THE NATIONAL LOGO EXCHANGE

Volume 3 Number 6 FORWARD 100! February 1985

Wordles and Turtles

by
Cathy Frank

"Mom, look at my wordles!" My fifth grade daughter proudly held up for my inspection a page full of work. At first, I didn't understand just what a "wordle" was supposed to be. But once I caught on, I became as enthusiastic as she about them.

You see, I enjoy introducing students to the "other half" of Logo, word and list operations, well before they have mastered all the turtle commands. Among other things, this enables me to use Logo with several curriculum areas besides the obvious ones.

However, it is not always easy to think of effective introductory projects in list processing. Some are too simple and unimaginative. Others are too difficult, requiring knowledge of several sophisticated commands. After my daughter's enthusiastic explanation, I realized how much fun, open ended, and challenging "wordles" would be as Logo list processing projects!

OK, WHAT'S A WORDLE?

A wordle is an incomplete word or group of words whose full meaning is implied by the way the words or letters are written. A simple example is

```
T
O
W
N
```

The intended word is DOWNTOWN, DOWN being implied by the way the letters of TOWN are printed down the page. Another more challenging example is

```
RIGHT DINNER
```

which is wordle-eze for RIGHT BEFORE DINNER! The possibilities are endless, as are the opportunities to use a variety of Logo list processing commands in procedures to display wordles.

The two examples given above use only the PRINT command.

```
TO WORDLE.1          TO WORDLE.2
PRINT [T]            PRINT [RIGHT DINNER]
PRINT [O]            END
PRINT [W]
PRINT [N]
END
```

It didn't take my students long to write another possibility for DOWNTOWN which used the REPEAT statement they had learned with the turtle.

```
TO WORDLE.3
REPEAT 4 [PRINT [TOWN]]
END

TOWN
TOWN
TOWN
TOWN
```

Another simple wordle is

```
TO WORDLE.4
PRINT [STAND]
PRINT [I DON'T]
END
```

I'll let you figure that one out.

If you use Terrapin Logo, PRINT [' '] is useful.

```
TO WORDLE.5
PRINT ['      T      ']
PRINT ['    *   H    ']
PRINT ['  *       E  ']
PRINT ['*           *']
PRINT ['  K       C  ']
PRINT ['    C   L    ']
PRINT ['      O      ']
END
```

Wordles continued

The CURSOR (or SETCURSOR) command helps expand wordle possibilities. Here are a couple of examples.

```
TO WORDLE.6
CURSOR 10 10 PRINT [S]
CURSOR 12 11 PRINT [T]
CURSOR 14 12 PRINT [A]
CURSOR 16 13 PRINT [I]
CURSOR 18 14 PRINT [R]
CURSOR 20 15 PRINT [S]
END
```

```
S
  T
    A
      I
        R
          S
```

```
TO WORDLE.7
CURSOR 14 10 PRINT [P]
CURSOR 14 11 PRINT [U]
CURSOR 14 12 PRINT [Z]
CURSOR 11 13 PRINT [PUZZLE]
CURSOR 14 14 PRINT [L]
CURSOR 14 15 PRINT [E]
END
```

```
          P
          U
          Z
       PUZZLE
          L
          E
```

Hardcopies of the above results can be printed, of course, by typing OUTDEV 1 (or (or .PRINTER 0) when you are finished.

DYNAWORDLES

As my students and I developed more examples, it occurred to me that there are wordles one could program which are unique to Logo and the computer; they would not be effective on paper. Here is an example which incorporates subprocedures, the REPEAT and CURSOR commands, variables, and recursion.

```
TO WORDLE.8
REPEAT 30 [FLASH]
END

TO FLASH
CURSOR 20 10 PRINT [AND ON]
WAIT 30
CURSOR 20 10 PRINT [ ]
WAIT 30
END

TO WAIT :TIME
IF :TIME = 0 STOP
WAIT :TIME - 1
END
```

What makes this exercise exciting to me is that writing wordles is fun and challenging with or without Logo. But doing it with Logo expands the wordle possibilities and adds a motivation for learning some of the word and list processing commands whose value is not always appreciated by the beginning list processor. Yet, the challenge is unlimited and open ended and can lead into the use of some rather sophisticated Logo commands.

To wordle a popular expression of a certain newsletter editor,

```
TO WORDLE.NLX :A
IF :A = 35 STOP
CURSOR :A 10 PRINT [100!]
WAIT 5
CURSOR :A 10 PRINT [ ]
WORDLE.NLX :A + 1
END
```

Just in case you had some trouble deciphering the wordles: WORDLE.4 - I don't understand, WORDLE.5 - around the clock, WORDLE.6 - downstairs, WORDLE.7 - crossword puzzle, WORDLE.8 - off and on, WORDLE.NLX - FORWARD 100!

(Ed. Note: For additional related ideas, see also "Is There Logo After Turtle Graphics?" Classroom Computer News, April 1983, and "Tipps for Teachers" and "Microworlds" in the February 1983 issue of The National Logo Exchange.)

Cathy Frank is a computer education consultant in Burlington, VT, and teaches Logo to 5th and 6th graders.

Pascal Newsletter Available

If you are using Logo as an introduction to Pascal, you may be interested in the Advanced Placement Computer Science (APCS) Newsletter published by Tim Corica. Its ten issues per year address Pascal questions, the AP exam, utilities, and the like. For more information, write Tim Corica, APCS Newsletter, The Peddie School, Hightstown, NJ 08520.

From the Editor

Stirring the Pot

One day recently, I stomped into the house half frozen. Posy had just made a pot of vegetable soup and she pressed a bowl of the steamy stuff into my grateful hands as I sank into a recliner. Ah, bliss. When I am subjected to such a dreamlike state, my mind often takes on a life of its own.

I began thinking about soup.

The soup I was enjoying had many different ingredients: tomatoes, potatoes, peas, beans, corn, chunks of meat, seasoning. Although I was aware of each different ingredient as I savored it, there was was a single taste sensation for the soup itself.

How did this come about? While Posy was cooking the soup, she had to stir it often so that it was heated evenly and nothing stuck to the pot and burned. Some of the flavor of each ingredient was absorbed by the others, but individual identities were retained.

But what does this have to do with Logo?

The Logo community is certainly made up of many different types of individuals. They share a common interest in the future of educational computing with this powerful language.

But, unless the "Logo broth" is stirred frequently, it may not pass the taste test. I'd like to encourage the healthy exchange of differing points of views, the airing of opinions and recommendations. For example, John Green's article, "Logo Under Fire," in the January issue of Classroom Computer Learning, gives an excellent focus. It contains an interview with Seymour Papert and suggests that confronting controversial issues is crucial.

