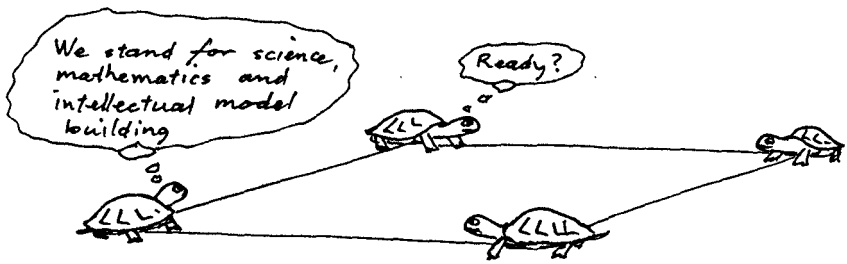
# POALL

## A Journal for Logo Users




## Turtles and Turtles

**Volume 5 Number 3, August 1990**

This issue is taking a long time to write, and as WCCE90 recedes into the past, my hastily scribbled notes become progressively less meaningful. Anyway, there are some brief notes of the Logo stream of the conference. As always, conferences are times to meet people who were simply names before, and this was certainly true of WCCE90, on a grand, international scale.

I sometimes wonder about the future of POALL. It began in 1985 as simply a means for my then students of SACAE Magill to share ideas and programs, but it grew. Mind you, it's been a struggle at times, and more than once I've thought of announcing 'POALL will cease with the ... edition.' It hasn't happened yet, but seriously, POALL needs your contributions.

A first for this issue are some photographs in a special centrefold. Perhaps the better ones are still in the other camera, but I think it's worth a try

Peter

POTS



Address subsriptions and items for publication to: The Editor, POALL
Plympton High School
Errington Street
Plympton S Aus 5038

# Turtles and Turtles

The idea of Turtles being animals and interacting is not new: it was promoted in Abelson and diSessa's seminal work *Turtle Geometry*, published ten years ago. In those days, however, Logo was still restricted to expensive machines; classroom computers were then rare. Even when the first Logos on the Apple and TI-99 appeared, there were limitations, because the Apple Logos had only one Turtle and TI Logo mathematics was integer only.

It is possible to make one Turtle do the work of several (see below), but the four Turtles in Atari Logo or LogoWriter make it much easier to play with Turtles that interact with one another, and the procedures that follow, all written in LogoWriter[1], are based on a portion of Chapter 2 of *Turtle Geometry*.

If the Turtle is an animal, then it has senses. Of course, to accurately model sight and hearing with our limited computing power would be impossible, but with TOWARDS we can have a rough approximation for sight, and we can model smell by measuring the distance between the Turtle and some target. Abelson and diSessa spend some time discussing Turtles with such model senses.

Interacting Turtles can use the same senses, and it soon becomes apparent that our very simple rules and models can lead to some surprisingly interesting and complicated behaviour. Let's begin with a prey animal that is unaware that it is on someone's menu. It moves around in a circle, while the predator uses a simple rule: if the smell becomes weaker, then turn right.

```
TO Predation :pspeed :pturn :fspeed :fturn
RG
TELL 0 ST SETC 4 PU
TELL 2 ST SETC 2 PU
MAKE "old 56
Predate :pspeed :pturn :fspeed :fturn
END

TO Predate :pspeed :pturn :fspeed :fturn
IF 10 > Distance ASK 0 [POS] [STOP]
Predator :pspeed :pturn
Prey :fspeed :fturn
Predate :pspeed :pturn :fspeed :fturn
END

TO Predator :speed :turn
TELL 0 FD :speed
MAKE "new Distance ASK 2 [POS]
IF :old < :new [RT :turn]
MAKE "old :new
END

TO Prey :sped :turn
TELL 2
FD :speed RT :turn
END
```

Try it with various combinations of speed and turn angles, perhaps 90 for the predator's turn angle, and with a speed double that of the prey. Start the two at random

[1] The Apple version, and using the `distance` procedure in TURTLE.TOOLS.

locations, rather than the fixed positions used here. Smell works well when the target is stationary, does it work so well with a moving target? What do you notice about the predator's path? Why do you think this happens?

