

Designing Computer-Based Microworlds

*Well-designed Logo procedures
can help children grasp ideas of intrinsic interest.*

R. W. Lawler
Le Centre Mondial L'Informatique
et Ressources Humaine
24 Rue Clemente Marot
Paris, France 75008

Designing computer applications for education might be called cognitive engineering, for its objective is to shape children's minds. That lofty goal must carry with it a commitment to cognitive science, the study of how knowledge functions and changes in the mind. In light of the profound influence of computers in the schools, designing educational applications without such a commitment would be irresponsible.

I believe that Jean Piaget, the Swiss student of knowledge, formulated the general solution to the problem of how intelligence develops. Although the field of cognitive science has advanced beyond Piaget's innovative theories by revising and extending them, his insights into the nature of learning continue to influence teaching methods. The union of computer microworlds and Piagetian theory is the subject of this article.

Piaget and Education

Central to the work of Piaget is constructivism, the view that the mind incorporates a natural growth of knowledge and that the mind's structure and organization are shaped by interactions among the mind's parts. In *The Science of Education and the Psychology of the Child* (The

Viking Press, 1971), Piaget challenges educators to answer two questions: How does instruction affect what is in the mind? and What remains in the mind from the process of instruction long after the time of instruction has passed? In the same work, Piaget disputes both the effectiveness and the ethical correctness of many of the practices of modern education:

If we desire to form individuals capable of inventive thought and of helping the society of tomorrow to achieve progress, then it is clear that an education which is an active discovery of reality is superior to one that consists merely in providing the young with ready-made wills to will with and ready-made truths to know with.

The Dilemma of Instruction

Given Piaget's view that learning is a primary, natural function of the healthy mind, we might consider instruction in any narrow sense unnecessary. Children (and older students of life as well) learn the lessons of the world, effectively if not cheerfully, because reality is the medium through which important objectives are reached. Nevertheless, in certain situations children often rebel against

the lessons society says they must learn. Thus the educator's ideal of inspiring and nurturing the love of learning frequently is reduced to motivating indifferent or reluctant students to learn what full functioning in our society requires.

Teachers face a dilemma when they try to move children to do schoolwork that is not intrinsically interesting. Children must be induced to undertake the work either by promise of reward or threat of punishment, and in neither case do they focus on the material to be learned. In this sense the work is construed as a bad thing, an obstacle blocking the way to reward or a reason for punishment. Kurt Lewin explores this dilemma in "The Psychological Situations of Reward and Punishment" (*A Dynamic Theory of Personality: Selected Papers of Kurt Lewin*, McGraw-Hill, 1935). The ideas of Piaget and Lewin have led me to state the central problem of education thus: How can we instruct while respecting the self-constructive character of mind?

Computer-Based Microworlds

In *Mindstorms: Children, Computers, and Powerful Ideas* (Basic Books, 1980) Seymour Papert pro-

poses computer-based microworlds as a general solution to the problem of motivation. One argument for Papert's proposal runs as follows: learning is often a gradual process of familiarization, of stumbling into puzzlements, and resolving them by proposing and testing simple hypotheses in which new problems resemble others already understood. Microworlds are in essence "task domains" or "problem spaces" designed for virtual, streamlined experience. These worlds encompass objects and processes that we can get to know and understand. The appropriation of the knowledge embodied in those experiences is made possible because the microworld does not focus on "problems" to be done but on "neat phenomena"—phenomena that are inherently interesting to observe and interact with.

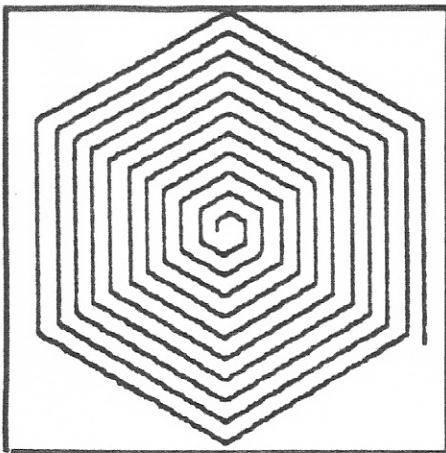
With neat phenomena, the challenge to the educator is to formulate so clear a presentation of their elements that even a child can grasp their essence. A well-designed computer microworld embodies the simplest model that an expert can imagine as an acceptable entry point to richer knowledge. If a microworld lacks neat phenomena, it provides no accessible power to justify the child's involvement. We can hardly expect children to learn from such experiences until they are personally engaged in other tasks that make the specific knowledge worthwhile as a tool for achieving some objective. This amounts to an appropriate shifting of accountability from students (who have always been criticized for not liking what they *must* learn) to teachers, those who believe that their values and ideas are worth perpetuating.

Computer-based microworlds help tailor instruction more closely to Piaget's idea of the natural mode of learning. I will illustrate this point by presenting two examples of computer-based microworlds.

