



development among young children with Jean Piaget (1896–1980), the leading educational psychologist of his generation. On moving to MIT, Papert began to work closely with an artificial intelligence expert, Marvin Minsky. In his work with LOGO, Papert attempted to bring the ideas of both his colleagues together, uniting theories of cognitive learning and artificial intelligence.

Work on LOGO continued throughout the 1970s, and other groups were set up to experiment with the new language. The most notable of these was based in Edinburgh. All of this development work was carried out in university research departments using mainframes or minicomputers. It was only with the arrival of microcomputers that LOGO became more widely available.

LOGO is a sophisticated language that needs a lot of memory, both for the code and as working space. The LOGO interpreters found on micros typically require around 30 Kbytes of memory, and another eight Kbytes or more for the graphics display. All of this before you even start programming! So, although it was possible to implement simple BASIC interpreters on home micros from the moment they were marketed, it was not until home computers with over 48 Kbytes of RAM were widely available that LOGO on micros became a feasible proposition.

But it was Seymour Papert's *Mindstorms* (Basic Books, 1980) that took LOGO out of the research departments and brought it to the attention of a much larger group of people. In his book, Papert develops a vision of how computers might be used in education. This is a result of the synthesis of three sets of ideas: theories of cognitive development, artificial intelligence and the movement in education towards child-centred learning. Papert wants to see children programming computers, rather than computers programming children (which, he argues, happens in most 'computer aided instruction').

The book looks forward to the emergence of a new 'computer culture', in which 'formal' ideas previously considered beyond the capabilities of children will be easily handled by them. They will be able to do this because of the way they have used computers to explore formal ideas. It is this active, co-operative (pupil-to-pupil and pupil-to-teacher) and unstructured exploration of ideas that constitutes the 'LOGO philosophy' underlying the language's use in education.

Papert writes and convinces by the sheer power of his rhetoric. However, there are a number of serious problems with the theory. There is very little experimental evidence to back it up, despite a number of studies; Piaget's theories of cognitive development are turned into a prescription for education in a way that Piaget never intended; and there are problem-solving areas (even in mathematics!) that LOGO doesn't cover.

As teachers use LOGO more widely in the classroom, they are finding that not everything works in the way that Papert describes, and they aren't getting the results they had hoped for. There



is a danger of disillusionment, but once we set aside the over-enthusiastic claims of what the language can do, LOGO still remains an excellent way of introducing computer concepts, of exploring certain kinds of ideas and developing problem-solving skills.

LOGO on present day micros has too little workspace and runs too slowly. To some extent it is a language that is waiting for the hardware to catch up with it. But as a language for learning it has no serious rivals.

Who Is LOGO For?

Who can benefit from learning to program in LOGO? We feel that many people, even experienced programmers, can learn a great deal from LOGO programming, including:

- Anyone who is new to computing, or new to programming;
- Anyone who likes playing with computers, and thinks computers ought to be fun to use;
- Anyone frustrated by a lack of expressive power in another programming language;
- Anyone who has an interest in thinking about thinking, learning, or teaching;
- Anyone who wants an insight into more advanced areas of computing, especially those that are related to the study of artificial intelligence.

Having said this, we must remind you that, like BASIC and machine code programming, LOGO is not for everyone. Specifically, LOGO might not be the best language for:

- Anyone who thinks using computers is 'work'. Some languages are designed for work, as are cart horses. But a cart horse is hardly the one you would choose for an afternoon ride in the countryside;
- Anyone who needs or expects a great deal of speed from the computer as it processes instructions. LOGO uses a lot of memory, and runs slowly on the present generation of microcomputers. (On comparable programs, LOGO can run at half the speed of BASIC.)

Even for these groups, however, a knowledge of LOGO can be very valuable. LOGO can be used to sketch out a solution to a problem and prepare it for translation into another language.

Authorised Versions

Shown here are the most comprehensive and well-documented versions of LOGO for the Commodore 64 (Terrapin-MIT), Sinclair Spectrum, and Atari computers (LCSI). Although expensive, these are the manufacturers' authorised versions of LOGO and will most closely resemble the original MIT language. Commodore 64 LOGO is available on disk for £34.95; Atari LOGO comes on cartridge for £59.95; and Sinclair LOGO is a cassette-based version for £39.95. There are less expensive LOGO programs on the market. Some, particularly for the Spectrum, cost as little as £8.95

Coming soon . . .



the Turtle