1982

LOGO and turtle robots

P. J. McKerrow
University of Wollongong, phillip@uow.edu.au

LOGO and Turtle Robots

Phillip John McKerrow

Department of Computing Science,
The University of Wollongong,
Post Office Box 1144,
Wollongong, N.S.W. 2500
Australia.

ABSTRACT

LOGO, a computer language in which communication with a turtle takes place, is being used to teach computer awareness, logical thinking, computer programming and mathematics. It is used to create an environment where learning is both natural and pleasurable.

LOGO is a practical application of Piaget's fundamental assertion that children learn by doing and by thinking about what they do. Some researchers are investigating the effectiveness of LOGO in a classroom situation while others are applying the basic concepts in new areas in an attempt to develop computing and robotics environments.

Keywords LOGO, turtle, robot, education, learning environment, programming, mathsland, turtle geometry, robotics, procedure, modularisation, models.

1. Introduction

Every so often a piece of basic research is carried out that, as it matures, has a significant impact upon the lives of people in a variety of disciplines. Seymour Papert's research [1] into teaching children mathematics is influencing teaching in a number of areas. Papert's research included the development of LOGO, a computer language in which communication with a turtle takes place. Some turtles are abstract objects that live on screens (figure 1). Others, turtle robots (figure 2), are physical objects that can be picked up, or patted, like any mechanical

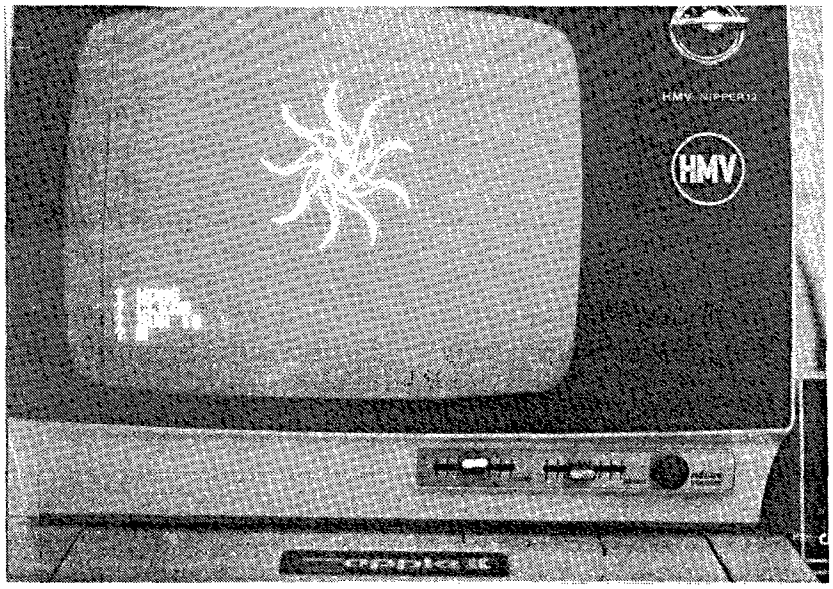


Figure 1. Screen-turtle executing a program to draw a sun (figure 7)

