrupted with a key press. Run it a couple of times by typing GROW.NUMBER 0 to verify that it is difficult to produce the same number more than once.

```
TO GROW.NUMBER :COUNTER
IF RC? PRINT :COUNTER STOP
GROW.NUMBER :COUNTER + 1
END
```


Random continued

Now change the procedure so that it uses the stopping value of COUNTER as the input to RANDOMIZE.

```
TO GROW.NUMBER :COUNTER
IF RC? (RANDOMIZE :COUNTER) STOP
GROW.NUMBER :COUNTER + 1
END
```

With the addition of a prompting procedure and an appropriately changed TRY.RANDOM procedure, we're in business.

```
TO GET.RANDOM.SEED
PRINT [PRESS ANY KEY TO BEGIN]
GROW.NUMBER 0
TRY.RANDOM
END
```

```
TO TRY.RANDOM
REPEAT 10 [PRINT RANDOM 5]
END
```

When I used a similar set of procedures with the children later, no one got the same results, even when each began with booting up. Thus, this is one way of generating more unpredictable sequences of numbers with Commodore Logo.

(Ed note: To eliminate the spurious character on the screen after GROW.NUMBER stops, change the appropriate line to IF RC? MAKE "KEY RC (RANDOMIZE :COUNTER) STOP. In case you are interested, RANDOMIZE accepts as input a number less than 2.147485E9.)

Patrick Salsich is an elementary teacher on special assignment in the Portland (OR) Public Schools to develop a program of Logo instruction for grades 1-5. He is pursuing a Masters Degree in Computer Science Education at the University of Oregon, and is the founder of the Portland Area Logo Users' Group.

Logo Templates for Apple

Keyboard templates which display Logo commands are available for the Apple II+ and IIe computers. A student model includes an attached booklet with pages which allow the insertion of lessons or assignments. An advanced model displays a wide variety of Logo commands.

For more information, write Integrated Computer Marketplace, Box 3610-I, SR 2, Tehachapi, CA 93561, or call (805) 822-7575.

Listful Thinking

by
Glen Bull and Paula Cochran

Making a List and Checking It Twice

It's that time again! Perhaps you and your students already have some holiday lists in mind -- things to do, things to buy, friends to remember. This month, we'd like to show you how to check your lists. We also have two holiday games for you.

MAKING A LIST

Next to that of the seven dwarfs, the list of the names of Santa's eight reindeer is among the most frequently forgotten lists of all time. How many can you remember? We'll start you off with a list of two reindeer.

```
MAKE "REINDEER [COMET DANCER]
```

As you recall from last month, INSPECT is a good way to look at a list. It prints one reindeer on each line. In Terrapin Logo, omit the brackets around [STOP].

```
TO INSPECT :LIST
IF :LIST = [] [STOP]
PRINT FIRST :LIST
INSPECT BUTFIRST :LIST
END
```

Here is your team so far, Santa.

```
INSPECT :REINDEER
COMET
DANCER
```

Last month, we made a list of pilgrims and created a short procedure to ADD more pilgrims to the list. This month, we can use this tool to ADD reindeer.

```
TO ADD :ITEM :LIST
MAKE :LIST LPUT :ITEM THING :LIST
END
```

```
ADD "CUPID "REINDEER
INSPECT :REINDEER
COMET
DANCER
CUPID
```

Listful Thinking continued

GUESS THE REINDEER

The tools INSPECT and ADD make a reindeer guessing game. If your Logo version does not have a READWORD primitive, a READWORD procedure is furnished at the end of the column.

```
TO REINDEER.CHALLENGE
INSPECT :REINDEER
PRINT [NAME A NEW REINDEER.]
MAKE "NAME READWORD
ADD :NAME "REINDEER
END
```

Oops! On inspecting the reindeer, it appears that someone added DANCER twice.

```
INSPECT :REINDEER
COMET
DANCER
BLITZEN
DANCER
```

No problem! You can use the EDNS (EDIT NAMES in Terrapin Logo) command to edit the list. This method for editing lists was discussed in last month's column.