But more such articles is not the full answer. Action is needed. Action such as more direct communication between Logo folk, at conferences and workshops, in faculty workrooms (I refuse to call them "lounges") and classes. Action such as letters to authors, publishers, manufacturers, with detailed suggestions, comments, complaints. Action such as each of us assertively trying to find the best way for us and our students to take advantage of the potential of Logo.

Stir the pot a little; it makes a better soup.

FD 100!

Tipps for Teachers

by Steve Tipps

Survey of Surveys

Every newspaper, magazine, and news broadcast seems to include a survey of some kind. What are the best sellers? Who is winning the election? How many people have quit using salt? Everybody likes to have their opinion counted, including students. This self interest can spark many classroom activities using surveys.

Surveys have two main parts: data collection and data display. Logo can be used for both.

GATHER YE INFORMATION

Many elementary mathematics texts include a section on surveys. Students ask their classmates about things such as favorite pets, TV shows, records, colors, etc. Classroom surveys can also be taken on kinds of shoes, colors of eyes, types of transportation to school, months of birth, and 1001 other topics of personal interest. Although the approach may differ, the topics can be used in kindergarten, sixth grade, high school, or college.

In taking a survey, students scurry around the room asking classmates questions. The first attempt may be a quite a mess, but, after this experience, the need for precise data recording is obvious.

Additionally, the way the question is asked makes a lot of difference in the results. The difference between open and fixed categories is clear. An open question is, "What is your favorite color?" The fixed or closed form asks, "Is your favorite color red, blue, green, or yellow?"

If the students want to ask the question of many people, an interactive computer program might be a good tool. Here is a probe for the favorite breed of dog.

```
WHICH IS YOUR FAVORITE KIND OF DOG?
     POODLE
     SPANIEL
     SHEPHERD
     COLLIE
     POMERANIAN
     OTHER
TYPE IN YOUR ANSWER.
```

Tipps continued

Here is another for birthday month.

IN WHICH MONTH WERE YOU BORN?
PLEASE TYPE IN YOUR MONTH.

Writing procedures for these probes is fairly easy. The logic is the same as that used when asking classmates.

Ask a question of a friend.
Check to see which category the answer fits.
Add a tally mark to that category.
Thank your friend.
Ask a question of someone else.

The skeleton program would include the same parts.

```
     TO SURVEY
     QUESTION
     CHECK   ; includes thanks
     SURVEY
     END

     TO QUESTION
     PRINT [IN WHICH MONTH WERE YOU BORN?]
     PRINT [PLEASE TYPE IN YOUR MONTH.]
     MAKE "MONTH FIRST REQUEST
     END

     TO CHECK
     IF :MONTH = "JANUARY MAKE "JAN :JAN + 1
THANKS STOP
     IF  :MONTH = "FEBRUARY MAKE "FEB :FEB +
1 THANKS STOP
     ......
     ......
     IF :MONTH = "DECEMBER MAKE "DEC :DEC  +
1 THANKS STOP
     PRINT  [ARE  YOU  SURE YOU TYPED  IN  A
MONTH?]
     PRINT [CHECK YOUR SPELLING]
     END

     TO THANKS
     PRINT  [  THANK YOU FOR  ANSWERING  THE
SURVEY.]
     BIRTHDAY.CAKE  ;  a turtle drawing of a
birthday cake
     END
```

Before starting the survey, all the values for the months need to be set at 0. This initialization process takes place before SURVEY starts by typing SETUP.

```
     TO SETUP
     MAKE "JAN 0
     MAKE "FEB 0
     ....
     ....
     MAKE "DEC 0
     END
```

When the survey is finished, check the totals by printing out the names of the variables.

```
     PO NAMES
     "JAN IS 10
     "FEB IS 8
     ....
     ....
     ....
     "DEC IS 12
```

Save the results of the survey by saving the workspace in a file which shows the date or group.

```
     SAVE "BDAY.SMITH.CLASS
     SAVE "BDAY.2.10.85
```

You can use the same survey another day in another class or with another group of people. At the end of two weeks you have information from several sources.

MAKE IT BETTER

The survey program is only a skeleton which has many ways to be improved. The way QUESTION and THANKS looks on the screen can be changed with CURSOR and better spacing. A picture, a pause, and more information are all good improvements.

The programming also has much that can be improved. The skeleton program has not used list processing. Instead of checking the :MONTH twelve different times, you could check it against a list of months.

```
     MAKE  "MONTHS  [JANUARY FEBRUARY  MARCH
APRIL MAY JUNE JULY AUGUST SEPTEMBER OCTOBER
NOVEMBER DECEMBER]
```

SETUP.LIST starts each month out with an empty list.

```
     TO SETUP.LIST
     MAKE "JANUARY []
     MAKE "FEBRUARY []
     ....
     MAKE "DECEMBER []
     END
```

Tipps continued

Now the BETTER.SURVEY can use a better way to check the lists.

```
TO BETTER.SURVEY
QUESTION
BETTER.CHECK
BETTER.SURVEY
END

TO BETTER.CHECK
TEST MEMBER? :MONTH :MONTHS
IFTRUE TALLY
IFTRUE THANKS
IFFALSE TRY.AGAIN
END
```

If the month is found, a tally is added to the list which is attached to the month name.

```
TO TALLY
MAKE :MONTH FPUT "X THING :MONTH
END
```

THANKS and TRY.AGAIN are simple or improved variations from the old SURVEY.

The results of the survey can be seen by printing out names from BETTER.SURVEY.

```
PO NAMES
"JANUARY IS [X X X]
"FEBRUARY IS [X X]
...
...
"DECEMBER IS [X X X X]
```

The results can also be saved in a file. It may be handy to ERASE all the procedures and SAVE a file which has only the variables names and values in it.

```
ERASE PROCEDURES
SAVE "MONTHS.SMITH.CLASS
```

HOW MANY APRIL BIRTHDAYS?

The number of birthdays for each month can be printed out with PO NAMES. This display is not very pretty. A nicer way to show the number of birthdays would be with a display on the text screen.