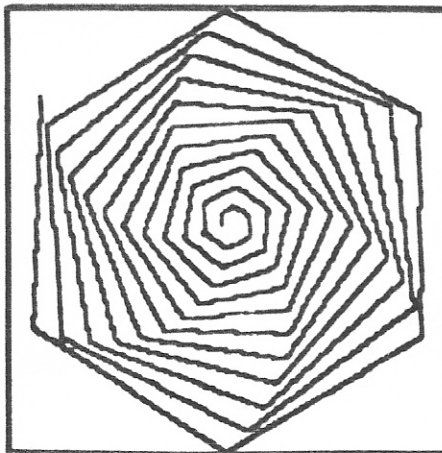
The POLYSPI Microworld

POLYSPI (from "polyspiral") is a name for a three-line procedure in the Logo language and for the class of de-

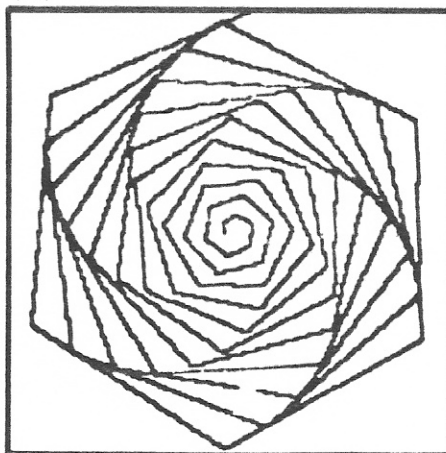
(1a) POLYSPI 10 60 3



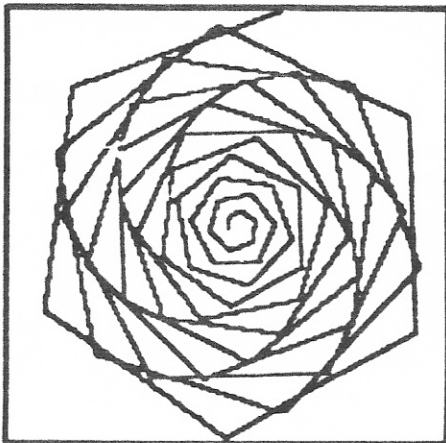
(1b) POLYSPI 10 61 3



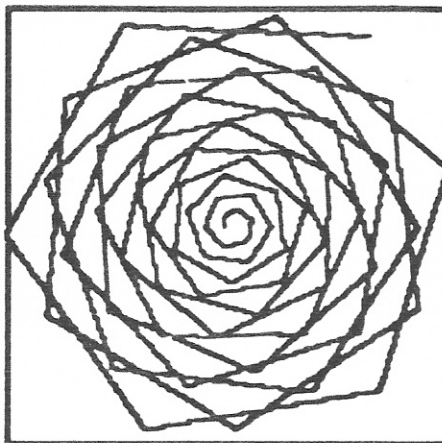
(1c) POLYSPI 10 62 3



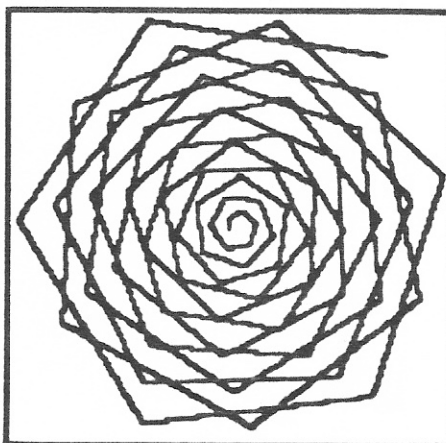
(1d) POLYSPI 10 63 3



(1e) POLYSPI 10 64 3



(1f) POLYSPI 10 65 3



Figures 1a-1f: Polypiral designs generated by changing one variable of the three in the POLYSPI procedure (shown in listing 1). The procedure's variables are DISTANCE, ANGLE, and CHANGE (in distance). The procedure draws a design by going forward the specified distance, turning at the specified angle, then increasing the distance by the specified change, going forward for the incremented distance at the specified angle, and so on. In this example, the distance variable and the change in distance are held constant. The angle variable is stepped up by one degree in each design. The strikingly different designs show the power of the concept of stepping variables.

signs produced by different executions of that procedure. Figures 1a-1f show some examples of POLYSPI designs. The POLYSPI procedure is stated in listing 1. Some of the designs are pretty, mainly because surprising spiral patterns emerge under certain conditions. The general appeal of POLYSPI designs largely accounts for the adoption of turtle graphics as a subsystem of languages such as Smalltalk, Pascal, and even some implementations of PILOT. The variability of the POLYSPI procedure sometimes permits even a beginner to surprise more expert users (as well as himself) with the discovery of beautiful designs.

The procedure in listing 1 and its designs comprise a microworld. The objects of the microworld are all the designs that the procedure can generate, an engaging and extensive domain for

exploration. More important, the designs are a class of "neat phenomena" whose generation can be made comprehensible with the following small set of ideas. First, the POLYSPI procedure provides a crisp model of variable separation: the three vari-

The POLYSPI microworld reveals the powerful idea of stepping variables.

ables DISTANCE, ANGLE, and CHANGE are each used once, and used differently, in a simple procedure text. Second, the difference in relative potency of the variables (the impact of a unit change on the produced design) is obvious and striking. (ANGLE and then CHANGE are much more potent than DISTANCE.)