What happens if the prey is able to detect the predator, ie. if both, in Logo terms, are measuring the distances between them? What if the predator uses sight and the prey smell, or vice versa? That leads to apparently simple 'chase and evade' strategies. The most basic strategy is for the predator to move directly towards the prey, which moves directly away. The result depends on speed in a straight line. But what if they are restricted to a small area, the prey makes random turns, or the predator is able to drive the prey into a corner? Here's the case of the prey heading at 90° from the hunter:

```
TO GiveChase :pspeed :espeed
RG
TELL 0 ST PU
TELL 1 ST SETC 2 PU
ChaseAux :pspeed :espeed
END

TO ChaseAux :pspeed :espeed
IF 5 > Distance ASK 0 [POS] [STOP]
Chase :pspeed
Evade :espeed
ChaseAux :pspeed :espeed
END

TO Chase :speed
TELL 0 SETH TOWARDS ASK 1 [POS]
FD :speed
END

TO Evade :speed
TELL 1 SETH 90 + TOWARDS ASK 0 [POS]
FD :speed
END
```

Again, try with various combinations of predator and prey speeds. You might want to try starting at random positions, especially if your Logo clones all the Turtles at the same spot.

What happens if you make one Turtle or the other controlled from the keyboard? What if one Turtle is attacking a target while the other is defending it? What are the best strategies for both players?

A well known problem is that of the four 'bugs' which start on the vertices of a square, and then, moving at the same speed, each follows the bug to its right. `FBsetup` could be rewritten using `EACH`, instead of the tail recursion used here:

```
TO FourBugs
FBsetup 0
FollowNext 0
END
```

```
TO FBsetup
IF :turtle > 3 [STOP]
TELL :turtle ST PU
SETPOS ITEM 1 + :turtle [[-80 80][80 80][80 -80][-80 -80]]
SETH 90 * :turtle
SETC 1 + :turtle PD
FBsetup :turtle + 1
END

     TO FollowNext :turtle
     TELL :turtle SETH TOWARDS ASK Mod4 :turtle + 1 [POS]
     IF 2 > Distance ASK Mod4 :turtle + 1 [POS] [STOP]
     FD 2
     FollowNext Mod4 :turtle + 1
     END

     TO Mod4 :no
     OP REMAINDER :no 4
     END
```

What is the distance travelled by each Turtle? What happens if they move at different speeds, or if they begin at the vertices of a rectangle rather than a square? What if, instead of following the Turtles on their right, they follow a different order? Do there have to be four on a square, or would three on an equilateral triangle work? What about other regular polygons?

For those with only one Turtle, here's a version of FourBugs:

```
TO FourBugs
CS FS HT
FollowNext [[-100 -100][[-100 100][100 100][100 -100]] 1
END

     TO FollowNext :turtles :colour
     IF 5 > Distance FIRST :turtles [0 0] [STOP
     IF :colour > 4 [MAKE "colour 1]
     SETPC :colour
     PU SETPOS FIRST :turtles PD
     SETH TOWARDS ITEM 2 :turtles
     FD 5
     FollowNext LPUT POS BF :turtles :colour + 1
     END
```

Try this one if you need a Distance procedure (yes, it's a oneliner):

```
TO Distance :target
OP SQRT ((FIRST :target - XCOR) * (FIRST :target - XCOR) +
          ((LAST :target - YCOR) * (LAST :target - YCOR))
END
```

If you want to experiment with different numbers of Turtles you'll need to work out the coordinates of the vertices of triangles, pentagons, etc. What's the easiest way? This is how we do it:

```
TO Polygon :size :vertices :vertex
IF 0 = :vertex [OP []]
FD :size RT 360 / :vertices
OP FPUT POS Polygon :size :vertices :vertex - 1
END
```

To find the vertices of a triangle: `SHOW Polygon 60 3 3.`

Interactions between bodies are important in Physics, interactions between animals are relevant to biology, the list goes on. We can experiment with Logo models.