Since SURVEY counted January birthdays with the variable JAN, the value of the variable can be used as an input to REPEAT. To see how many January birthdays are in the group surveyed, just type

```
REPEAT :JAN [ PRINT1 "J ]
```

Creating a display of all the birthday months requires a separate line for each month. Use the cursor to create a display on the TEXTSCREEN.

```
    TO DISPLAY
    NODRAW
    CURSOR 0 10  PRINT1 "JANUARY
    CURSOR  12 10 REPEAT :JAN  [PRINT1  "J]
PRINT [ ]
    PRINT1 "FEBRUARY
    CURSOR  12  11 REPEAT :FEB [PRINT1  "F]
PRINT [ ]
    PRINT1 "MARCH
    CURSOR  12 12 REPEAT :MAR  [PRINT1  "M]
PRINT [ ]
    ....
    ....
    ....
    ....
    PRINT1 "DECEMBER
    CURSOR  12  21 REPEAT :DEC  [PRINT1  "D
PRINT [ ]
    END
```

Calling DISPLAY gives a chart such as this one.

```
JANUARY     JJJ
FEBRUARY    FFFF
MARCH       MM
APRIL       AAAAA
MAY         MMMM
JUNE        JJ
JULY        JJJ
AUGUST      AAAA
SEPTEMBER   SS
OCTOBER     OO
NOVEMBER    NNNN
DECEMBER    DDD
```

BETTER.SURVEY collects X'S for each answer; this provides an easier way to write a display procedure. BETTER.SURVEY uses recursion to work through the list of MONTHS and set the LINE for the display.

```
TO BETTER.DISPLAY :MONTHS :LINE
IF :MONTHS = [] STOP
CURSOR 0 :LINE
PRINT1 FIRST :MONTHS
CURSOR 12 :LINE
PRINT THING FIRST :MONTHS
BETTER.DISPLAY BUTFIRST :MONTHS :LINE + 1
END
```

BETTER.DISPLAY shows an X for each birthday from BETTER.SURVEY.

Tipps continued

JANUARY	X X X
FEBRUARY	X X X X
MARCH	X X
APRIL	X X X X X
MAY	X X X X
JUNE	X X
JULY	X X X
AUGUST	X X X X
SEPTEMBER	X X
OCTOBER	X X
NOVEMBER	X X X X
DECEMBER	X X X

BEYOND THE SURVEY

Try making a survey to collect information. The topic of the survey and the sophistication of the programming are adjustable to any level. If you find that students need a Logo tool which they cannot invent for themselves, you can provide a skeleton program as a hint for elaboration. After you have them hooked on surveys, it is time for another look at how to organize and compare the results. That will be the topic for next month's column.

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.

Terrapin Utilities II Disc Announced

Terrapin announces the release of the Utilities II Disk for Terrapin Logo. The disk contains over two dozen files for application in many areas of Logo use. Included are programs to print pictures from Terrapin Logo with the Apple Imagewriter, drive the Apple Color Plotter, load and save KoalaPad pictures (it works with Power Pad pictures, too), draw low resolution graphics, write text characters on the Logo graphics screen, and a translator to run Apple Logo files. Although Terrapin versions 1.3 and 2.0 are recommended for the Utilities II disk, a USEFUL.PROCS file is furnished so that earlier versions may be used. The disk is accompanied by excellent documentation. To order, send $20 to Terrapin, Inc., 222 Third Street, Cambridge, MA 02142, or call (617) 492-8816.

NLXionary

A Lectionary of Selected Logo Readings

Commentary by
Robbie and Griff Wigley

"To Sprite or not to Sprite?" by Jim Muller, Turtle News, Young Peoples' Logo Association, March 1983.

>We resurrect this two year old article from the archives because it seems the sprite vs. turtle issue is still current. Several versions of Logo on the market implement sprites or give turtles sprite features. Teachers must make decisions on how to approach them.

>Jim Muller addresses the article mainly to Apple owners who wonder whether it's worth the extra cost to add a sprite board to their computer. He adds that others must concern themselves with how to use their sprites if they are provided in their Logo version. "Electronic deficiencies" of sprite implementation are a source of irritation to Muller. For example, the "no more than four in a row" limitation, being unable to draw (TI Logo), lack of shape rotation as orientation changes, and sprite commands different from those of the turtle. He contends that these machine limitations prevent the "no threshold, no ceiling" experience so critical to Logo.

TWO BY TWO

>He compares the work of two students, one a virtuoso of sprite flashiness without any creative thinking, the other a visualizer of results while developing turtle procedures without reference to the display screen. He wonders which has progressed at his/her own level of understanding; which has progressed from one dimensional to multi dimensional thinking; which has come to see some of the power of imagination. He cites examples of students who, "after learning that virtually anything can be done in turtle graphics, ...have become frustrated with the limits imposed by sprites. It is like once tasting the excitement and strategic limitlessness of chess, you are told you must play only checkers."

>In answer to those complaining of boredom with turtle graphics, Muller suggests versions of Logo with multiple tur-

NLXionary continued

tles, such as Radio Shack's Color Logo. He sees a smooth transition between single and multiple turtles, as opposed to the "total break in symmetry for the purpose of animation" with sprites. Each turtle can draw and be assigned a different shape that will automatically turn in the direction assigned. He believes these features are legitimate extensions of the original concepts presented in initial implementations of Logo. And he sensitively worries about physically and developmentally disabled students who, having tasted success with turtle graphics, must contend with machine limitations and the confusing features of sprites. "...what problems might arise from having to explain why the computer won't do what they want it to do? ...the need for symmetry becomes crystal clear."

SYMMETRY IS IMPORTANT

>We agree with Muller that the additional features of a Logo version such as Color Logo extend traditional turtle graphics in a consistent fashion. The "symmetry," as he puts it, is clearly a desirable feature that would eliminate some of the confusion as a student moves to more varied uses of the language. However, we can't see how telling a student that a computer can't do what he or she wants is going to cause them major problems. Muller nearly becomes patronizing of students with learning difficulties at this point. Students could be told that the computer can't handle their good ideas yet. Finding a creative way around a computer's limitations can even be exhilarating.

>While multiple turtles are a desirable extension for Logo versions, Muller must admit that some students will become immediate mode virtuosos of these Logo versions, too. Most students (and even some teachers!) treat potent learning tools in trivial ways at times. When a student spends session after session designing shapes for the multiple turtles to "carry," yet doesn't progress to building procedures, will Muller blame Radio Shack? We think it better to assess the needs and abilities of the student, as well as the type of instruction he or she is receiving than to blame the hardware.

A PERSONAL PERSPECTIVE

>We both have taught Logo for nearly three years. Much of our early teaching was in camp and learning center settings where students' learning was of the short and intense variety during non-school hours. For the last year and a half, we have taught Logo at a small independent K-4 school during the school day. The overall student / computer ratio is 4 to 1, with computers both in a lab and in individual classrooms. The major mistakes with our early teaching was our tendency to hurry...the "got to give 'em their money's worth" kind of approach. And we immediately concurred with those who said, "Don't start them out with sprites - they're so flashy and fun that they'll never learn the important programming concepts." But now the pressure to hurry (real or imagined) is off. Our present students will be around for a while. Learning is more individually paced. And we think differently now about sprites.

>Until we introduced the sprites, we were forever using "turtle activities" for the younger group to keep them interested: mazes, shoot-at-the-targets, copy-the-drawing, play turtle, and whatnot. In our minds, we were doing it to prepare them better for "the real Logo world." It has since dawned on us that education has always done that number on students, and here we were doing it in the name of Logo.