The POLYSPI microworld reveals the stepping of variables as a powerful idea. By stepping variables I mean identifying one variable as a dimension of examination and holding all other variables constant while the chosen one is varied incrementally. In short, this microworld provides a clear model of how particular things may be generated through their intersecting dimensions of variation. Piaget judged variable-stepping to be an essential component of formal operational thought. The idea is a powerful one because it is almost universally useful; it is crucial to the process of scientific investigation.

Within the microworlds of turtle geometry, the insights achieved with POLYSPI exploration are easily extended to a related microworld of INSPI designs. The INSPI procedure differs from POLYSPI only in that the

Listing 1: The POLYSPI procedure, written in Logo. From only three variables—distance, angle, and change in distance—this procedure can generate a remarkable variety of polyspiral designs. The procedure draws by going forward the specified distance, turning at the specified angle, then increasing the distance by the specified change, going forward for the newly incremented distance at the same specified angle, and so on. Some designs drawn by POLYSPI appear in figures 1a-1f.

```
TO POLYSPI :DISTANCE :ANGLE :CHANGE
  FORWARD :DISTANCE
  RIGHT :ANGLE
  MAKE "DISTANCE :DISTANCE + :CHANGE
  POLYSPI :DISTANCE :ANGLE :CHANGE
END
```

change value is applied to the ANGLE variable instead of to the DISTANCE variable. (For a case study of a child's ability to grasp and extend this idea, see my article "Extending a Powerful Idea," in a forthcoming issue of *The Journal of Mathematical Behavior*.)

The BEACH Microworld

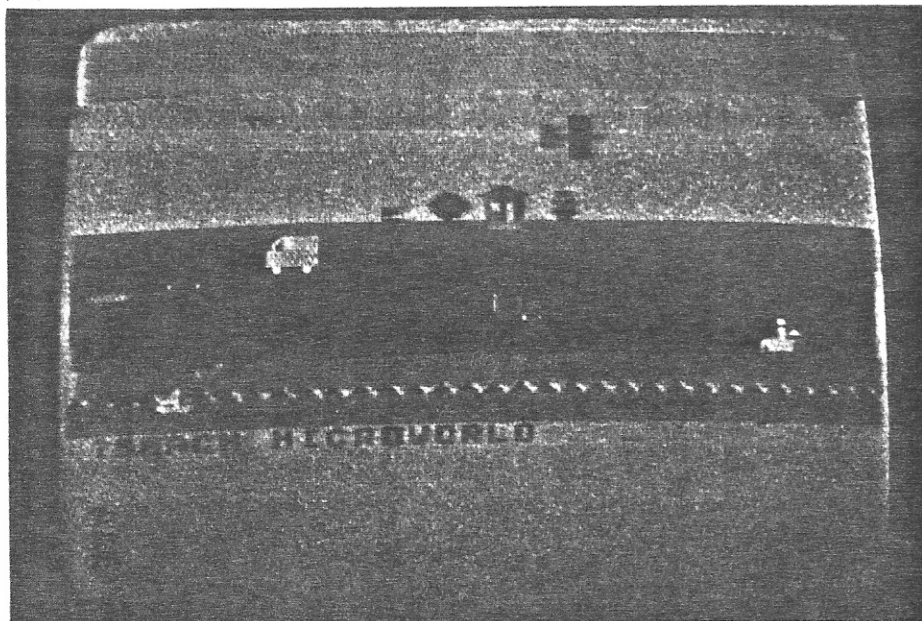
The adolescent's initiation to formal thought differs greatly from the preschooler's introduction to reading, yet both learning experiences involve

grasping central representations. What the prereader learns in an alphabetic language is a serial symbolic representation for words that signify the names of objects, actions, and so on. Let me here describe a Logo microworld for learning the alphabetic language. This microworld helped my 3-year-old daughter learn to read with minimal direct instruction.

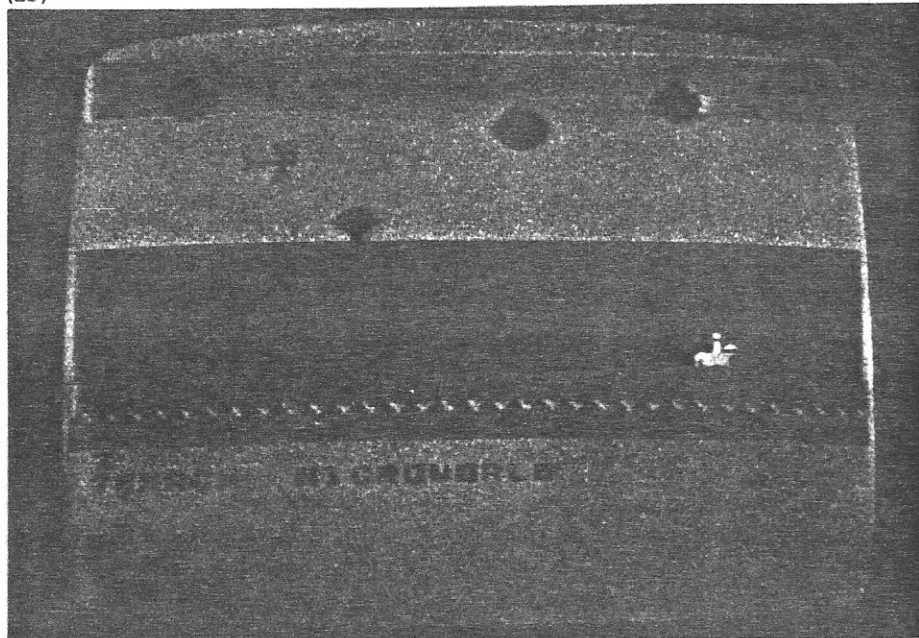
While previous Logo implementations focused on a single, all-im-

