Daniel H. Watt, Ph.D.
Educational Alternatives
Gregg Lake Road
Antrim, New Hampshire 03440

Teacher-Made Microworlds: or Training Teachers to Use Logo Vs. Training them to Teach It

In this paper I want to share my experience using Logo as an environment for helping teachers create microworlds that can be used in a broad range of educational settings. I believe that training teachers to make microworlds can make a difference in two areas of great concern to Logo-using educators: effectively integrating Logo and the Logo philosophy into the traditional school curriculum, and training teachers to use Logo with vision as well as expertise.

Logo was intended to be both a computing language and a learning environment. Although it was designed to have "no threshold and no ceiling" (Papert, 1980), that is, to be suitable for advanced programmers as well as beginners, Logo's use in educational settings has been limited mainly to children in the elementary and middle school grades. In many instances, Logo is taught as a programming language, stressing the mastery of different commands and programming techniques, at the expense of the mathematical and problem solving ideas embedded in the language (Watt and Watt, 1985). In addition, most Logo use has been limited to relatively simple turtle graphics procedures, ignoring Logo's potential for interactive programming and information processing.

Logo classrooms could look very different a few years from now. Students will still learn Logo as a programming language; but they will also use Logo and Logo-based activities to explore other content areas. And in addition to creating their own programs from scratch, students and teachers will use Logo procedures created by others.

There are many possibilities for using Logo microworlds in content areas such as art, music, science, mathematics, language arts and problem solving. Logo microworlds can even be used to help students learn aspects of Logo itself. Teachers in such classrooms will have to be knowledgable Logo users themselves, with the ability to support students in learning any programming techniques they need to achieve their goals, and knowledge of many ways to help students make connections between their Logo experiences and powerful ideas in each of the areas mentioned.

It will take time to develop a cadre of Logo teachers with these kinds of expertise and to disseminate appropriate teaching materials. It will also take vision and planning on the part of those who are doing teacher training and preparing Logo materials for commercial publication.

During the past few years Molly Watt and I have taught several

graduate-level Logo courses and workshops which have included the design and development of Logo-based microworlds. The participants have ranged from complete novices to teachers who already know some Logo programming and appreciate Papert's educational philosophy. The result has been a group of teachers with a deeper appreciation of what Logo is all about -- teachers who have begun to shift focus from teaching Logo to their students, to using Logo to create new types of learning environments in their classrooms. And in the process of learning to use Logo themselves, they may have become better teachers of Logo as well.

What is a Microworld?

The concept of a microworld is not very well-defined, although the term is beginning to be widely used. In Mindstorms, Seymour Papert described microworlds as "incubators of knowledge" -- environments in which children can acquire powerful ideas by carrying out activities within them. The best known example is the Logo microworld of the turtle, in which important mathematical ideas can be learned through exploration (Papert, 1980).

Robert Lawler, who has developed a number of microworld environments along the lines suggested by Papert, has offered a useful set of criteria for the design of microworlds. In Lawler's view, a microworld should be based on "'neat phenomena' -- phenomena that are inherently interesting to observe and interact with." Furthermore, it must include "powerful ideas." According to Lawler such ideas are simple enough that they can be easily understood by the learner; they are general, having application outside of the microworld itself -- otherwise why bother creating it; they are immediately useful to the learner -- otherwise why bother learning them; and they are syntonic, closely related to the learner's existing mental models (Lawler, 1982b).

In addition a microworld designer must take into account the interests and knowledge of the particular learner or learners he is designing for. Lawler, for example, has designed most of his microworlds primarily for his own children, drawing on their particular interests, environmental setting, and learning needs for inspiration (Lawler, 1982a, 1982b).

Microworlds, whether on or off a computer, can take many forms. They can be games, simulations, programming languages, tools for creative expression, or environments (like sandboxes) that encourage playful exploration. One distinction that is important for me, however, is what microworlds are not: microworlds are not computer-based lessons, tutorials, or games designed to lead learners to particular sets of facts, skills or conclusions in a structured sequential manner.

In the rest of this paper I will describe some examples of microworlds made by teachers who have worked with me and with Molly Watt, as well some that we have made ourselves. The

examples include several different kinds of applications, and have been constructed using different levels of Logo programming skills. Some of them have been described previously in the Proceedings of the 1985 ECOO/AEDS Conference (Watt, 1985).

Word-Action Microworlds

Word-action microworlds are among my favorites. The idea was first articulated by Seymour Papert and developed by Lawler (Lawler, 1982a,b). They are designed for use by very young children, having their first encounters with written words.

The idea is to create an environment that parallels the way children learn to speak by issuing commands. The word "Mama" or "Daddy" brings a parent from the other side of the room. "Up" means "pick me up and carry me." Saying the words "milk" or "juice" will often result in a drink. And so on.