SPRITES BEFORE TURTLES?

>Sprites seem to be a better environment than the turtle (instant or otherwise) to explore for many first and second grade students. Colors, shapes, directionality, and movement are all a big deal to these students. A sprite environment easily lends itself to exploration and creation without having to devise endless "props" to keep them interested. They are concrete and fun, machine limitations not withstanding.

>To us, third and fourth graders generally seem most ready for the full Logo language, though we wouldn't deny that some children can dive in with sustained interest prior to that. We do think that it's easier to teach many fundamental Logo programming concepts with turtle graphics. But many students are more motivated to put those concepts to use with sprites than with the turtle. We coach them in their work with both.

>As the computer technology develops, hardware limitations and distinctions between sprites and turtles will fade. Soon, every turtle will be a dynamic turtle, with touch, speech, and voice recognition capabilities. But questions of how to teach what and why will remain. We, too, need environments to explore.

NLXionary continued

(Authors' Note: We acknowledge that much has changed in the two years since Muller's article was written. We invite a response from the author and other NLX readers. Our address is 918 College Street, Northfield, MN 55057.)

Robbie and Griff Wigley are co-owners of Family Computing, Inc. Robbie teaches Logo to grades K - 4 in Prairie Creek Community School in Northfield, MN. Griff is a project facilitator in the Faribault Public Schools, and conducts Logo workshops for teachers throughout the state.

TurtleTips

by
Donna Lanyi
and Jane Toth

Have you discovered the creative students in your classroom? Some of these students may be unexpected shining stars at the computer. Good for them! Students' abilities and talents with turtle graphics are growing quickly after five months of school. In order to meet their needs, you need to be prepared to work with students at various levels (a Logo understatement!). We have found two ways to help students at both ends of the spectrum: task cards and the PRINT command.

TEACHER HINTS

There are always a few students who are unable to be productive at the computer using the discovery methods of Logo. These students need guidance on what to try next. Using task cards helps to give these students a base from which to work. To assist these students, we have developed task cards which contain certain Logo concepts.

Each card has the students try a few tasks, record what they see, and then use these ideas to make a picture or project of their own. Task cards can also be used with students at the top end of the spectrum in order for them to experience new concepts.

If you do not want to develop your own Logo task cards, see the "Teacher to Teacher" column in the October 1984 issue of the NLX for a review of a half-dozen sets which are available commercially. (Ed. Note: In addition, EduComp has just published sets of cards for primary and elementary use. EduComp, 14242 Wyeth Avenue, Irvine, CA 92714.)

For students who want to go beyond turtle graphics skills, introduce the PRINT command. PRINT makes it possible for students to write poetry, stories, letters, jokes, spelling words, or vocabulary words. It opens up a whole new world of Logo to your students. The SETCURSOR (or CURSOR) command adds the pizzazz, as Cathy Frank shows in her Logo wordles article on page 1 of this issue!

PROCEDURE OF THE MONTH

HAPPY VALENTINE'S DAY! Encourage students to create valentines with Logo. Hearts are not easy to draw, but are fun to try. Use the NLX ABC's to write a valentine message on the drawing. Print the valentine drawing out on paper, if possible. If you are interested in a free listing of a sample heart procedure, please send us a self addressed stamped envelope.

We invite elementary teachers and students to send 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.

Logo Notes

Microquests, a newsletter featuring a fully developed monthly Logo adventure, promises many stimulating ideas for projects. Powerful ideas and concepts from science and mathematics are presented and discussed in each issue. Published by Kathleen Martin and Donna Bearden, well known Logo authors, Microquests is available for $25 per year (September through May). Write to Microquests, PO Box 337, Grapevine, TX 76051.

The National Council of Teachers of Mathematics (NCTM) announces a Logo design contest in conjunction with Mathematics Education Week activities for April 14 - 20, 1985. For more information, write NCTM, 1906 Association Drive, Reston, VA 22091, or call (703) 620-9840.

Listful Thinking

by
Glen Bull and Paula Cochran

Power Tools

Did you ever have four or five PRINT statements in a row? Would you like a way to print five lines of text without putting PRINT in front of each line? You'll learn how in this month's column, if you use the WRITE procedure. It's great for creating adventure stories, for example. This month we're also going to look at two other list tools, SELECT and CREATE.

THE TEXT COMMAND
(Or "Wow! What Is All that Stuff in There?")

For nearly a year we haven't used a graphics command in this column. That isn't because lists aren't useful in graphics applications. On the contrary, they are very useful for graphics. (Tom Lough notes that a list of turtle commands can correspond directly to the addition of a series of vectors in physics.) However, we're trying to help dispell the myth that Logo is only a graphics language. Now we're going to ask you to type in a graphics procedure, but only so that we can look at its text. First define this procedure.

```
TO HAT
SETH 90
FORWARD 50 BACK 40
LEFT 90
REPEAT 4 [FORWARD 30 RIGHT 90]
END
```

Then, in the immediate mode, look at the text of the HAT procedure, using the TEXT command.

```
?PRINT TEXT "HAT
[] [SETH 90] [FORWARD 50 BACK 40] [LEFT
90 [REPEAT 4 [FORWARD 30 RIGHT 90]
```

When we look at the text of the procedure, it seems to be a list of commands. But what is the empty list [] at the beginning? To find out, let's make a new procedure that has an input. This new procedure, called CAP, lets us make a hat of any size.

```
TO CAP :SIZE
SETH 90
FORWARD :SIZE BACK :SIZE * ( 4 / 5)
LEFT 90
REPEAT 4 [FD :SIZE * (3 / 5) RT 90]
END
```

```
?PRINT TEXT "CAP
[:SIZE] [SETH 90] [FORWARD :SIZE BACK :S
IZE * ( 4 / 5 )] [LEFT 90] [REPEAT 4 [FD
:SIZE * ( 3 / 5 ) RT 90]]
```

The mystery of the empty brackets at the beginning is solved. That's where the inputs (such as :SIZE) are listed if there are any. If there are no inputs to the procedure, empty brackets appear at the beginning.