portant agent—the turtle—TI Logo also has sprites. A sprite is a video-display object that has a location, a heading, and a velocity, but no drawing capability. It may be associated with a shape (which it "carries" and which assumes one of 16 colors). The shapes can be easily defined and changed by the Logo user. There may be a maximum of 25 shapes. The importance of a multitude of easily discriminated objects for early language applications cannot be overestimated. TI Logo has a second graphics system, "tile graphics," that is compatible with the sprite graphics system. The static tiles, which may also assume 16 different colors and exhibit modifiable shapes, provide a suitable "background" for the movements of the dynamic sprites. The result is the opportunity to create scenarios that have many moving objects with different shapes and different colors and a static but vivid backdrop. The BEACH microworld permits the creation of such scenarios, as the scenes in photos 1a and 1b illustrate.

(1a)



(1b)



Photos 1a-1b: Two scenes from the BEACH microworld. Photo 1a shows a scene with many objects. Photo 1b shows a scene typical of those drawn by a 3-year-old child. The author's daughter learned to read 30 words by exploring the BEACH microworld, which the author and his children created using TI Logo with the Texas Instruments 99/4A microcomputer.

Meaningful Names Ease Learning

Because Logo gives the user great freedom to define and name procedures, appropriate descriptive English words can be used. For example, SUN can be the name of the procedure that creates a yellow ball on the display. The word UP can name the command that increases the value of a sprite's *y* coordinate. Repeated often enough, UP puts the SUN in the sky above the BEACH. Another word, such as SLOW

or FAST, can set the SUN in motion. Because new procedures are easily defined, the child, a family member, friend, or teacher could even make the SUN ZOOM if the child wishes. Such flexibility permits the microworld to be tailor-made to suit any child. To the extent that the child participates in defining the objects to be part of the world, their attributes, and the actions they are to perform, the microworld is also constructed by

the child herself. My 3-year-old, Peggy, her older siblings, and I chose about 20 objects to populate her world, designed and made shapes to represent them, and wrote the procedures to create and manipulate them. The vocabulary of her BEACH world includes the following:

OBJECTS

BEACH, BIRD, BOAT, BOY, CAR, DOG, FISH, GIRL, HOUSE, JET, KID, MAN, MOON, OAK, PINE, PLANE, PONY, STAR, SUN, TRUCK, VAN, WAGON

ACTIONS

UP, DOWN, MOVE, BACKWARD, FAST, FLY, HALT, SAIL, SHOW, SWIM, TURN, WALK, ZAP, ZOOM, PAINT BLACK, PAINT GREEN, etc.

When Peggy began to play with this computer microworld, she did not recognize any words except "by," and she had no idea what that meant. Her ability to discriminate between letters and name them was undependable and idiosyncratic. She began keying words, copying them letter by letter from a set of 4- by 6-inch cards I had made. Soon she began keying her favorite or most-used words from memory, and later she was able to read those individual words in other contexts. Now she deals with the written language one word at a time (as infants begin to speak with specific signification). To handle sentences (other than "paint some-color-name") or begin phonetic decoding of words, she will need more complex microworlds.

For Peggy, the learning of reading and the learning of writing have been synchronized (as speaking and interpreting speech are for the toddler); she learned to read her 30-word vocabulary by learning first to "write," i.e., key the words on the computer terminal. Writing was an essential part of controlling the computer microworld that engaged her. My role as teacher changed from taskmaster to occasional consultant. I would answer questions Peggy brought me after she had tried to work with the constructed reality of the BEACH microworld, and I helped her when she had problems, but I offered her no lessons beyond the rule that words are keyed letter by letter,

left to right, and that a specific symbol meant she should press the Enter key.

I make no claim that computer microworlds can teach all reading skills, nor that this specific BEACH world would appeal to other children in different circumstances. I do, however, see the BEACH microworld as a prototype of the various worlds that others may fashion for small children.

Design Heuristics: Powerful Ideas

A computer microworld should be constructed around a powerful idea, one worth the instructor's time to develop or the student's time to explore. Who decides if an idea is sufficiently powerful? You do, at first, when you design a microworld. Next, the students determine the worth of the microworld as they incorporate the idea into their minds or reject it.

