Logo: A Language for All Ages

Michael Tempel
Logo Foundation

This article originally appeared in *Special Issue: Computer Science K–8: Building a Strong Foundation*

Logo is most often thought of as a computer environment for young children with an emphasis on graphics. But it is much more. Logo is a sophisticated programming language, a dialect of LISP, that can be used by learners of all ages in a wide variety of ways.

A guiding principle of Logo development has been that it should have a "low threshold" and a "high ceiling." You should be able to enter the Logo room easily with no hurdles to jump over and no big step up. Then, once inside you should be able to move seamlessly from simple explorations to complex projects.

A second goal is that the Logo room should have "wide walls." That is, people with different interests, tastes, and learning styles should all be comfortable. There should be a variety of domains in which to develop projects with Logo.

In the 45 years since Logo began, the room has gotten bigger. The threshold is lower, the ceiling is higher and the walls have moved outward. To see how this has been happening we can begin with Logo as it was known to most people in the early 1980s when it emerged from the research environment at MIT and found its way into schools and homes. The most popular versions of Logo at that time were for the Apple][. The turtle geometry component was by far the most widely used.

Young children could begin exploring shapes and write procedures to draw them. Simple shapes could be combined into more elaborate designs. Older students could use the turtle for complex mathematical explorations[1].

Turtle geometry is just one domain for Logo explorations and projects. During the early years of development Logo was used in many areas including music, robotics and language[2]. In fact, the first versions of Logo had no turtle. The name Logo, which means "word" in Greek, was chosen to emphasize that the language was well-suited to working with words and sentences in contrast to the numeric focus of most programming languages at the time[3]. Over the past 30 years, hundreds of versions of Logo have been developed[4] with increasingly diverse capabilities.

Animations and Games

While turtles were crawling around on Apple][screens, personal computers that doubled as game machines - such as the Atari 800- supported colorful multiple turtles, also called "sprites," that could wear different costumes and be set in motion. Animations and video games emerged as the favored projects on these machines[5]. This functionality is now standard in many current versions of Logo, including MicroWorlds and Scratch. These modern implementations also include drawing tools and allow the importing of different kinds of media – images, video, sound, and music.

Robotics

In the mid-1980s, work on robotics versions of Logo was underway at the MIT Media Lab. Logo programs received information from light, touch, and other sensors, and

activated motors and lights. LEGO TC Logo and Control Lab were widely used products that grew out of this research.

In the early 1990s work began on Programmable Bricks. The Brick, which you could hold in your hand, had a microprocessor inside. As with the earlier Logo robotics environments, the Programmable Bricks worked with sensors, motors and lights. But now, with a downloaded program the Brick could be disconnected from the computer and on its own, as part of a vehicle exploring its environment, for example. The RCX and NXT from LEGO grew out of this project along with several types of Crickets, which are smaller programmable bricks[6]

Modeling and Simulation

Mitchel Resnick's 1994 book *Turtles, Termites and Traffic Jams* described Star Logo, a system he developed to allow exploration and modeling of decentralized systems. Star Logo has thousands of turtles that can be programmed to interact with each other and with patches of background. One can simulate the emergent behavior of, for example, a termite or ant colony, or explore how traffic jams form or forest fires spread. A similar program derived from StarLogo is NETLogo[7].

Blocks Programming

In 2006 a new version of Logo called Scratch was developed by the Lifelong Kindergarten Group at the MIT Media Lab[8]. It was designed to enable creation of games, animations and multimedia projects. The key difference from earlier versions of Logo that had such capabilities was the use of Blocks Programming. Instead of lines of text, programs are constructed by snapping together blocks that fit into one another like jigsaw puzzle pieces. Program structure is represented visually. A great advantage of Blocks Programming is that it is almost impossible to make the kinds of syntax errors, common with text programs, that result from typos and incorrect punctuation. The different types of blocks are of different shapes and fit only where they are syntactically appropriate.

Blocks Programming did not originate with Scratch. It was first developed in 1995 as Logo Blocks for the Programmable Brick, which is still in use today. However, the immense popularity of Scratch has brought Blocks Programming to the forefront in the educational technology community.

Another important aspect of Scratch is the community that has grown around it. The Scratch Web site has over a million members and almost two and a half million projects that have been posted. There are also forums for discussion and sharing is part of the Scratch culture. Many projects are developed by remixing those that were previously posted.

Logo is widely used at the elementary and middle school level where it can be integrated into many subjects. This is less the case at the secondary and college levels, with the exception of StarLogo and NetLogo[9].

Logo can also be used to teach a formal computer science course[10]. Even if Logo is not the language of choice for such a course the earlier experiences that students likely have had with Logo can be tapped. Concepts that students encounter in such a course including, structured programming, algorithms, data types, and objects may have been met in an informal way in earlier work with Logo. Secondary teachers may draw on this experience.

After 45 years Logo remains a language for learning for people of all ages and accommodates an ever widening variety of interests and learning styles.

---

[1] For advanced applications of turtle geometry see Abelson, Hal and diSessa, Andrea *Turtle Geometry*, MIT Press, 1980

[2] The variety of research areas during the 1970s may be seen in the MIT Logo Memos. Many of these are available at http://www.sonoma.edu/users/l/luvisi/logo/logo.memos.html

[3] Goldenberg, E. Paul and Feurzeig, Wallace *Exploring Language with Logo*, MIT Press 1987

[4] For a comprehensive list of current and past versions of Logo see the Logo Tree Project http://www.elica.net/download/papers/LogoTreeProject.pdf

[5] See Tempel, Michael and Chafiian, Hope "Computer Games for Kids, by Kids," *Logo Update*, Volume 7 Number 2, Spring 1999
http://el.media.mit.edu/logo-foundation/pubs/logoupdate/v7n2/games.html
and Part 2 in Volume 7 Number 3, Summer 1999
http://el.media.mit.edu/logo-foundation/pubs/logoupdate/v7n3/games.html

[6] Currently available crickets include the Super Cricket - http://gleasonresearch.com/prod.php?sku=SUPERCX
and PICO Cricket - http://www.picocricket.com/

[7] See NetLogo http://ccl.northwestern.edu/netlogo/ and
StarLogo http://education.mit.edu/starlogo/

[8] The Scratch Web site is at http://scratch.mit.edu

[9] This is not due to any inherent limitations in Logo, which has been used at the high school level, especially for mathematics. See Cuoco, Albert *Investigations in Algebra* and Lewis, Philip G. *Approaching Precalculus Mathematics Discretely*, both MIT Press, 1990

[10] See Harvey, Brian, *Computer Science Logo Style*, MIT Press 1984, second edition 1997. This three-volume book by is an excellent basis for a computer science course using Logo.