THE WRITE COMMAND

So far we appear to have found a bizarre way to look at the text of a procedure. But what if we wrote a procedure that contained English sentences instead of Logo commands? As you type this in, be sure to press the RETURN key at the end of each line shown below.

```
TO MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
END
```

After this MYSTERY procedure is defined, we can print the text in it using the TEXT command. However, what a mess!

```
?PRINT TEXT "MYSTERY
[] [SUNLIGHT DAPPLED THE FOREST FLOOR.]
[JOHNATHAN SLOWLY AWOKE, AND STRETCHED.]
 [HE COULD FEEL THE ROUGH BARK OF AN OAK
] [AGAINST HIS BACK. THROUGH THE LEAVES]
[HE GLIMPSED A PEAKED CAP ATOP A SMALL]
[FIGURE.]
```

We can get rid of the empty brackets at the beginning by printing everything but the first list in the MYSTERY procedure.

```
?PRINT BUTFIRST TEXT "MYSTERY
[SUNLIGHT DAPPLED THE FOREST FLOOR.] [JO
HNATHAN SLOWLY AWOKE, AND STRETCHED.] [H
E COULD FEEL THE ROUGH BARK OF AN OAK] [
AGAINST HIS BACK. THROUGH THE LEAVES] [H
E GLIMPSED A PEAKED CAP ATOP A SMALL] [F
IGURE.]
```

Listful Thinking continued

It still looks like a bit of a mess. However, there is a procedure from an earlier Listful Thinking column which can come to the rescue. The list tool INSPECT was developed to print out lists in neat column. You will find the INSPECT procedure at the end of this column. If you don't already have INSPECT in your toolbox, type it in now. Here's how to use INSPECT.

```
?INSPECT BUTFIRST TEXT "MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
```

This is the basis for a procedure to WRITE the text.

```
TO WRITE :PROCEDURE
INSPECT BUTFIRST TEXT :PROCEDURE
END
```

Here's how WRITE is used.

```
?WRITE "MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
```

Compare the text in a procedure designed for use with WRITE and a procedure consisting of a series of PRINT statements.

```
TO MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
END
```

```
TO MYSTERY.II
PRINT [SUNLIGHT DAPPLED THE FOREST FLO!
OR.]
PRINT [JOHNATHAN SLOWLY AWOKE, AND STR!
ETCHED.]
PRINT [HE COULD FEEL THE ROUGH BARK OF!
 AN OAK]
PRINT [AGAINST HIS BACK. THROUGH THE L!
EAVES]
PRINT [HE GLIMPSED A PEAKED CAP ATOP A!
 SMALL]
PRINT [FIGURE.]
END
```

When you use WRITE you save time (by not typing PRINT over and over). It's also easier to see where the end of each line will fall.

SELECTING STUDENTS

In a previous column, we developed a list tool called PICK for selecting items randomly from a list. For example, you could use it to pick students.

```
?MAKE "CLASS [TOM SUE JEFF SAM SALLY]
?PRINT :CLASS
TOM SUE JEFF SAM SALLY
?PRINT PICK :CLASS
JEFF
```

The list is not changed when a student is picked.

```
?PRINT :CLASS
TOM SUE JEFF SAM SALLY
```

Over the long term, PICK should choose each student about the same number of times. However, in the short term some students may be chosen more often than others. What we want is a means of selection without replacement. In other words, if we put all the names in a hat and pulled one out, we would not put it back in the hat again.

The DELETE procedure from last month can be combined with PICK to make a procedure which SELECTs without replacement. Listings for PICK and DELETE appear at the end of the column. Type them in now if you don't already have them.

```
TO SELECT :LIST
MAKE "ITEM PICK THING :LIST
DELETE :ITEM :LIST
OUTPUT :ITEM
END
```

Here's how to use SELECT. Note that, as input to SELECT, we use the name of the list as input (with ") instead of the value of the list (with :), so that the globally assigned values are deleted. (Ed. Note: See also Jim McCauley's related remarks in the Q & A column in this issue.)

```
?MAKE "CLASS [TOM SUE JEFF SAM SALLY]
?PRINT :CLASS
TOM SUE JEFF SAM SALLY
?PRINT SELECT "CLASS
SAM
?PRINT :CLASS
TOM SUE JEFF SALLY
```

The way in which SELECT differs from PICK is that SELECT randomly chooses an item from a list and deletes it from the list.

Listful Thinking continued

SELECT gradually destroys the list as it randomly selects and deletes. We need a way to recreate the list without typing it in again. Let's start by making a procedure that contains a list of class names. This may seem odd, but by now we hope we've built up some credit.

```
TO CLASS.ROLL
TOM KELLER
SUE MORGAN
KYSER LOUGH
SAM JOHNSON
SALLY ALLYSON
END
```

Use the TEXT command to look at the students in the procedure.

```
?PRINT TEXT "CLASS.ROLL
[] [TOM KELLER] [SUE MORGAN] [KYSER LOUG
H] [SAM JOHNSON] [SALLY ALLYSON]
```

Now let's make a list from the text of the CLASS.ROLL procedure. Here's how to create a list from a such a procedure.

```
TO CREATE :LIST.NAME :PROCEDURE
MAKE :LIST.NAME BF TEXT :PROCEDURE
END
```

Using CREATE, we can make a list out of the CLASS.ROLL.

```
?CREATE "CLASS "CLASS.ROLL
?PRINT :CLASS
[TOM KELLER] [SUE MORGAN] [KYSER LOUGH]
[SAM JOHNSON] [SALLY ALLYSON]
```

Now we can CREATE a new list any time we want. It doesn't matter if SELECT destroys the list of students.

```
?PRINT SELECT "CLASS
SUE MORGAN
?PRINT :CLASS
[TOM KELLER] [KYSER LOUGH] [SAM JOHNSON]
[SALLY ALLYSON]
```

POWER TOOLS

This month we have presented three new tools: WRITE, SELECT, and CREATE. As you experiment with them, we hope that you will discover how powerful these tools can be (yes, power tools!). Although they won't chop down any cherry trees, they can be combined to build exciting projects for your students. Next month, we'll show you how to use your new power tools to build a STORY WRITER.

PROCEDURES

Here are Apple Logo and IBM Logo procedures.

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

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

TO DELETE :ITEM :LIST
MAKE :LIST DELETE.LOOP :ITEM THING :LIST []
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST EMPTYP :OLDLIST
IFTRUE [PRINT SE :ITEM [NOT FOUND]]
IFTRUE [OUTPUT :NEWLIST]
TEST :ITEM = FIRST :OLDLIST
IFTRUE [OP SE :NEWLIST BF :OLDLIST]
OP DELETE.LOOP :ITEM BF :OLDLIST LPUT
       FIRST :OLDLIST :NEWLIST
END
```

Here are Terrapin Logo procedures for the Apple and Commodore.

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

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

(In Terrapin Version 1, PICK also requires these procedures: ITEM, ITEM.LOOP, COUNT, and COUNT.LOOP.)