CHECKING THE LIST

But, why not let Logo check a reindeer before you add it to the list? Here is a way to find out whether a reindeer is already a member of Santa's team.

```
PRINT MEMBERP "DANCER :REINDEER
TRUE
PRINT MEMBERP "RUDOLPH :REINDEER
FALSE
```

***In Version 1 of Terrapin Logo, you need to use the MEMBER? procedure found at the end of the column. In Version 2 of Terrapin Logo, the MEMBER? command is built in.

The MEMBERP command is used to check the list before adding a reindeer:

```
TEST MEMBERP :NAME :REINDEER.
```

This line doesn't do anything by itself, but we can use it to improve the REINDEER.CHALLENGE.

```
TO REINDEER.CHALLENGE
PRINT [NAME A MISSING REINDEER.]
MAKE "NAME READWORD
TEST MEMBERP :NAME :REINDEER
```

```
IFTRUE [PRINT SE :NAME [IS ALREADY
HARNESSED.]
IFFALSE [ADD :NAME "REINDEER]
END
```

Here's what happens if you try to add a reindeer that's already on the list.

```
REINDEER.CHALLENGE
NAME A MISSING REINDEER
DANCER
DANCER IS ALREADY HARNESSED.
```

THE ANAGRAM GAME

An anagram is a word made from the letters of another word. We have used Logo to make a holiday anagram game. There are two skills involved in playing this game: combining letters to make new words, and remembering what words have already been used. Here is what it's like to play:

```
ANAGRAM
THINK OF A WORD THAT ONLY USES
LETTERS FOUND IN "GREETINGS"
```

```
WHAT IS YOUR WORD?
SING (typed by player)
THANKS! THAT IS A NEW ONE.
```

```
NOW THE WORDS ARE:
TIN
GET
TRITE
SING
```

You don't have to use all of the letters in the key word, and you may use some letters more than once. But you can't use a word that is already on the list! You must think of a new one.

```
WHAT IS YOUR WORD?
GET (typed by player)
I ALREADY KNOW THAT ONE.
```

The ANAGRAM game starts off with instructions for the players.

```
TO ANAGRAM
MAKE "KEY.WORD "GREETINGS
MAKE "WORD.LIST []
PRINT [THINK OF A WORD THAT ONLY]
PRINT SE [USES LETTERS IN] :KEY.WORD
PRINT []
GET.GUESS
END
```

Listful Thinking continued

Now the player guesses a word.

```
TO GET.GUESS
PRINT [WHAT IS YOUR WORD?]
MAKE "GUESS READWORD
TEST MEMBERP :GUESS :WORD.LIST
IFTRUE [PRINT [I ALREADY KNOW THAT
ONE.]]
IFFALSE [ADD.TO.LIST]
PRINT []
GET.GUESS
END
```

***In Terrapin Logo, omit brackets around consequences. For example, type

```
IFFALSE ADD.TO.LIST instead of
IFFALSE [ADD.TO.LIST]
```

If the player guesses a word that is already a member of the WORD.LIST, the program says, "I already know that one." If the guess is not a member of the list, the program adds it to the list.

```
TO ADD.TO.LIST
ADD :GUESS "WORD.LIST
PRINT [THANKS! THAT IS A NEW ONE.]
PRINT []
PRINT [NOW THE WORDS ARE:]
INSPECT :WORD.LIST
END
```

Use CONTROL-G to stop the ANAGRAM program.

Challenge: Can you improve ANAGRAM so the program stops when a player types QUIT?