If you need a little guidance when you design a computer-based microworld, Papert (in *Mindstorms*) offers four criteria for powerful ideas: they should be simple, general, useful, and syntonic. The idea behind a microworld must be formulated as simply as possible; an idea can be powerful only when understood. Even if an idea is embodied in a specific microworld, it will not be useful through extension unless it is general.

Reality dictates the candidates for powerful ideas. Society also declares what ideas are important: if you can't read, for example, a technological society relegates you to subhuman status. But it is your own mind, more than any advice, that can tell you what ideas are powerful. Your own insights enable you to integrate important experiences. An idea is powerful, then, if it gives form to your understanding of life. It follows that you cannot inspire others with an idea unless it has first inspired you.

Interconnection of Knowledge

What Papert labels the "syntonic" characteristic focuses on how an idea assumes power within the mind of an individual. An idea is powerful for a person if it relates and unifies knowledge gained in diverse experiences.

An idea gains power if it can be reduced to a concrete model that serves as a metaphor for the interpretation of subsequent problems. Such a model helps explain which aspects of new problems must be considered, which may be neglected, and which anomalies must be explained away on a basis of local evidence. Models prove more or less powerful depending on the individual's interests and experiences.

The most essential characteristic of powerful ideas is their relation to the individual's previous knowledge. You can tell students that one situation resembles another, but recognition of such comparisons is more powerful if it is the students' own discovery. They will make the connections between the structures of one idea and another at a level of detail appropriate to their specific prior knowledge and feelings. This internalization is the basis of an idea's power for the individual.

An analogy may help here. If you solder a connection at too low a temperature, you can get mechanical binding but undependable electrical contact. Ideas imposed by instruction are like badly soldered joints. Only the individual has the power to fuse connections between new ideas and his or her own most personal thoughts and feelings. These connections alone can make an idea an important part of how the person sees the world and behaves in it.

Paradoxically, an excellent way to harness the students' understanding for engagement with ideas is to liberate their expressiveness. Because Logo is a vehicle for free exploration, knowledge built from Logo is syntonic, appropriate to the person, and experienced as an authentic, intimate part of the self. Such is the power of an approach to learning that frees the individual to create within a social context that makes our culture's most powerful ideas accessible.

I/O and Applications Design

An application design negotiates between a specific objective and the potential of the equipment. Computers are general-purpose symbol manipulators, so they can deal

abstractly with an idea. What any computer system can do in an interesting way, however, depends on its input/output (I/O) devices. Look for something special about a machine's I/O to suggest the kinds of neat phenomena the system could exhibit. Consider these examples from previously implemented Logo systems:

- The accessibility of the robot floor-turtle world to a child's physical intervention can lead even a small child into simulating the turtle's actions and into debugging procedures (after fixing a procedure "manually," a child can become more engaged in fixing it symbolically).

- Turtle graphics—whose appeal depends largely on the emergence of patterns from simple procedures that command the drawing of many lines—came into its own only with the general availability of bit-map-based displays.

- Logo on the GTI-3500 had a significant potential for engineering and physics simulations because a hardware-implemented "spin" primitive extended the forward and right primitives of "classical" Logo.

- The TI 99/4A joins together a general-purpose microprocessor (where TI Logo is implemented) with a special-purpose graphics processor that manipulates the sprites that give TI Logo its most striking effects.

As increasingly powerful microprocessors become affordable, the special quality of each will bring new potential for creating engaging microworlds. More powerful microprocessors and graphics slave processors may, for example, bring molecule modeling within reach. Local networks of small machines may permit group simulation of economic and political situations (as in games) that are now too abstract, rule-driven, and theoretical to interest many young people. There will continue to be opportunities for creating microworlds around the most powerful ideas of contemporary science and technology.

bridge between less precise and more formal systems. The commands of Logo are designed to communicate with a computer and its output devices, but the extension of Logo through procedures whose names are natural-language words can make the objects and actions more comprehensible. This ability to be extended is a key feature for young children.

But Logo is only a quasi-natural language; a Logo procedure must run on a machine. Further, the objects of a Logo microworld are formal; they can be completely defined by a specification of their state variables. One of the simplest of these objects is the Logo screen turtle. Once you have specified the turtle's location, heading, and pen position, there is no more to say about it. The operations of a microworld are also completely specifiable in terms of the effects they have on state variables. The RIGHT and LEFT commands, for example, modify the heading of the turtle but do not affect its location. Given the

object orientation of Logo and the ease of specifying the interaction of state-change operations with state variables, a first criterion for the quality of any Logo implementation (an application microworld or the interpreter itself) is the clear presentation of the state variables to someone using the system. Two examples of representation inadequacies in TI Logo can clarify the point: although the heading of a sprite is a significant state variable, it cannot be determined by inspecting the object's appearance (the shape carried by the sprite) when its velocity is zero; it is impossible to determine visually which sprite is the "current" object, i.e., the one or ones that will respond to the next Logo command. Ideally, the equivalent of a SHOWTURTLE/HIDETURTLE set of commands would show which is the active sprite. Whatever the limitations of a specific Logo implementation, anyone who designs a computer-based microworld should strive to represent all the state vari-

Objects in Microworlds

Logo procedures can serve as a

ables in a visible, obvious way. Doing so enhances the comprehensibility of the ideas embodied in the objects and their manipulations.