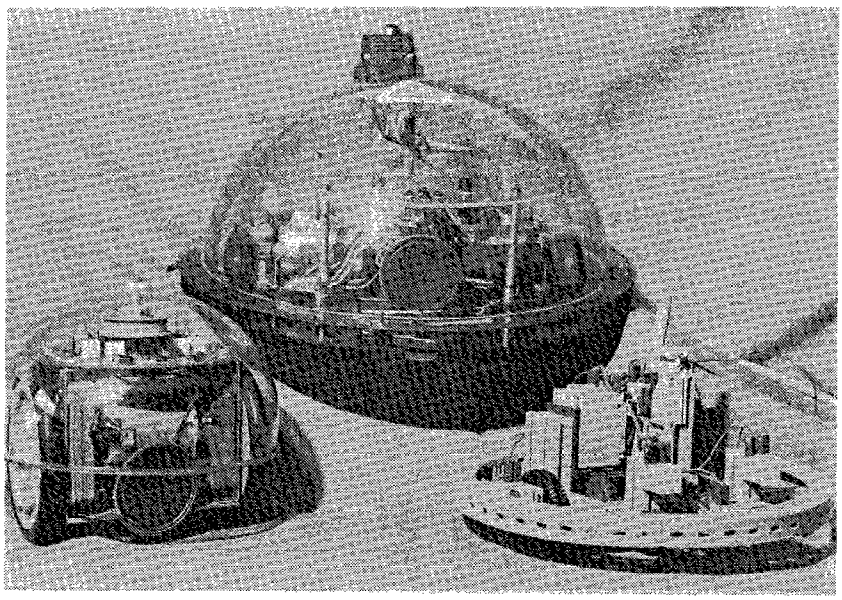


Figure 2. A school of turtles - left to right Wolt from Terrapin, Tassie the Tasman, and Trugon out of Fischertechnik.

toy.

In the Australian state of Tasmania [2,3], LOGO and turtle robots are being used extensively in introductory programming and computer awareness courses at both primary and high school levels. At the University of Edinburgh, Department of Artificial Intelligence [4], LOGO is being used to teach children mathematics as part of a research project to determine why mathematical ability improves through programming. Papert's group went on to develop a whole new area of mathematics, turtle geometry [5].

Some have taken the basic theory developed by Papert and applied in other areas including teaching the physically and mentally handicapped [7]. Research is also being conducted into developing a computing environment [8] where children learn computing naturally in contrast to Papert's mathematical environment or a normal language environment.

2. Papert's Research

Papert, a student and colleague of Piaget [9], is convinced of Piaget's fundamental assertion that children learn by doing and by thinking about what they do. Exploration of their environment leads to creative thinking about that environment. From his childhood he thought in terms of models and formulated the idea that a concept is easy to understand if you can assimilate it to your collection of mental models. This raised, recursively, the question of how one learned these models. His research led him to the conclusion, that immersing a child in the correct environment would produce a natural situation where the child would be stimulated to learn, just as children learn language because they live in a language environment.

From this came the vision of a place called Mathsland; a mathematical environment to be created using computer technology and working physical models. He saw that computers would bring a revolution in education only as new methods, utilising the power of the technology, were developed. Dressing up the same old stuff in shiny new technology does not improve either the way it is taught or the effectiveness of the teacher. A new technology requires new creative teaching methods.

The idea that children learn by doing gave rise to two directions of research:

(i) Activities that allow a child to manipulate and extend his environment had to be invented. Physical models for children to master and apply to projects, where unexpected events may lead to new discoveries, were to be designed. The opportunity to create, manipulate and extend gives the child a sense of power of applied knowledge and a self-confident realistic image of himself as an intellectual agent. The turtle robot is a particularly good example of a physical model that provides better things for children to

do.

(ii) Giving children better ways and means to think about what they do involves looking for activities whose structure allows a child a clear view of his own intellectual activity and so help him to achieve a more articulate understanding of doing. Fundamental to creative activity is the heuristic process of exploration and making hypotheses. The LOGO language and associated turtle geometry were designed to enable this creative process.

3. LOGO

The human engineering of LOGO is beautiful. It develops from the known to the unknown, from the concrete to the abstract. The abstract computer concept of a variable is one of the last things to be learned, not the first as in most other languages. LOGO starts with a concept a child is familiar with; motion within space. As children move and have a sense of position they can readily relate to the turtle's movements.

The turtle can move, draw with a pen, toot, flash lights and sense obstacles in response to simple LOGO commands (figure 3) entered interactively from a keyboard. From the start the computer serves the child. It is not uncommon to see children pretending to be turtles in the school play ground after an introductory class.

Initially the child keys in a command and watches the turtle respond, finding that the turtle obeys his command. From this simple beginning, student tasks increase in complexity while each level of complexity is complete in itself making the experience personally rewarding. Soon the student sets about accomplishing specific tasks. To draw a square, initially (figure 4) ten commands are given; the turtle responding after each one. Instruction sequences like this soon become tedious to enter and the student seeks a more efficient way. A simple procedure mechanism allows a new word to be added to the language (figure 4) so that now in response to the keyword TRIANGLE the turtle draws the complete triangle. Thus the student is painlessly introduced to the concepts of modularisation and design. The definition of TRIANGLE remains in memory as long as the student wants it. Also it can be saved on a disc-file for future use.

Each new command that is defined becomes part of the language and can be used to define further commands. Thus a new level of the language is defined, consisting of modules developed from the basic command set. In this way the language is extended by the student who tailors it to meet his individual interests and abilities.