In a word-action microworld typing a word on the keyboard produces an action on the TV screen. Current examples are based on the use of sprites, because they can take on a variety of shapes and move freely across a computer screen. One example, the Farm microworld, was created in the summer of 1984 in Vancouver, British Columbia, by David Bell, Leon Lebrun, Larry Wiebe and Elaine Willis for Bell's four-year-old son, who lived in a farming community and was fascinated by farm animals and equipment. If the child types a name such as PIG, COW, TRACTOR or SUN, an object with the appropriate image appears on the screen. Action words such as WALK, DRIVE, FLY, UP, DOWN, LEFT or RIGHT, set an object in motion or change its motion. WHITE, YELLOW, RED, GREEN, or BLUE change its color. "Reality" is not a limitation. In this world pigs can fly and trees can walk -- but only if the child wants them to.

The objects and vocabulary of a word-action world must be tailored to the individual user. Gerri Sinclair, also in Vancouver, built a microworld full of city vehicles -- cars, trucks, planes and trains -- for her two-year-old son. Bob Lawler used a BEACH setting for his children, who lived near the seashore. The important thing is that the same Logo procedures can be used for any number of worlds, just by changing the names and the shapes. So although the size and complexity of current examples are limited by memory limitations, it's easy to make lots of different ones, that can be used one at a time.

The Logo Poet

The Logo poet is one of many microworlds oriented around generating language patterns using parts of speech. The version I often use as a starting point generates "poems" that sound something like Haiku (Watt, 1983). Other possibilities involve random sentences, conversational programs (Abelson, 1982), or language games such as Madlibs (Watt and Watt, 1985). Here is a sample poem (Watt, 1983):

```
EACH LIMPID POND
ONE BIRD RACES OVER
  THE FOSTY FIR
WILD BLUE MOON
```

Each new poem follows the same pattern of nouns, verbs, adjectives, and so on, but with different words, chosen randomly from long lists of words. To change the results, alter the pattern or substitute your own word list. In this environment, you can explore the structure of language, the cultural values of different people, and the usefulness of parts of speech when used to construct human expression, rather than simply to analyse it.

Programs like this can be used for fairly sophisticated language explorations. A college class has used the POET program as a starting point for a close analysis of traditional Haiku forms (Hoffer and Semmes, 1984). Others have made the selection process more complex so that the words chosen for a given poem are related to each other in predetermined ways (Sharples, 1980). Gerri Sinclair developed a version which generates an unending sequence of "neurotic" statements, that can only be stopped by pressing the interrupt key. The possibilities have only begun to be tapped.

Learning Logo Itself

A number of microworlds have been developed to help beginners learn Logo itself. Rather than have to deal with all of Logo's complexity, such environments simplify Logo, making it a less of a challenge, and providing more focus. Properly designed, such microworlds do not replace the full Logo environment, but lead a learner into it more gently, or help a learner master some complicated aspect of turtle graphics. And I have learned from working with Molly Watt that these environments can be created using the simplest Logo programming techniques.

During the past year, Molly Watt and I have taught Logo to a group of 3-to 8-year-olds at the Price Farm School, an alternative school in Antrim, New Hampshire. For these students we created an environment that simplified certain aspects of Logo, without altering the fundamental structure of the language. We made a few new turtle "primitives" like these:

```
TO F :DISTANCE
REPEAT :DISTANCE [FORWARD 10]
END

TO R :TURNS
REPEAT :TURNS [RIGHT 10]
END
```

And so forth.

The microworld includes a few other simplified commands for saving and printing pictures. Otherwise, the children have access

to all the conventional Logo commands as needed. They still have to type inputs to move the turtle, but inputs from 1 to 9 have clearly observable results -- ten times the effect of normal inputs to FORWARD and RIGHT.

The REPEAT command slows the turtle down, and makes it appear to move and turn in response to F, B, R and L commands. And when the students use large input numbers, the turtle takes a long time to move and turn, which makes the effects of large rotations and patterns that wrap around the screen much more concrete than in conventional Logo implementations. For example, the command R 36, turns the turtle 360 degrees to the right, wheras the corresponding Logo command, RIGHT 360, produces no effect at all.

The Price Farm microworld also has the advantage that the transition to "real Logo" is an easy one -- just substitute FORWARD 100 for F 10, and RIGHT 90 for R 9. And it is extensible in the normal Logo way -- students add their own procedures to the microworld as they go along.