```
TO ITEM :NUMBER :LIST
OUTPUT ITEM.LOOP :NUMBER :LIST 1
END

TO ITEM.LOOP :NUMBER :LIST :COUNT
IF :LIST = [] OP (SE [THERE ARE ONLY] :COUNT
        [ITEMS IN THIS LIST.] )
IF :NUMBER = :COUNT OP FIRST :LIST
OP ITEM.LOOP :NUMBER BF :LIST :COUNT + 1
END

TO COUNT :LIST
OUTPUT COUNT.LOOP :LIST 0
END
```

Listful Thinking continued

```
TO COUNT.LOOP :LIST :NUMBER
IF :LIST = [] OUTPUT :NUMBER
OP COUNT.LOOP BF :LIST :NUMBER + 1
END

TO DELETE :ITEM :LIST
MAKE :LIST DELETE.LOOP :ITEM THING :LIST []
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST :OLDLIST = []
IFTRUE PRINT SE :ITEM [NOT FOUND]
IFTRUE OUTPUT :NEWLIST
TEST :ITEM = FIRST :OLDLIST
IFTRUE OP SE :NEWLIST BF :OLDLIST
OP DELETE.LOOP :ITEM BF :OLDLIST LPUT FIRST
        :OLDLIST :NEWLIST
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.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

A Logo learning experience was used to explore experimentally the effects of programming on the conditional reasoning ability of fifth grade children. Conditional reasoning statements incorporate the "if - then" connective and commonly occur in our daily communication. Conditional branch statements appear in most high level computer languages available today. The statements (instructions) change the natural way in which computer programs are usually executed. The changes which occur depend on certain conditions being satisfied.

THE STUDY

Fifth graders, in a randomly assigned experimental group, learned to use the conditional branch statement in Logo in individual projects they worked on during a 15 week treatment period. About 2 hours per week were spent with teachers who used a guided problem centered approach to instruction. The need for conditional branch statements grew out of programming projects which involved game playing, question asking / answering procedures, and recursive procedures. A total of 42 children participated as randomly assigned experimental or control subjects. Control group children were not involved with Logo.

RESULTS

A traditional scoring of tests given at the end of the study showed no significant difference in scores of the two groups. An additional scoring procedure, however, indicated significantly better performance of the experimental group on one of the conditional reasoning principles. One other finding from the study of interest for further investigations: reading achievement scores for the control group showed a significant increase during this period!

>>>Seidman, R. H. (1981). The Effects of Learning the Logo Computer Programming Language on Conditional Reasoning in School Children (Doctoral dissertation, Syracuse University, 1980). Dissertation Abstracts International, 41, 2249B.

(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 Dissertation 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 is a doctoral candidate at the University of Virginia.

Graphics Dump From Logo

The UniPrint Parallel Printer Interface for the Apple II+ and IIe computers is compatible with Apple Logo, and permits a graphics dump from Logo to more than 25 different popular parallel printers. The modest price of $89 includes the interface card, Centronics cable, and instructions. Write to Videx, 1105 NE Circle Boulevard, Corvallis, OR 97330, or call (503) 758-0521.

Teacher to Teacher

Logo, Words, and Ideas Teacher Training Materials

reviewed by
James Fry

Logo, Words, and Ideas, by Thomas Boe, is a set of teacher training materials and lesson plans from the Minnesota Educational Computing Corporation (MECC) for a five session Logo course for Apple Logo teachers working with students in grades 5 - 12, computer coordinators and resource persons, and teachers of language arts, science, math, and computer science who are interested in using Apple Logo in their curriculum areas.

The introductory pages give a clear explanation of the goals, prerequisite skills, class format and setting (an Apple lab is recommended, with no more than two participants per computer), organization of the materials, and the preparation of participant diskettes. I welcome this information at the beginning of the manual.

The goals of the training activities are designed to help teachers:

-challenge students who are ready to go beyond turtle graphics,

-integrate Logo into language arts, science, and math,

-encourage both top down design and bottom up exploration.

The materials are designed for five sessions of at least two hours each. It is also suggested that the class sessions be spaced at least a week apart, to allow the participants time to experiment on their own.

ON THE INSIDE

The manual has five sections. The first section contains instructor notes, including helpful scripts for Logo lessons, and suggestions pertaining to materials, time, objectives, preparation, hands-on activities, and follow-up activities. This section is useful for both experienced and inexperienced Logo instructors.

The second section is a set of forty transparency masters of procedures and command explanations. The large print produces clear easy-to-read masters which give commands and their meanings and explanations, suggested activities, sample procedures, and drawings of what the procedures produce on the computer screen.

A set of participant activity pages makes up the third section, which contains explanations of each lesson and extended activities which coordinate well with the instructor notes.

The fourth section is devoted to participant notes; copies of transparencies for each lesson are provided, with space for personal notes.

The final section contains reference materials, including a summary of Apple Logo commands and word and list primitives. A bibliography and an index of word and list primitives round out this section.

THE FIVE SESSIONS

The five sessions focus on:

* recursion - investigating tail and embedded recursion and developing a theory to explain it

* introduction to Logo words - using word and list primitives

* singluar to plural - applying word and list commands to a group problem solving activity and identifying the steps in creating a computer solution to a problem

* hurkle project - participating in a Logo project activity which provides a variety of choices to explore

* mathematics and science - working with coordinate geometry and absolute angular directions and applying angular and coordinate commands to applications in mathematics and science.

IN SUMMARY

MECC's Logo, Words, and Ideas contains specific information, objectives, and support materials for both experienced and inexperienced Logo teachers. Although there is structure in the materials, creative exploration and discovery in the lessons is encouraged.

It is available for $49 from MECC, 3490 Lexington Avenue North, St. Paul, MN 55112, or call (612) 481-3527.

Teacher to Teacher continued

Logo and Mathematics Workbook Elementary Grade Materials

One interesting idea to come out of the Logo 84 conference is the Logo and Mathematics Workbook written by Reinhold Wappler of the New Canaan (CT) Country School. Wappler shared copies with colleagues at the conference, and showed that it was designed to be used as off-computer practice of what is learned in the school computer labs.

He described the book as having "emerged from three years of teaching mathematics via Logo in grades 2 through 5." And don't let the word "workbook" in the title mislead you; the book contains a reasonable balance of directed activities and suggested extensions, all related to specific mathematics concepts. Included are sections on lines, arcs, circle drawing commands, regular polygons, procedures, x- and y-coordinates, headings, and variables. In addition, class or group projects on string art, nested figures, and patterns of numbers and variables are suggested. The book is conveniently set up for use with any version of Logo.

SETHEADING :MATH

Those of us who have worked with younger students often find that, if left undirected, they learn very little from Logo and may lose interest quickly. The exercises in the book are all based on particular mathematical ideas appropriate for second through fifth graders. For example, Logo is easily adapted for the exploration of mathematical concepts such as external angles, coordinates, decimals, and number lines.

A helpful addition to this part of the book would be scripts for the teacher or suggestions for the computer activities which precede the appropriate workbook pages.