CHECKING IT TWICE

The ANAGRAM program can remember the words you typed in already, but it has a flaw.

```
WHAT IS YOUR WORD?
CELEBRATION
THANKS! THAT IS A NEW ONE.

NOW THE WORDS ARE:
TIN
GET
SING
CELEBRATION
```

The program accepted CELEBRATION even though many of the letters in "Celebration" are not in "Greetings". In some versions of Logo, MEMBERP or MEMBER? check to see if a letter is a member of a word. Here is a

table we compiled for several versions of Logo.

| <u>Name</u> | <u>Version</u> | <u>Check Letters?</u> |
|-------------|-----------------|-----------------------|
| MEMBERP | IBM Logo | Yes |
| MEMBERP | Apple Logo | No |
| MEMBERP | Apple Logo II | Yes |
| MEMBER? | Terrapin 1 Logo | Yes (Procedure) |
| MEMBER? | Terrapin 2 Logo | Yes |
| MEMBER? | Commodore Logo | Yes |

Even though the MEMBERP command in Apple Logo will not check to see if a letter is a member of a word, we can write a procedure that will, just as we did for Version 1 of Terrapin Logo. We'll call it MEMBERX.

```
TO MEMBERX :ITEM :LIST
IF EMPTY? :LIST [OUTPUT "FALSE]
IF :ITEM = FIRST :LIST [OUTPUT "TRUE]
OUTPUT MEMBERX :ITEM BUTFIRST :LIST
END
```

Here's how to use MEMBERX to see if a letter is in a word using Apple Logo. If you are using IBM Logo or Apple Logo II, substitute MEMBERP. If you are using Terrapin or Commodore Logo, substitute MEMBER? in place of MEMBERX.

```
PRINT MEMBERX "N "CANDLE
TRUE
PRINT MEMBERX "V "CANDLE
FALSE
```

Here's a procedure to compare all the letters in a guess with the letters in the key word "Greetings". You don't have to understand everything about how this procedure works to use it. It follows the same principles that were used to check words in the reindeer list.

```
TO COMPARE.LETTERS :GUESS :KEY.WORD
IF :GUESS = " [OP "TRUE]
TEST MEMBERX FIRST :GUESS :KEY.WORD
IFFALSE [PRINT (SE FIRST :GUESS [IS
NOT A LETTER IN] :KEY.WORD)]
IFFALSE [OUTPUT "FALSE]
OP COMPARE.LETTERS BF :GUESS :KEY.WORD
END
```

***In Terrapin Logo, use MEMBER? instead of MEMBERX and omit brackets around consequences. For example, type:

```
IF :GUESS = " OP "TRUE instead of
IF :GUESS = " [OP "TRUE]
```

Listful Thinking continued

Here's how to use COMPARE.LETTERS to check all the letters in a guess.

```
PRINT COMPARE.LETTERS "GRIPE "GREETINGS
P IS NOT A LETTER IN GREETINGS
FALSE
```

```
PRINT COMPARE.LETTERS "TIN "GREETINGS
TRUE
```

Now add instructions to GET.GUESS to check the letters in each guess with COMPARE.LETTERS. If the letters are OK, the GET.GUESS procedure goes on to see if the guess is already on the word list.

```
TO GET.GUESS
PRINT [WHAT IS YOUR WORD?]
MAKE "GUESS READWORD
TEST COMPARE.LETTERS :GUESS :KEY.WORD
IFFALSE [STOP]
TEST MEMBERP :GUESS :WORD.LIST
IFTRUE [PRINT [I ALREADY KNOW THAT
ONE.]
IFFALSE [ADD.TO.LIST]
PRINT []
GET.GUESS
END
```

***In Terrapin Logo, omit the brackets around consequences. For example, type:

```
IFFALSE STOP instead of
IFFALSE [STOP]
```

If the guess is still OK after it has been checked twice, it is added to the word list. If a player tries to use illegal letters in his guess, the program stops.

We hope you enjoy making lists and checking them twice. What's on our gift list? We would like to receive some of the games you make with these list tools.

Happy holidays!