Other microworlds provide simple challenges that help learners build confidence in estimating angles and distances. The target game, SHOOT (Watt, 1983, derived from an idea by Lawlor), has led to a number of interesting variations. Elaine Willis, of Vancouver BC, made a "socially positive" modification, in which the goal of the game is to toss a piece of litter into a trash barrel. (A successful toss results in the message: "Thanks for keeping the workspace tidy.") Cathy Frank, of Burlington Vermont, has separated the elements of the target game into two games, People-Pop, which focuses on estimating distances between squares on the screen, and Catch-Us, which focuses on estimating rotations.

Another microworld that has spawned many variations is the House microworld, developed by Barbara Husbands of Peterborough, New Hampshire, working with Molly Watt. In this microworld, the teacher provides procedures to draw a house outline, a door, windows, trees and flowers. The students have to learn their way around the screen and account for the state of the turtle (position, heading and penstate) in order to place these drawings where they want them on the screen. Many other settings come to mind (decorating a Christmas Tree is one variation I've seen), and students can add their own procedures to the mix as well.

Other teachers in our classes have developed microworlds that let students explore the mathematics of turtle geometry, highlighting such areas as polygons and stars, symmetry, rotation, pattern making, tessellations, and recursive fractal designs.

Microworlds and the Future of Logo

I believe that microworlds have an important place in the future of education in general, and of Logo in particular. They provide educators with ways to informally extend the richness of a classroom environment while incorporating computers into their

ongoing curriculum. Although the process of microworld development has barely begun, I can foresee a time -- perhaps 3 to 5 years from now -- when there will be enough educators with knowledge of Logo, and a culture rich enough to support sharing of ideas and programs, so that microworlds can become an important part of the learning process.

The microworlds I've mentioned in this paper barely scratch the surface of what's possible. There are many other microworlds ripe for development. Some of them may require a bit more programming expertise than the ones I've already mentioned. For instance, many of the activities developed during the first decade of the Logo project at MIT have not yet been widely used. These include dynaturtle games (Watt, 1981, diSessa and White, 1982), Logo music, (Bamberger, 1974, 1982), and using Logo procedures to model animal behavior (Abelson and Goldenberg, 1977).

The most important benefit of training teachers to create their own microworlds may be that it focusses attention on Logo as an environment for learning rather than as a subject to be taught. At the same time it provides an appropriate domain for the development of their own Logo skills, making them better teachers of Logo in the long run. And finally, it helps to bring into being a collection of useable microworlds that can be shared with other teachers.

This leaves us with the challenge of finding ways to disseminate teacher-made microworlds. Such products should not be sold like commercial software. That would make them too expensive, and could defeat their purpose as extensions of Logo. On the other hand, the simple swapping of program disks is also inadequate.

A teacher-made microworld should be considered a professional product. Its procedures should be well-structured, so that they can be easily used and modified. And it should be well documented, so that both teachers and students can understand its purposes and use it effectively.

Perhaps we should start a kind of "microworld journal," with an editorial board to evaluate both the programming style and the educational usefulness of a particular product, edit the program and documentation if necessary, and disseminate disks for many different versions of Logo. Such a project could begin on a small scale, but in the long run, it could have immense benefits for the continued growth of the Logo community and the development of a flourishing Logo culture.

References

Abelson, Harold, Logo for the Apple II, Byte Books / McGraw-Hill, 1982.

Abelson, Harold and E. Paul Goldenberg, "Teacher's Guide for Computational Models of Animal Behavior," MIT Logo Memo #46, MIT

Logo Group, 1977.

Bamberger, Jeanne, "What's in a Tune?" MIT Logo Memo #13 MIT Logo Group, 1974.

"Logo Music," Byte, August 1982.

diSessa, Andrea A. and Barbara Y. White, "Learning Physics from a Dynaturtle," Byte, August 1982.

Hoffer, Bates L. and Pat Semmes, "Haiku and Nim: Logo in the Language Arts," Proceedings, National Educational Computing Conference, Dayton, Ohio, 1984.

Lawler, Robert W. a) "In the Lap of the Machine," Boston Review, June 1982.

b) "Designing Computer-based Microworlds," Byte, August, 1982.

Papert, Seymour, Mindstorms, Basic Books, 1980.

Sharples, Mike, "A Computer-based Language Workshop," Sigcue Bulletin, Vol. 14(3), 1980.

Watt, Daniel, "Games With a Dynamic Turtle," unpublished material, 1981.

Watt, Daniel, Learning With Logo, Byte Books/McGraw-Hill, 1983.

Watt, Daniel, "Creating Logo Microworlds," Proceedings, ECOO/AEDS, 1985 (Joint Conference of Educational Computing Organization of Ontario and the Association for Educational Data Systems), Toronto, Ontario, April 1985.

Watt, Molly and Daniel Watt, Teaching With Logo, Addison-Wesley, 1985.