Comments on a New Microworld

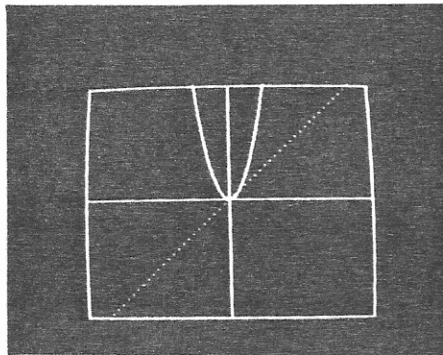
One of the objectives of Logo is to put power in the hands of beginning users. Even powerful ideas usually come from striving to reach a simple, down-to-earth objective. To demystify designing a microworld, I would like to present a few comments on some work in process. I wanted to develop an effective way to present some ideas of algebra to a 12-year-old. I remembered a casual comment of a former MIT Logo colleague, Andy diSessa, that one of the most powerful ideas accessible through Logo was embodied in "procedures that output." At the time, I was mystified, even though Andy had explained that his comment was based on the fact that such a procedure was equivalent to a mathematical function. That observation came back to me. Algebra is about mathematical functions. Although I couldn't fully appreciate Andy's comment, it focused my attention on a personally comprehensible way of expressing mathematical functions in Logo.

Common mathematical functions assign the value of one variable (call it y) to the value of some expression based on another variable (call it x). Assigning values is just what the MAKE command does. If a superprocedure controlled assigning to y the value of an x -based expression for the domain of possible values of x , it would generate any function expressible in the Logo language. When given two inputs (x, y), the DOT primitive of Apple Logo draws a dot at the screen location of those coordinate values. If the value of x is incremented across the domain of possible x -coordinate values, and y is specified in terms of the value of x , DOT can be used to plot discrete sketches of mathematical functions. A second method of drawing functions is possible. If those "dotted" locations are used as the position coordinates of a SETTURTLE (SETPOSITION) command, the screen turtle will

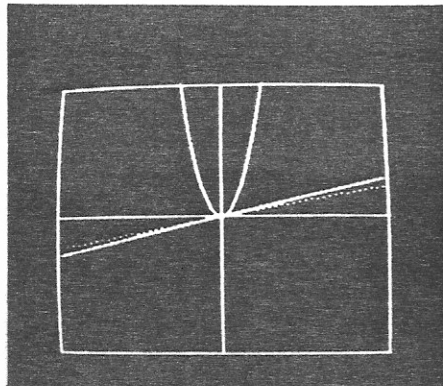
draw a line-segment approximation to a mathematical function. These are the ideas around which the PLOTTING microworld is constructed. Photos 2a and 2b show examples from the PLOTTING microworld.

How can a person go from common experiences to a new idea by doing something only slightly unusual—but with that small difference providing access to a range of significant

(2a)



(2b)



Photos 2a, 2b: Two examples of the PLOTTING microworld. Photo 2a shows the contrast between the straight-line plot of $y=x$ and the parabolic plot of $y=2x^2$. Photo 2b contrasts the same parabolic plot with a plot of $y=.25x$ (segmented line), an attempt to fit the slope (heavy line) of the parabola at a point. See listing 2 on page 158 for the DOTPLOT procedure from the PLOTTING microworld.

phenomena? Think about what kinds of experiences younger students might have had that could support learning about mathematical functions. Any child who uses Logo for a while learns to define specific variable values using the MAKE primitive; for example:

```
MAKE "MY.NAME "BOB
MAKE "MY.AGE 42
```

The minimal significant complication possible in the specification of a variable is to make its value depend on something else, such as keyboard input. It is common for beginners to write routines such as the greeting below for inclusion in some more ambitious program:

```
TO GREET
  PRINT [WHAT'S YOUR NAME ?]
  MAKE "WHO READWORD
    ;accept keyboard input
  PRINT [SENTENCE
    [GLAD TO MEET YOU,] :WHO]
END
```

We can start with nonarithmetic examples of variables as functions of other variables. They can be simple or complex. Graphs of equations can be viewed as another, more specific form of a familiar kind of relation—a new representation for a familiar idea. The algebraic formulas with which we usually associate the graphs of equations are seen as another description of a correspondence relation, a description that is specific and limited, but very powerful.

Making clear the connection between concrete uses of programming variables and mathematical functions is one justification of a PLOTTING microworld. This idea is one I judge to be powerful. The programming needed to make a Logo PLOTTING subsystem is nearly trivial (see listing 2), but that is precisely the virtue of a powerful language: its expressiveness makes ideas and functions stand clear of accidental complications.