Combining modules allows the students to design and experiment with more complex structures. Mistakes made during

construction of a structure are immediately obvious increasing the child's understanding of the complexity of the problem. For example a square and a triangle can be combined to draw a house (figure 5) but only one combination is correct. If the turtle is not located at the correct position when it starts to draw the triangle, the roof will not sit neatly on top of the walls to form a house. A street can be drawn by placing several houses side by side and then a suburb by placing several streets side by side.

Once again the program is starting to become cumbersome and keying in many lines of code tedious. In each of the above examples there is considerable repetition. The student now progresses to the programming concepts of loops and recursion. The keyword SQUARE can be defined in two different ways (figure 4). Now more complex geometric shapes including polygons, curves and sodems can be constructed but the iteration never stops.

A new concept, the idea of a counter, is introduced (figure 6) allowing the loop to be terminated after the required number of iterations. The REPEAT construct is not available in some implementations of the language requiring the IF THEN control structure to be used (figure 7). From here it is a simple step to the abstract concept of a variable. Up to now the figures have been of a fixed size. By use of program variables both figure size and loop counters can be altered allowing one procedure to cover many circumstances (figure 7). Introducing arithmetic and logic functions enables decision making and recursive definition of procedures. In this environment, the understanding of simple recursion is natural.

Research studies [10] of children learning LOGO indicate that pupils normally go through three stages in learning to program and to solve problems. The first is a copying stage where the pupil is eager to produce pleasurable effects without understanding how these effects are achieved. The second stage is characterised by a growing concern to write programs in a "professional" LOGO style. In the final stage, achieved only by some pupils, the system is perceived as a means to an end not as an end in itself. This is the creative stage where the programming is less polished but more relevant to the problem at hand.

Where LOGO is used as an introductory programming language, courses tend to be designed such that students leave LOGO at the second stage and move onto other languages. This highlights the problem that many people involved with computers get so wrapped up in computing that it becomes an end in itself. To effectively teach computer literacy the transition to the third stage is necessary. Those using LOGO as a tool to create a mathematical environment make this transition by entering the world of turtle geometry. Along the way many basic geometric concepts are learned in addition to skills in problem formulation, problem solving, logical design methodology, program modularisation and program debugging.

4. Turtle Geometry

To the uninitiated the idea of a computational style of geometry based on the motion of a turtle seems ludicrous. However, just like other geometries it is based on a set of fundamental concepts and has developed to the point where a complete text [5] has been written. This text is a delightful introduction to a wide variety of geometric concepts and models, enabling the reader to explore the realm of turtle geometry.

One of Euclid's fundamental concepts is the point, which is an entity that has position but no other properties. The fundamental entity of turtle geometry is the turtle; a dynamic device which has both position and direction. In this the turtle is like a person, or an animal or a machine, enabling it to serve as a first representative of formal mathematics for a child.

The LOGO environment, which is based on fundamental geometric concepts, can be used as a tool to teach turtle geometry by allowing the student to experiment with an infinite range of geometric figures and patterns. From this basis the student progresses to a formal examination of these shapes including the development of theorems. Topics covered include polygons, looping, symmetry, random motion, modelling, vector operations, three dimensional shapes, projections, topology and general relativity.

5. Turtle Robots

In Papert's words [1] a turtle is a computer-controlled cybernetic animal. It is an object to think with, an object in which there is an intersection of cultural presence, embedded knowledge, and the possibility for personal identification. It exists within the cognitive miniculture of the LOGO environment. The turtle serves no other purpose than being good to program and good to think with. Some turtles are abstract objects that live on computer systems. Others are physical objects that can be picked up like any other mechanical toy.

A variety of turtle robots (figure 1) are available. Each has two motors for movement, a pen control mechanism, lights, a speaker, touch sensors located around the perimeter and control electronics. To turn, a turtle robot rotates around its centre, one motor turning forward, the other turning backward. Communication between the computer and the turtle robot [11] can be either parallel or serial, depending upon the design of the turtle. In both cases the connecting cable can get in the way, but the parallel connection is more cumbersome. Educators reject radio or infra-red links on the basis that they add an extra element of magic.