One of the powerful ideas in algebra, trigonometry, and calculus is that a change in the value of one variable produces a change in the value of another as a consequence. With this idea in mind, Wappler develops activities for turtle heading and position, and how these change as the turtle moves from one place to another.

Especially convenient teaching aids include a transparent turn indicator you line up with the turtle to find the number of degrees for a left or right turn, and a heading circle, used to estimate the current heading of the turtle.

IN SUMMARY

If you are a classroom teacher looking for some practical mathematics ideas and activities to use with Logo, give Wappler's Logo and Mathematics Workbook a try. Its 100 pages provide you with new ways of doing familiar activities.

Logo and Mathematics Workbook is available from the author. Send $13 to Reinhold Wappler, 252 Carter Street, New Canaan, CT 06840. For addresses outside the United States and Canada, enclose $5 extra for air mail postage.

Send your review reactions and comments to Logo Information For Teachers (LIFT), PO Box 5396, Plymouth, MI 48170.

James Fry teaches with Logo at Novi Woods Elementary School in Novi, MI, and is a cofounder of LIFT.

Q and A

by
Jim McCauley

Rather than address a specific question this month, let me deal with a subject of considerable confusion, both to a number of correspondents to this column and many of my students. The subject is the manner in which values are bound to variable names in Logo.

In the December Q & A, I introduced the two methods by which values are associated with names: by global assignment with MAKE and by binding to local variable names on entry to a procedure. I encouraged readers to try to use local bindings to the maximum extent possible, so that they might avoid the confusion presented by mixing local and global bindings. At the time, I didn't include an example of such a confusion; let me do it now. It serves to illustrate several points simultaneously.

Let's begin by creating a global assignment with MAKE:

```
MAKE "X "FRED
```

Now, let's define a simple procedure that accepts no inputs. All it does is print the value bound to X.

```
TO CONFUSING
PRINT :X
END
```

Q and A continued

When we type CONFUSING, the result is

```
FRED
```

However, if we create a superprocedure that calls CONFUSING, we can create a truly interesting mess.

```
TO SUPERCONFUSING :X
PRINT :X
CONFUSING
END
```

Oh, my. How does this procedure behave? The value FRED is already bound globally to the variable X. What happens when we try to associate a new value to the local variable X in SUPERCONFUSING? Furthermore, how does CONFUSING behave when it is called from within SUPERCONFUSING? Does it use the globally assigned value of FRED or does it use the value bound to the variable X in SUPERCONFUSING?

MOVING RIGHT ALONG

Let's end this Logo soap opera. If we type

```
SUPERCONFUSING "STUFF
```

we get in return

```
STUFF
STUFF
```

Oho! So when we pass a value to the variable X in SUPERCONFUSING as we call upon it, the value passed as a parameter (STUFF) supersedes the value that is bound globally (FRED). Furthermore, when CONFUSING is called from within SUPERCONFUSING, it "inherits" the value of X bound on entry to SUPERCONFUSION. In other words, in the versions of Logo that are available on microcomputers today, values that are bound locally take precedence over values that are bound globally.

This has the potential for causing some difficulty. We can see that CONFUSING has two very different behaviors. If it is called from TOPLEVEL, it uses the globally bound value for X. But if it is called from within another procedure which has a different value bound to X, that is the value which is used instead. While this erratic behavior can be usefully exploited in some special circumstances, it has a tendency to trap the unwary.

HOW TO UNDERSTAND?

Perhaps the best way to understand what is going on is to think of it this way. Many Logo primitives and procedures are able to accept inputs and bind them locally to variable names. Why not simply think of Logo's TOPLEVEL as a continually running primitive that can accept values bound to variable names, and that the MAKE primitive is the means by which this is accomplished? (Actually, this isn't too far from the truth.)

You should be aware that other computer languages use very different strategies for establishing the relationships among the local and global variables in use. There may even be some versions of Logo developed that do not follow these binding conventions. As you begin to write more complicated programs, take special care of how you treat your variables.

(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

Stand in a Corner

While the cherry trees were blooming in Washington, DC, last December, normally sunny Southern California was slogging through unseasonable winter weather. With some schools even closed due to snow (!), my students and I were shut in during our regular recesses because of the rain.

INDOOR SPORTS

An indoor game that my students (5th and 6th graders) played and really enjoyed that day forms the basis of this month's challenge. It gives us adults an opportunity to experience an example of the integration of classroom experience into the Logo envi-

NLXual Challenges continued

ronment. It also gives us a change to explore one of Piaget's postulates of intelligence.

In a slim book called The Psychology of Intelligence (Littlefield Adams & Co., 1976), Jean Piaget postulates that all problem solving activities involve several distinct stages. He calls the first stage "anticipatory schemata," representing the initial plan of attack for a given problem. This is, according to Piaget, a necessary component of intelligent behavior.

THE GAME

The game is generally called something like "Four Corners," and is played in a room with four designated stations or areas. In our case, we used the corners of the room. Each station is designated by a number, 1 through 4.

A stationmaster is appointed. This persons counts reasonably slowly to ten with closed eyes while everyone else (including the teacher!) scrambles to get to any one of the stations. Then the stationmaster calls one of the station numbers. Whoever happens to be in that station is "out" and has to sit down.

The game resumes with another round, using the players who are still "in." The game ends when one player is left, who becomes the new stationmaster.

The game is a unique "possession" of my students. It was spontaneously introduced by a student,, the rules were modified by our group, and the game has since spread to other classes in the school. (I know it didn't really originate with us, but we still call it "ours.") A spontaneous Logo problem was created when someone suggested, "Hey! Let's write a Logo stationmaster game!"

THE CHALLENGE

So, this month's NLXual Challenge to you is to write a stationmaster program that displays the four stations, "counts" to ten, and then randomly chooses a station. This program should possess all of the (minimal) abilities of a human stationmaster.

Before you begin to code your solution in the version of Logo available to you, check to see whether Piaget is correct; have you already formulated a plan? Do you know what you would like the finished product to look like? You might change your preliminary schemata as you progress, but the existance of such a schemata is a function of your own mental functioning, and will, in all likelihood, be a unique spontaneous creation. Try to preserve the core of your original plan in your final program.

Send your working solution (or even a valiant attempt) along with a self addressed stamped envelope to:

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

After compiling the contributed solutions, we will select several of the most interesting or unusual and redistribute them to each contributor. I promise to include several solutions already written by some of my students, too.

(Ed. Note: If any NLX readers have favorite Logo problems or challenges to share, include them with your solutions.)

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) 1985 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

This month, I'd like to introduce you to one of the newest Logos to be released. It's called MightyLogo, and was developed in Japan by Epson for their 8 bit computer, the QC-10 (known as the QX-10 in the rest of the world).