Extending the PLOTTING World

If we look beyond the simple plotting of functions, the intellectual extensions of such a microworld can be simple and striking. Consider these two possibilities. First, when the domain of x is specified with beginning, end, and increment or step-size (to control the grain of the plotted function), the slogan through which continuity is often expressed becomes an almost obvious consequence of the "dotted" representation: "you give me an *epsilon*, and I can give you a *delta* such that whenever the difference between successive values of x

Listing 2: The DOTPLOT procedure, written in Logo. The procedure plots y as a function of x for values of x incremented by CHANGE. Resulting plots appear in photos 2a-2b.

```
TO DOT.PLOT :CHANGE
MAKE "X - 135
PRINT [MAKE "Y A FUNCTION OF :X]
MAKE "FUNCTION READLIST
LABEL "AGAIN IF :X > 135 [STOP]
RUN :FUNCTION
IF NOT OR :Y > 119 :Y < -119 [DOT SE :X :Y]
MAKE "X :X + :CHANGE
GO "AGAIN
END
```

Listing 3: The LJ1 procedure, written in Logo. Using the variables COEFF1, COEFF2, and CHANGE, the procedure draws Lissajous figures like those shown in photos 3b, 3c, and 3e.

```
TO LJ1 :COEFF1 :COEFF2 :CHANGE
MAKE "ANGLE 0
MAKE "FUN1 (SENTENCE [MAKE "X 100 * SIN :COEFF1 * :ANGLE])
MAKE "FUN2 (SENTENCE [MAKE "Y 100 * COS :COEFF2 * :ANGLE])
PENUP RUN :FUN1 RUN :FUN2 SETPOS (SENTENCE :X :Y) PENDOWN
LABEL "AGAIN MAKE "ANGLE :ANGLE + :CHANGE
RUN :FUN1 RUN :FUN2
SETPOS (SENTENCE :X :Y)
GO "AGAIN
END
```

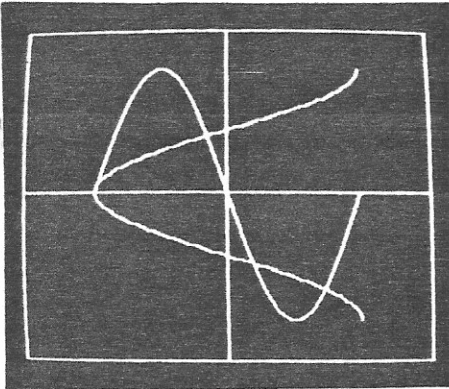
is less than *delta*, the difference between successive values of y will be less than *epsilon*." Second, consider the implications for understanding the differential calculus. When plotting the value of a function, it is simple to save the value of the prior point-couple and calculate the slope of the function. This is an empirical form of differentiation. A microworld of plotting tools (whose activities could include plotting functions, the empirical derivation of slopes of those functions, and curve

fitting—with the plotting tools—to those empirically derived slopes) could provide a body of practical experience about the relations between functions and their slopes. This experience, for which differential calculus will later provide a theory, will make the calculus easier to appreciate and assimilate.

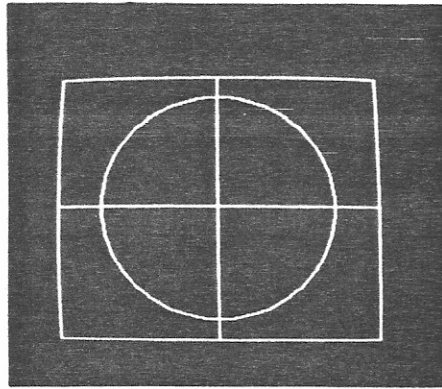
These ideas may interest a math teacher or a psychologist, but would any child be interested in plotting mathematical functions? Are there any accessible neat phenomena? This

is the most important final question the creator of every microworld must face. The concrete appeal of this microworld must be the creation of appealing (and possibly puzzling) graphic designs. The beauty of turtle-geometry designs derives from the use of repetition and variables in simple procedures. This observation suggests that we look at repeating functions such as those produced by the sine and cosine primitives. Photos 3a-3f show six designs made from combinations of sine and cosine functions. These designs, generically known as Lissajous figures, are my candidates for neat phenomena of the PLOTTING microworld. [Editor's Note: Named for French physicist Jules Lissajous, each of these figures consists of the series of plane curves traced by an object that executes two mutually perpendicular harmonic motions. . . . P.L.] The method of the procedure shown in listing 3 is to calculate a screen location with x as a sine function of an angle value and y as a cosine function of the same angle. The design is made when the turtle draws a line as it moves from one calculated location to the next one. The procedure is stopped manually.