PROCEDURES

Here are Terrapin Logo procedures for MEMBER? and READWORD.

```
TO MEMBER? :ITEM :LIST
IF :LIST = [] OUTPUT "FALSE
IF :LIST = " OUTPUT "FALSE
TEST :ITEM = FIRST :LIST
IFTRUE OUTPUT "TRUE
OUTPUT MEMBER? :ITEM BUTFIRST :LIST
END
```

```
TO READWORD
MAKE "INPUT REQUEST
IF :INPUT = [] OUTPUT "
OUTPUT FIRST :INPUT
END
```

A READWORD procedure is in the Apple Logo STARTUP file. In case it has been erased, here is a listing for Apple Logo.

```
TO READWORD
MAKE "INPUT READLIST
IF EMPTY? :INPUT [OUTPUT " ]
OUTPUT FIRST :INPUT
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

Teacher to Teacher

Logo, Math, and Art Books for Grade 4 through 12

reviewed by
Lou Wichert

Books Reviewed:

Discovering Apple Logo: An Invitation to the Art and Pattern of Nature and Computer Art and Animation: A User's Guide to Commodore 64 Logo, by David Thornburg, published by Addison-Wesley.

WHAT IF ...

...problem solving and creative thinking were priorities in the classroom?
...students were encouraged in projects of their own design?
...music and math, or art and science, were taught as one subject?

David Thornburg must have lived through these experiences! His books, Discovering Apple Logo and Computer Art and Animation, give teachers new tools to

Teacher to Teacher continued

experiment with and new approaches to teaching math and art with Logo. In both books, an abundance of teaching ideas are concisely and concretely presented in a manner sure to spark the creative mind.

As we read these books, I and the other members of our review group could not resist jotting teaching ideas in the margins as they cascaded into our minds. Thornburg offers dependable directions, creative ideas, and stimulating projects to try ... a find for busy classroom teachers!

Thornburg moves the reader from the beginning stages of Logo into the more complex manipulation of the language through specific examples and challenging questions, interspersed with a great sense of humor.

SIMILARITIES

Each book contains:

- *Basic beginning directions for getting started with the turtle and turtle graphics.

- *Introduction of primitives, color, procedures, variables, and editing commands.

- *Utility commands, such as SAVE.

- *X and Y coordinates with illustrations of uses.

- *Commands and operations which can be used with list processing, such as MAKE, PRINT, and SENTENCE.

- *Illustrations on how to incorporate a counter into a procedure.

- *Demonstrations of procedures with explanations given to help the reader understand the process.

- *Numerous ideas and projects to try.

- *An attractive layout, including generous margins and spacing, bold print, and clear crisp illustrations.

- *An easy to read style of writing.

- *Excellent resources for the Logo teacher who would like to move beyond static images.

DIFFERENCES

- *After the introduction, each book moves into an area of specialization, such as geometry, art, or computer animation.

- *Discovering Apple Logo has a geometry emphasis. It offers the Logo user an opportunity to make use of geometric principles while looking to nature as a guide.

- *Computer Art and Animation has an emphasis on animation, not surprisingly. The computer becomes a new tool for creative expression.

UPON CLOSER SCRUTINY

Thornburg states in Discovering Apple Logo that ancient mathematics has enchanted people for a millenium. He moves into this realm with the introduction of the golden mean and the Fibonacci series. What is the pattern when given the numerals 1 1 2 3 5 8 13 21 34 55 and so on? This numerical series is based on the golden mean. This pattern appears in many natural materials, such as pine cones or sunflower seed clusters. This is a fascinating subject for the math teacher who is interested in using Logo to illustrate this powerful concept.

Tessellations (patterns made from interlocking motifs) and star formulas are illustrated and explained.

Have you ever thought about how to measure a coastline? If you stretch our a piece of string to its fullest length, it is straight, and covers a specific distance. Take that same piece of string and make it curve and bend; notice that the distance it now covers is less. We have now entered the study of fractal geometry.

A fractal is defined as a mathematical set or concrete object whose form is extremely irregular and/or fragmented at all scales. Fractal geometry and the triadic Koch curve (a famous fractal curve) are used and explained in the study of a coastline.

Randomness in geometry allows the replication of the random patterns that we see in nature. Logo is the vehicle to use with students to illustrate these patterns.

An extensive annotated bibliography is included, and contains sections on turtle geometry, polygons and polyhedra, static symmetry, dynamic symmetry, tessellations, fractals, patterns in nature, and recursion.

In Computer Art and Animation, Thornburg moves beyond the basics of Logo into the realm of computer graphics. The kaleidoscope has always been a fascinating art form. The author shows us ways to create our own Logo kaleidoscope.

Thornburg describes multiple turtles (called "sprites" in Commodore Logo) and demonstrates how to move them on the screen. The art form now becomes animation, with helpful hints on how to do the animation and preserve the stories.

Exciting and innovative ideas to use with photography and video cassette recorders and a chapter on additional drawing aids complete the text. Creative teens love producing their own films; what better way to create good community public

Teacher to Teacher continued

relations than with a video production of Logo masterpieces?

This book is the culmination of Thornburg's "playing around" with computer art and animation, and describes what he learned along the way.

The final statement in the book says it: "Above all, have fun!"

IN SUMMARY

Thornburg has provided beginning Logo directions and moved into specialized areas. He has taken some often used school projects and revitalized them for the computer with Logo. He has given teachers a "shot in the curriculum" by asking:

...what better way to illustrate geometric principles than with the computer?

...why not use the computer as an art form?

...why not combine art and math? science and music?

We hope these stimulating texts continue. As teachers use them, they themselves could become the tools for curriculum change.

Additional books by David Thornburg.

Addison-Wesley, 2725 Sand Hill Road, Menlo Park, CA 94025.

Computer Art and Animation: A User's Guide to TI/99-4A Color Logo. 1983, \$12.95.

Computer Art and Animation: A User's Guide to TRS-80 Color Logo. 1983, \$12.95.

Picture This! An Introduction to Computer Graphics for Kids of All Ages. 1982, \$14.95.

Picture This Too! An Introduction to Computer Graphics for Kids of All Ages. 1982, \$14.95.

Practice This! An Introduction to Computer Graphics for Kids of All Ages. 1982, \$14.95.

Compute Publications, PO Box 5406, Greensboro, NC 27403.

Every Kid's First Book of Robots and Computers. 1982, \$4.95.

Please send your comments, questions, and concerns on this review to Logo Information For Teachers (LIFT), PO Box 5396, Plymouth, MI 48170.

Lou Wichert is a media specialist in the Farmington (MI) Schools system and an educational consultant.

Q and A

by

Jim McCauley

Q. Why do some Logo people advise against the use of MAKE and global variables? I find them really useful.

<[-?->

A. Computer languages always offer some way of "binding" values to variable names so that the programmer can work with the name of some value rather than the value itself. This allows the same program to produce different outputs when different values are associated with the variables within it.

LETTING IT, MAKING IT

In BASIC, the programmer has only one way of binding values to variables: by using the equals sign to assign new values to variable names. For example, this line in a BASIC program:

```
100 LET A = A + 1
```

assigns a new value to the variable name "A". The new value depends on the previous value; if A is bound to 3, running this line would assign the value 4 to A, and running it again would assign 5 to A. This line is an example of a "global" assignment statement in BASIC; when the value of A is changed in this way, the new value applies to every occurrence of A in the entire program.

The MAKE command allows one to do the same thing in Logo:

```
MAKE "A :A + 1
```

has the same effect as the line of BASIC above. If the value of :A is 3, then after this line is run, :A has a new value of 4, and that new value will be true for every :A in any Logo procedure in memory.

ANOTHER WAY

There is another way of binding values to variables in Logo that BASIC and some other computer languages lack. Logo is a procedural language, and its procedures accept inputs. For example:

Q and A continued

```
TO ADD1 :NUM
PRINT (SENTENCE :NUM [PLUS 1 IS ] :NUM + 1)
END
```

This procedure accepts a number as input, binding it to the variable name :NUM, and prints a sentence about what happens if you add 1 to :NUM. You might think that this binding is the same kind as MAKE produces, but that's not the case; as soon as ADD1 finishes executing, there is no longer anything bound to :NUM.

If :NUM occurs in some other procedure in the workspace, whatever is bound to :NUM in ADD1 won't interfere with it. The binding to :NUM is local; it applies only within ADD1 and any procedures that ADD1 might call on while it is running. ADD1 contains only primitives, but if it did call on another procedure, :NUM would still carry the value passed to it when ADD1 was called, unless a new value for :NUM is "sent" to the procedure called from within ADD1.

The ability to bind values locally (instead of only globally) is one of Logo's great strengths, because it allows us to harness the power of recursion. Here's a simple tail-recursive variation on ADD1:

```
TO ADD2 :NUM
IF :NUM > 50 [STOP]
PRINT (SENTENCE :NUM [PLUS 2 IS ] :NUM + 2)
ADD2 :NUM + 1
END
```

This will add 2 to the number bound to :NUM and tail-recursively "call itself" to continue adding 2 to the sum of :NUM and 1 until :NUM is greater than 50, when it will stop. Notice that we didn't have to use MAKE to assign a new value to :NUM; passing it tail-recursively to the new copy of ADD2 took care of that automatically. Here's a somewhat trickier variation:

```
TO ADD3 :NUM
IF :NUM > 50 [OUTPUT :NUM]
OUTPUT (SE :NUM [PLUS 3 IS] ADD3 :NUM + 3)
END
```

Type this in, and then call it with PRINT ADD3 40. The result resembles something children who have just learned to add use to drive their parents crazy at the dinner table.

THE MAIN MATTER WITH MAKE

The real problem with excessive use of MAKE is that one tends to carry over old programming habits from other languages (especially BASIC), thus missing the elegance of the parameter-passing style of programming and use of recursion that is so natural to Logo.

Use of MAKE inside recursive programs is potentially dangerous; the usual consequences are mysterious bugs related to confusion of local and global bindings. In addition, MAKE statements have a voracious appetite for memory. It is not unusual for a recursive procedure with MAKE to produce an "out of memory" error message.

I like to encourage my students to bind values to variables as inputs, rather than using MAKE, because of the clarity and elegance of that style of programming. Also, if my students have learned to program another language that makes heavy use of the assignment statement (like BASIC or Pascal), I find that this helps them make a clean break from previous ideas about programming that may cripple their capacity to learn more powerful and flexible techniques in Logo.

(Ed. Note: If you have questions about Logo programming techniques or the thinking which leads to Logo programming, send them to NLX Q & A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

NLXual Challenges

by
Robs Muir

Of Sunshine and Shadow

One of the more exciting developments in Logo-based education is the concept of a microworld - a tightly controlled environment designed to encourage exploration of a specific idea or unit of curricula. A popular Logo microworld is the dynaturtle, a turtle consciously designed to simulate an object that obeys Newton's Laws of Motion

NLXual Challenges continued

(see Andy diSessa's "Unlearning Aristotelian Physics: A Study of Knowledge-based Learning," Cognitive Science 6, 1982).

Perhaps the most well known Logo microworld is the turtle itself; turtle graphics represents a controlled environment intended to encourage discoveries of geometric and mathematic ideas. In fact, Logo could exist without the turtle microworld; the language just wouldn't be quite so graphic.

Seymour Papert calls microworlds "incubators for knowledge." It is a pity that there aren't more well-developed microworlds such as dynaturtles to use in our classrooms.

LONG AGO IN A MICROWORLD FAR AWAY

This month's problem is an attempt to create a SUNWALKER microworld. Here's the scenario...

Tom Lough, walking with his family late one afternoon, observed his son, Kyser, discover his shadow. On his short little legs, Kyser began chasing his shadow across the lawn. After finally catching him, Tom began to wonder what Kyser's path would look like if he were allowed to follow his shadow from sunup to sundown, and suggested this problem as a result.

Since the sun's position changes during the day, the direction of Kyser's shadow would also change. Can you intuit Kyser's path? Now write a Logo program that simulates our SUNWALKER's path.

Here's a possible general method. Set the turtle in the upper right corner of the screen with the pen up. Guide it horizontally across the screen, simulating the sun traveling from east to west. After each incremental movement of the sun turtle, move it to the walker's position, give it

an appropriate heading, move it FD a few steps, and return to the sun position.

Perhaps (but not necessarily) the solution could make use of those little known primitives, LABEL and GO.

LCSI Logos can invoke the WINDOW command and set the sun far away (SETPOS [10000 10000]), and then start it moving westward. Sprite Logos, such as Commodore's, make it easier to code a solution; one sprite could be the SUN, while a second becomes the WALKER.

Is your guess about the walker's path borne out by the turtle? Remember that the sun doesn't travel east to west in the summer in the Arctic; it makes a complete circle about the sky. What path would the turtle "walk" if the sun orbited around HOME in a screensized circle?