Some turtle robots use stepper motors, others permanent-magnet direct-current motors. To reduce cost both types operate on open-loop, consequently the stepper motor versions give more

accurate and repeatable control. This can be a problem when trying to draw a square but in robotics experiments is closer to nature (how many people can walk in a straight line or turn exactly ninety degrees). Most turtle robots come as complete units from the manufacturer and can not be extended for other applications.

Trugon, a robot being developed as part of a research program, is fabricated from Fischertechnik model construction components allowing a greater flexibility in experimentation. A junior high school child can construct Trugon, gaining an understanding of robotics, mechanics and electronics in the process. Extra sensors can be added or the basic unit redesigned as desired enabling the development of a robotics environment for both teaching and research. The current model performs all the functions of a commercial turtle robot. It can be controlled from a micro-computer parallel input/output port or from a push button box. More sophisticated electronics, incorporating an on board micro-processor, are being designed. A range of sensor and actuator modules, to be used in robotics experiments, are planned.

Robotics experiments ranging from finding a wall (figure 8) to traversing a maze can be undertaken using feedback from the touch sensors. On some turtle robots the touch sensors are operated by a spring loaded bumper bar, on others by the dome which pivots on a central gimbal. When pressing against a wall the dome drags on the wall, as the robot turns, holding the sensor switch closed longer than necessary. Lamp and speaker are used to provide feedback to the student and to make the process more interesting.

The simple wall crawling algorithm in figure 8 is adequate for straight walls and right turns but temporarily looses the wall on left turns. The program should be rewritten to overcome the problem of recursion stack overflow but concurrent operation would be a better robotics solution. Due to the sequential nature of LOGO the program can not test the sensors while the turtle robot is moving resulting in jerky motion. When developing a robotics environment language extensions to facilitate concurrent operation, and to enable additional sensors and manipulators to be added easily, are required.

6. Conclusion

A child's intellectual growth must be rooted in his experience. Computation with LOGO and turtle robots gives children unprecedented power to invent and carry out exciting projects. By doing, so much of what has been perplexing to children is turned into transparent simplicity, the abstract into concrete instruments which can be employed to achieve personal goals.

LOGO is considered to be the best of the existing educational languages. It is being used in many locations to teach

mathematics, computer programming and computer awareness, and as a research tool. A number of research projects in computing, helping the handicapped and robotics are drawing from Papert's work. Implementations of the language exist on a variety of machines from mainframes to personal computers [12,13]. In the absence of a defined standard a number of variations and extensions are in existence including floating point arithmetic and computer music.

There has been relatively little quantitative evaluation of either LOGO or its proponents' claims in the classroom [7,14]. Part of the problem involves developing a rigorous control group methodology for assessing the impact of computers on the educational process. For many teachers the subjective results from using it in a class-room situation is sufficient evidence.

Pupils obviously enjoy working with LOGO providing evidence that it is a powerful teaching tool. Students have little difficulty learning LOGO but experience some difficulty with name/value distinctions [15]. Two common misconceptions are that procedure names have to describe their action in order to work and that variable names are computationally related to their values.

7. Bibliography

1. Papert, S., Mindstorms - Children, Computers, and Powerful Ideas, Harvester Press, 1980.

2. Wagner, R. Turtle Technology to Teach Computing, Panorama, Education Department, Tasmania, Vol. 1, No. 4, November 1978.

3. Wills, S. Computer Education in Tasmanian Schools, The Australian Computer Bulletin, 1980, pp.22-23.

4. Howe, J.A.M., O'Shea, T. Learning mathematics through LOGO, ACM Sigcue, Vol. 12 No. 1, 1978, pp. 2-11.

5. Abelson, H., di Sessa, A. A., Turtle Geometry, MIT Press, 1981.

6. Cohen, H. A., Escher: A Block-Oriented Graphics Language for Microcomputers, National Microcomputer Conference on Personal Computing for the Eighties, ACS, Canberra, 1980, pp.14:1-8.

7. Allison, L., Edmiston, P., A LOGO Survey, The Australian Computer Bulletin, October, 1981.

8. Dromey, R. G., Before Programming - On teaching Introductory Computing, Department of Computing Science, The University of Wollongong, Preprint No. 81/5.

9. Piaget, J., The Origins of Intelligence in Children, International Universities Press, New York, 1952.

10. Howe, J.A.M. Teaching Mathematics Through Programming, Department of Artificial Intelligence Report, University of Edinburgh.