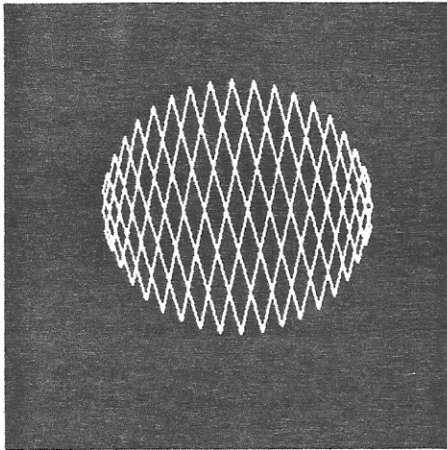
Lissajous figures are similar to POLYSP1 designs in general character because they are made of line segments that show natural classes or families of shapes and occasionally emerge as surprisingly beautiful. Like INSP1 designs, they are somewhat mysterious to those who think more concretely than formally. Are they



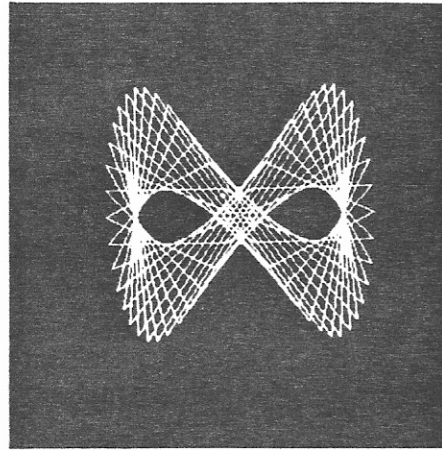
(3a)



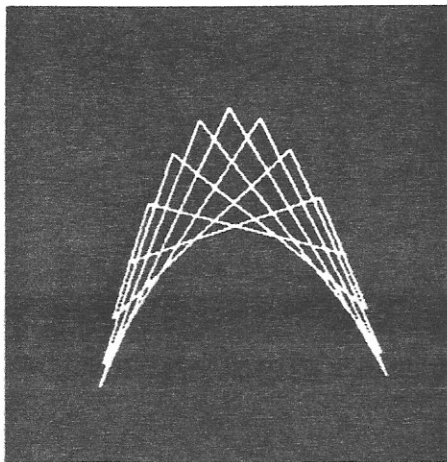
(3b)



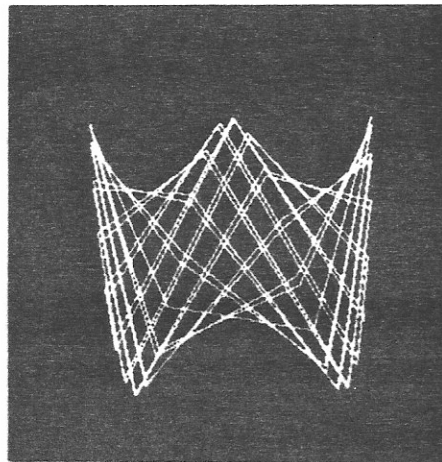
(3c)



(3d)



(3e)



(3f)

Photos 3a-3f: Six more designs from the PLOTTING microworld. The author regards designs made from sine and cosine functions as neat phenomena; i.e., phenomena inherently interesting to observe or interact with. In each design, a procedure calculates a screen location with x as a sine function of an angle value, and y as a cosine function of the same angle. The turtle draws a line by moving from one calculated location to the next.

Photo 3a shows a plot of $y = \sin x$ and $x = \cos y$. The designs in photos 3b, 3c, and 3e were plotted using the LJ1 procedure shown in listing 3, with the following respective sets of values for the three variables (COEFF1, COEFF2, and CHANGE) in the procedure: 1, 1, 1; 77, 23, 9; 1, 2, 84. The design in photo 3d was plotted by a similar procedure that plots cosine against sine, using the values 13, 26, 6. Photo 3f resulted from another similar procedure that plots cosine against cosine, in this case using the values 7, 28, 43.

"neat" enough? Will their appeal be universal? Such questions are clearly impossible for me to answer. If this microworld is of limited interest, perhaps you will have a better idea. More power to you!

The Challenge to Educators

If Piaget's vision that education should involve active discovery of reality is correct and Papert is right in saying that computer-based microworlds provide a solution to the central problem of education, the challenge of education will be more technical in nature than theoretical or ideological.

Clearly, the computer revolution is having a significant impact on education. But that revolution is only worthwhile if it liberates people, which it can by offering educators two remarkable opportunities: with computer technology, teachers will be able to help children expand their love of learning; in turn, teachers will achieve a kind of professional status long denied them. Teachers, programmers, and other microworld designers will be the architects of inner space, proposing ideas and creating tools that will enrich our minds. ■

Acknowledgments

Many people in the extended Logo community have contributed to the ideas of this paper. Seymour Papert's influence is central and obvious. Conversations with people who have worked for or visited Logo Computer Systems in Boston and New York have inspired observations or explanations. Dan Watt's editorial comments have been helpful. The article was revised for publication while the author was a consultant to Le Centre Mondial L'Informatique et Ressources Humaine. Thanks to Guy Montpetit for use of the text processor and to the Spencer Foundation for support during a period when these ideas were developing.

References

1. Lawler, R. *One Child's Learning: an Intimate Study*. Unpublished doctoral dissertation. MIT, 1979.
2. Lawler, R. "Some Powerful Ideas." This collection of short pieces is serialized in *Creative Computing*, April, 1982 and following months.
3. Papert, Seymour. "Computer-based Microworlds" in *The Computer in the School: Tutor, Tool, Tutee*. Robert P. Taylor, ed. New York: Teachers College Press, 1980.