When you have a working solution (or even a valiant attempt), send your solution, along with a self-addressed stamped envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

We will compile the contributed solutions, select the most interesting or unusual, and redistribute them to each contributor. Since our "challenge community" involves problem solvers from six countries, the contributions are quite diverse. Solutions are available for each problem, but you must contribute your attempted solution (however meager) to receive these "answers."

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

The National Logo Exchange, copyright (c) 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, \$25 (US) per year, mailed first class from Charlottesville, VA. \$5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of \$2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.....Tom Lough

NLXTRA

前 100

by
Hillel
Weintraub

Last month, I promised a review of Epson's Mighty Logo, but it arrived too late for me to write about this month. It is still "on the stack," however.

Instead, let me tell you about another Logo version I have been using. It is for the Epson PX-8, a portable computer being marketed in the USA and Europe under the name Geneva. It has been great fun to sit on the train and write programs in Logo!

This is a prototype version, and Epson is still considering various applications for it. If you have any ideas for capabilities you would like to see incorporated into a portable Logo, drop me a line, and I'll pass your suggestions along to Epson. They would be interested in knowing what kind of applications NLX readers envision for such a system.

LOGO VERSUS THE JAPANESE EDUCATIONAL SYSTEM

Let me tell you something of Logo's future in Japan and the general direction of the educational use of computers here. This past summer, while visiting the USA, I talked with Prof. Papert about Japan and Logo. He commented that, in France, the national educational committee had just pronounced Logo the preferred language for primary school children. I got the impression that this didn't just happen overnight, but, rather, was accomplished after a lot of behind-the-scenes work by a lot of dedicated people.

Prof. Papert felt that Japan and France had much in common, due to the centralized educational bureaucracy (called Mombusho in Japan). The kind of ground work done in France, he said, was also necessary in Japan in order to help the Mombusho see the value of using computers in ways other than just drill and practice. He felt that the Logo community should join forces with other like-minded groups of people interested in the ideas of Montessori, Piaget, Dewey, or progressive education in general. That community would then grow large enough and influential enough to affect public education decisions.

After I returned to Japan, I talked about Prof. Papert's idea with Hiroyoshi Goto, of the BYNAS Division of the UNY Company. More than anyone else in Japan, I believe, Goto-san has been working to bring Logo to the attention of the public, especially educators. He agreed with Prof. Papert's idea; in fact, BYNAS has just opened an office in Tokyo (their home office is in Nagoya). One of their main purposes is to help distribute their Logo Journal and disseminate information about Logo.

Living in Japan will do nothing if not teach one patience. That is just what it will take, along with a lot of hard work, to see any real inroads in the very conservative approach to education which is predominant in Japan. Even more than the Mombusho, the university entrance examination system is a force which keeps educational innovation and experimentation to a minimum. This test is the sole basis upon which a student's entrance to a university is decided.

Because the "name" of the university one attends is one of the most influential factors in being selected by a company (and thus being set up in a lifetime job), the students themselves often constitute the most conservative force! "Don't give me that creative stuff; just tell me what I need to memorize for the test!"

As you can see, there is a long way to go, but that is one of the things that makes living in Japan interesting! Here, our motto is MAE 100!

(Ed. note: Send your ideas on the portable Logo to Hillel at Doshisha International High School, Tatara, Tanabe-cho, Tsuzuki-gun, Kyoto-fu 610-03 Japan.)

Hillel Weintraub is a teacher at Doshisha International High School in Kyoto-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

Dr. Logo is getting an enthusiastic reception with European computer manufacturers, according to a recent issue of Digital Research News. The manufacturer of the Apricot computer includes a copy of Dr. Logo with each machine purchased. Two other companies have announced similar plans.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