References:

Abelson, H. and diSessa, A. *Turtle Geometry* MIT Press, 1980
Abelson, H. and Goldenberg, P. *Teacher's Guide for Computational Models of Animal Behaviour* MIT AI Memo No 432, 1977
Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology* MIT Press, 1986

# Current Thoughts on the State of Science Education

## or: Top Ten Things Wrong with Science Education

by Gary Stager, based on the testimony of David Florio - National Science Foundation

1 Not enough elementary school science.

2 Existing science curriculum is dominated by recitation and textbook- *not discovery.*

3 The push for *'More Science'* tends to make things worse.

4 We don't engage enough students through a variety of experiences.

5 Science does not *'happen'* in 45 minutes. There is a critical need for long-term multi-sensory/interdisciplinary projects.

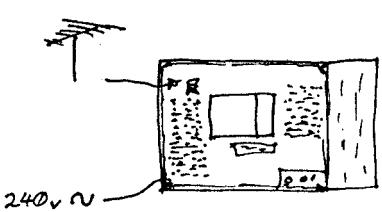
6 Kids don't have adequate skills or tools to do high school science.

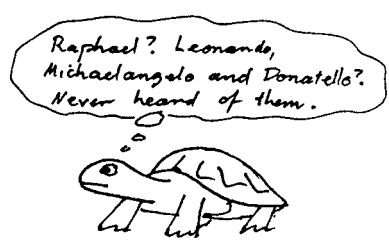
7 Only 7 - 8% of college students can successfully do college-level science.

8 **'Back to Basics' = Back to Simple.**

9 Science and math are not inate abilities.

10 Recent study declared that 75% of all American students did not have adequate science and math skills to prepare them for our increasingly technical world.



# World Conference on Computers in Education

A few brief, and not necessarily coherent, impressions of the Logo presentations.

## Pre-Conference Workshops

I had hoped to be able to sit at the feet of Dan and Molly Watt, Brian Silverman, and others, but found myself tutoring in the LEGO/Logo workshops. I therefore missed some things, but was able to see lots of interesting LEGO/Logo instead.

Stars of the show were the students from Coombabah in Queensland and MLC in Melbourne. They all seemed to thoroughly enjoy themselves, and impress the adults with the way they worked and explained what they were doing. Steve Ocko, Mitch Resnik, Brian Silverman and Gary Stager were there even though Seymour Papert wasn't. Topics were Smart Buggies (David Williams) Primary LEGO/Logo (Greg Grimmet) Sensors and Feedback (Alan Mapp), Traffic Lights (Marg Fallshaw), Cogs and Gears (Debbie Goldman) and Walking Machines (Peter Carter)

One group at the Walking Machines station was determined to build a biped, rather than a statically stable machine. Their solution was to make the legs move rapidly, so that the machine had no time to fall far before the other leg came down again. It turned out as a duck, complete with head and tail, and tended to lose pieces of itself, but it certainly moved, and was a very satisfying experience for all concerned.

The other impressive machine was the scanner, or fax machine. It was a modified plotter (from the kit), with an optosensor in place of the pen. As it scanned, a screen Turtle kept pace, PD and PU corresponding to the optosensor's TRUE and FALSE. The images were distorted (the usual repeatability problem with LEGO machines) but certainly recognisable. The machine spent the rest of the week on the LEGO stand in the exhibition.

The LEGO workshop notes are available as one in the series of Sunrise Notes, from ACER, POB 210, Hawthorn Vic 3122.

I did manage a few minutes with Boxer, but my attempts led only to a couple of crashes. Boxer was running on a set of Silicon Graphics workstations, and it seems the problems could be to do with the LISP compiler. There are apparently plans to implement it on 386 level machines, which would make it a proposition for some schools, but there's obviously a lot of work to do yet. In the meantime, Logo gets better.

## Day One:

Opening keynote speaker was Alan Kay, whose influence has been felt throughout personal computing, indeed, Kay has been called the 'Father of the Personal Computer' from his pioneering efforts at Xerox: Dynabook, SmallTalk, the Star etc.

The text of his address is apparently to appear in *Australian Educational Computing*, but the emphasis was clear from the start: knowledge as things was outmoded, and learning is a process. We saw some video of children using Playground, a visual programming language implemented on Macintosh, and some examples from the Vivarium project, computer animated marine biology. Along the way, we learned that MS-DOS is like a screwdriver without a handle, and that technology is anything that wasn't around when we were born.