11. McKerrow, P. J., Controlling a Turtle with an Apple, National Microcomputer Conference on Personal Computing for the Eighties, ACS, Canberra, 1980, pp.18:1-11.

12. Nelson, H., LOGO for Personal Computers, Byte, June 1981, pp.36-44.

13 Abelson, H., & Klotz, L., LOGO for the Apple II, Massachusetts Institute of Technology, Report, 1981.

14 Ross, P. & Howe, J., Teaching Mathematics Through Programming: Ten Years On, Computers in Education, North-Holland, 1981.

15. de Boulay, B., & O'Shea, T., Teaching Novices Programming, Department of Artificial Intelligence Research Paper No. 132, University of Edinburgh, 1980.

Turtle State	Turtle Motion
DEVICE n (robot or screen)	FORWARD n
TOOT n	BACK n
BEEP n	LEFT n
LAMP ON/OFF	RIGHT n
PEN UP/DOWN	
FTOUCH	Arithmetic & Program
BTOUCH { Touch	Control
RTOUCH { Sensors	
LTOUCH	+ - * / ( )
	= <> < > <= >=
Procedure Definition	IF...THEN...ELSE...
	MAKE var expression
TO name parm1 ...	GOTO line no.
10 ..... etc	REPEAT n
END	

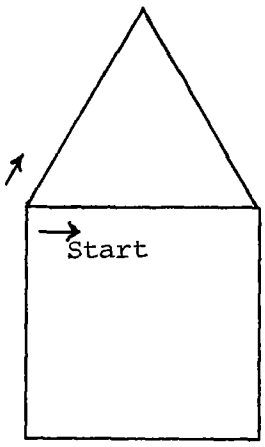
Figure 3. LOGO Language - Basic Command Set

```
PEN DOWN
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
PEN UP
```

```
TO TRIANGLE
10 FORWARD 50
20 RIGHT 120
30 FORWARD 50
40 RIGHT 120
50 FORWARD 50
60 RIGHT 120
END
```

Figure 4. Interactive program to draw a square and a procedure to draw a triangle.

```
TO HOUSE
10 SQUARE
20 LEFT 60
30 TRIANGLE
END
```




```
TO STREET
10 HOUSE
20 RIGHT 60
30 PEN UP
40 FORWARD 60
50 PEN DOWN
60 STREET
END
```

Figure 5. The structure of a house and a street.

```
TO SQUARE
10 FORWARD 20
20 RIGHT 90
30 GOTO 10
END
```

```
TO SQUARE
10 FORWARD 40
20 RIGHT 90
30 SQUARE
END
```

```
TO QCIRCLE
REPEAT 90
10 FORWARD 1
20 RIGHT 1
END
```

Figure 6. Eliminating repetition by iteration and recursion & controlling the number of iterations with a counter.

```
TO ARCR SEG DEG
REPEAT DEG
10 FORWARD SEG/3
20 RIGHT SEG
END
```

```
TO ARCL SEG DEG
10 FORWARD SEG/3
20 LEFT SEG
30 MAKE DEG DEG-SEG
40 IF DEG >0 THEN(ARCL SEG DEG)
END
```

```
TO RAY SEG
10 ARCL SEG 90
20 ARCR SEG 90
30 RIGHT 160
40 ARCL SEG 90
50 ARCR SEG 90
END
```

```
TO SUN SIZE
10 MAKE RAYS 9
20 RAY SIZE
30 MAKE RAYS RAYS-1
40 IF RAYS > 0 THEN (GOTO 20)
END
```

Figure 7. A series of procedures to draw a sun (figure 1).

```
TO FINDWALL
10 FORWARD 10
20 IF NOT FTOUCH THEN (FINDWALL) ELSE (TOOT 6  RIGHT 90)
END
```

```
TO CRAWL
10 IF NOT FTOUCH THEN (MOVE)
20 RIGHT 2
30 LAMP ON
40 CRAWL
END
```

```
TO MOVE
10 FORWARD 2
20 IF LTOUCH THEN (CRAWL)
30 LEFT 2
40 TOOT 3
50 LAMP OFF
60 CRAWL
END
```

Figure 8. Simple Robotics Programs to find a wall and to crawl along it.