Can Logo programming serve as an effective teaching tool in elementary school mathematics? This question served as the focus of a study which investigated Logo programming as one way to facilitate the understanding of number sequences and variables.

Learning to program involves active learning and can be highly motivating. Through programming activities, children can express their abilities as well as develop problem solving skills.

Eighteen 5th grade children who had no prior programming experience participated in the fifteen week study. The first five weeks were spent in learning the basics of the Logo language.

The next five weeks were devoted to learning to write procedures. During this time, the children were grouped on the basis of ability and randomly assigned to one of three instructional groups. Each group was given programming tasks along with varying amounts of information for completing each task: a complete sequence of steps to carry out, an incomplete computer program, or no additional information.

The last five week period was spent on programming tasks without any information other than the task itself.

Results indicate that programming can be a useful teaching tool in elementary mathematics. Children in the experimental group were able to write programs successfully, and demonstrated an ability to use variables. From a practical standpoint, the programming experiences facilitated the development of problem solving behaviors and served to "turn on" some children who seemed less motivated in the regular classroom environment.

>>>Milner, S. D. (1973). The Effects of Teaching Computer Programming on Performance in Mathematics (Doctoral dissertation, University of Pittsburgh, 1972). Dissertation Abstracts International, 38/08, 4183-A.

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertations and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

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

PRINT-IT! is a plug-in card which provides the Apple computer with a Print Screen capability similar to that of the IBM PC. This allows you to make a printer copy of any screen display, whether it is graphics, text, or both. All you have to do is push a button, then press <return>.

Working with both serial and parallel printers of several dozen models, the PRINT-IT! card allows the selection of several printing options, including both high and low resolution, page 1 and page 2 side by side, and expanded, inverted, or rotated images.

The PRINT-IT! card comes with an external button, a printer cable, and a concise clearly written 25-page manual. The price is \$199. For more information, write Texprint, 220 Reservoir Street, Needham Heights, MA 02194, or call (617) 449-5808.

NLX subscriber and author Sharon Burrows reports that teachers in her school system are having difficulty with Apple Logo and the Apple IIc computers. It seems that Apple Logo disks purchased more than a year or two ago do not boot on the IIc.

If you are in this situation and desire to run your Apple Logo on a IIc, send your old disk with \$15 to Logo Computer Systems, Inc. (LCSI), and they will send you a replacement which will boot.

By the way, LCSI has a new address and telephone number. LCSI, 555 West 57th Street, Suite 1236, New York, NY 10019. (212) 765-4780.

Krell Software Corp. announces its Waterloo Logo for the IBM PC. The package includes a 3-disk set and documentation for \$99.95. For more information, write Krell Software Corp., 1320 Stony Brook Road, Stony Brook, NY 11790, or call (800) 245-7355.

LOGO 85

FIRST ANNOUNCEMENT/CALL FOR PAPERS

The Second International Logo Conference
Massachusetts Institute of Technology
Cambridge, MA

July 22-25, 1985

LOGO 85 is being organized to give people actively engaged in Logo work a continuing opportunity to share what they have learned, developed and discovered. In order to facilitate the building of networks, the conference will provide a variety of settings: major speakers, presentations, and special interest groups.

CALL FOR PAPERS: Themes to be considered during the conference include evaluation, teacher education, and the integration of Logo into the curriculum. A short paper (not more than 5 pages) outlining the major features of the presentation should be submitted to:

Professor Hal Abelson
LOGO 85 Program Co-Chair
MIT Laboratory for Computer Science
NE43-410
545 Technology Square
Cambridge, MA 02139

no later than 15 February 1985. Papers selected for presentation will be published in the conference pre-proceedings. Proposals from school children are encouraged.

For general information and registration information, please contact:

Special Events Office
Massachusetts Institute of Technology
Room 7-111
Cambridge, MA 02139