How is MightyLogo different from other Logos? What are some of its strengths and weaknesses? Given that I have worked with it only a month and that I have not yet seen students use it extensively (the real test!), please regard the following comments as preliminary.

I always tell my students to find positive qualities when evaluating anything which represents the effort of others. There is no doubt that much effort has been put into MightyLogo. Epson has attempted to tie the software into the hardware capabilities. MightyLogo uses the multifont board (part of the Epson hardware). Everything in this version is written to the graphics screen; there is no text generator or text page. This means that, as with the MacIntosh, graphics and writing (text) can be mixed together easily. This isn't quite as good as it sounds, however, because the writing erases any graphics image it touches.

But the real reason MightyLogo uses graphics for all its writing is to allow 16 different fonts (writing styles). Needless to say, this provides for some fancy on-screen displays. And, not surprisingly, all of this can be printed out easily in attractive style on Epson's well-known printers, using either a SCREEN DUMP key or a special primitive which allows you to send control codes directly to your printer.

Perhaps the most important new feature is the ability of MightyLogo to communicate directly with a plotter, such as the Epson HI-80 printer/plotter. Using the PLON (plotter on) command, the movements of a screen turtle can be plotted directly on paper, giving some of the same benefits of watching a robot turtle drawing lines on the floor. Since the HI-80 allows for 16 different colors (4 cartridges can be used at once), plotting in color provides students with beautiful multicolored graphics designs to take home.

Some other convenient features of MightyLogo include simple ways to rename primitives and set up self starting files, and the ability to save directly from the editor to the disk. The graphics are drawn much faster than most popular Logo versions, but the writing and editing is rather slow.

This brings me to the real point of this month's column. We are seeing so many new versions of Logo with exciting features added to make each more attractive than other Logo versions. Recently, I've been reading the publications and Logo manual of The Lisp Company (John Allen's TLC) version of Logo. They have been working with LISP for twenty years and have a real commitment to Logo as a tool for thinking. Their own version of Logo (I haven't yet seen it) sounds as if it was made with a lot in mind besides being more "jazzy" that the others. TLC's goal seems to be helping users develop their own thinking and creative powers.

When I see new versions of Logo with FILL commands, multifonts, 2400 different color combinations, and supersonic turtles, I think we need to remind ourselves not to be overattracted to these frills. Instead, we need to insist that the Logos we purchase for our education and that of others be designed with a firm commitment to that education.

MightyLogo has a lot going for it, especially the ease of communication with printing devices. How we could use these capabilities to extend our learning and our understanding of the learning processes is the challenge. As for Epson and other companies who produce new versions of Logo, let us hope that their contact with the Logo educators stays strong; don't let the philosophy behind Logo get lost in the competition to have more and more attractive versions.

If you want more information about Logo in Japan, write to me at Doshisha International High School, Tatara, Tanabe-cho, Tsuzuki-gun, Kyoto-fu, Japan 610-03.

MAE 100!

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).

The Los Angeles County Office of Education's Teacher Education and Computer Center is hosting a Logo conference for teachers K - 12 at the Universal Sheraton Hotel in Universal City on Saturday, April 13, from 8 a.m. to 5 p.m. For more information, call Darrell Smedley at (213) 922-6680.

Logo 85, the Second International Logo Conference, is scheduled for July 22 - 25 at the Massachusetts Institute of Technology in Cambridge, MA. Themes announced for the conference include evaluation, teacher education, and the integration of Logo into the curriculum. For more information, write to the MIT Special Events Office, Room 7-111, Cambridge, MA 02139.

WCCE/85, the World Conference on Computers in Education, will be held in Norfolk, VA, July 29 to August 2. Included in the program are pre-conference Logo workshops and a full day of Logo panel discussions on various international and national issues. A Logo Birds-of-a-Feather session will give an opportunity for informal sharing. For more conference information, write to Richard L. Dobson, Jr., AFIPS, 1899 Preston White Drive, Reston, VA 22091, or call (703) 620-8935.

The West Coast Logo Conference is scheduled for November 21 - 23, 1985, at the Pacifica Hotel in Los Angeles, CA. Papers on research relating to Logo are being invited. The conference is co-sponsored by Pepperdine University, Computer Using Educators (CUE), the International Council for Computers in Education (ICCE), and the Los Angeles Teacher Education Computer Center (TECC). For more information, write to West Coast Logo Conference, Pepperdine University, 3415 Sepulveda Boulevard, Los Angeles, CA 90034, or call (213) 306-5600.

The Northwest Council for Computer Education hosts their 14th annual conference February 7 - 9 in Eugene, OR. "Computers in Education - What's Next?" includes over 90 sessions focusing on current educational computing issues and applications. For more information, call Maggie Moore at (503) 754-4031.

The World Congress on Education and Technology is scheduled for May 22 - 25 in Vancouver, British Columbia, Canada. For more information, call Marie-Catherine Berger at (604) 734-2721.

Logo Hawaiian this summer! Join Glen Bull, Paula Cochran, and Tom Lough for "Logo in the Classroom," a 3-hour graduate Logo course from the University of Virginia, to be taught at the famed Punahou School in Honolulu, Hawaii, July 8-19, 1985. The course concentrates on Logo applications in language arts, science, and mathematics content areas for teachers with Logo experience. For Logo beginners, an outstanding graduate introductory course is offered, taught by Elaine Blitman and Barbara Jamile. Make your plans early for these popular offerings. For more information, call Educational Spectrum, Inc., at (800) 223-2544. New York residents call (212) 686-5810 collect.

Educational Spectrum also announces "Microcomputers in the Elementary Instructional Program," and "Microcomputers in the Secondary School Curriculum," 3-hour graduate courses offered through the California State University at Los Angeles, to be taught July 19 through August 3, 1985. Included will be instruction on Logo and other software, such as word processors and data bases. For more information, call Educational Spectrum at the above numbers.

Logo Computer Systems Inc. (LCSI) and Educational Spectrum announce "A Workshop for Experienced Logo People," to be held August 5 - 9 in New York City. Advanced Logo programming and applications in mathematics, science, language, and social studies will be guided by a distinguished faculty of experienced Logo instructors and lecturers. The course fee is $495. For more information, call Michael Tempel at (212) 765-4780.

The Central Pennsylvania Special Education Resource Center is sponsoring a free workshop, "Logo: Beyond Turtle Graphics," in Harrisburg, PA, on Saturday, May 4, with a focus on applications and teaching strategies for Logo's list processing capabilities. For more information, call Gay Holston at (717) 657-5840.

James Hurley's Logo Physics is now available. The 228 page paperback book is designed for use with high school and college general physics courses. For more information, write Holt, Rinehart and Winston, 383 Madison Avenue, New York, NY 10017.