## Elementary: Maths

Bruce McMillan: Exploring Two-Dimensional Planes with a Logo Microworld

I missed this session because I was at a multimedia presentation, but the abstract:

> 'This paper discusses children's learning of a particular mathematics concept. Aconstructivist perspective on learning (based on the work of Vygotsky and Piaget) is adopted, and it is suggested that these theoretical frameworks are particularly appropriate both for integrating ideas about children's understanding of mathematics, and for investigating the context within which the learning occurs.
>
> The paper outlines a study involving 20 eight- and nine-year-old children's use of a Logo microworld which deals with the notion of number planes: that is, with the location of points on a two-dimensional surface. The explicit task involved production of a picture comprising any combination of eight different elements, all of which could be located on the screen (and subsequently printed out) using simple ordered number pairs, or coordinates. The implicit task involved learning about ordered number pairs, and an understanding of map-reading skills.
>
> The results will be presented in terms of the change in knowledge about number planes, and the extent to which this knowledge appears to generalise map-reading skills. Some implications for mathematics learning will also be discussed.'

## University: Computing and Other Disciplines

Stewart Denenberg: A Course for the Non-major: Teaching Collaborative Problem Solving via the Classical Computer Science Space-Time Tradeoff

I was being a Geographer at the time, so missed it. Again, from the abstract:

> '...We propose a course which includes the use of a friendly but nontrivial programming language (Logo) coupled to a problem set which demonstrates the efficiency of top-down design as a problem solving technique.
>
> ...In this Alphabet project, student groups design and implement an alphabet of letters that are to be drawn on the screen via Turtle Graphics in Logo. The end result of the project is to make the Turtle trace out variable-sized letters on the screen as the associated keys on the keyboard are pressed. While the instructor supplies the driver program, the students work together to determine a reasonable set of "core shapes" which can be used to assemble each letter of the alphabet. Throughout the four parts of the project design is constrained by assigning "costs" to the memory space and execution time associated with the procedures which draw the core shapes and the letters. Some lively discussions develop around the ideas of primitives and parsimony that are surprisingly similar to those in an upper-level computer science class in programming language design.'

## Elementary: Panel

LEGO/Logo: Building New Worlds in the Classroom

Members of the panel were Steve Ocko, Tom Lough, Liddy Nevile, James Ridgway and Brian Silverman, and chaired by Mitch Resnik, with Seymour Papert contributing from the audience.

Each spoke for a few minutes and then there were questions, with most receiving positive and encouraging answers. A 'devil's advocate' wanted convincing evidence from longitudinal studies to convince him that LEGO/Logo was not just another fad.

## Research: MIT Learning Lab

Mitchel Resnick: Logo: Using Thousands of Turtles to Explore Self-Organizing Behaviour

There were to have been other presentations at this session, but the speakers had been unable to attend. Uri Wilenski and Aaron Brandes were to have described 'The Treasure Game' based on the idea of 'hot and cold', in which children received feedback indicating how far the Turtle was from the treasure. There isn't a paper in the

proceedings, but the study is described in *Constructionist Learning*, edited by Idit Harel, who was also unable to be present, and published by the MIT Media Laboratory.

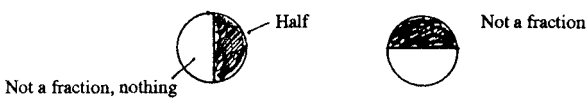
Mitch began by listing a number of examples of 'artificial life' generated by computers: Dawkins and evolution (The Blind Watchmaker), Travers and ant pheromones, Reynolds and bird flocking, Braitenburg and vehicles etc. Logo Turtles and LEGO machines can be given senses, to simulate animals, and Mitch showed pictures of several, and showed a LEGO vehicle with its own on-board processor. Several of the images had been generated by Logo running on a Connection Machine[2], with each of the machine's processors controlling a Turtle.

Among the concepts developed: feedback, equilibrium, multiple levels of description, the role of randomness, adaptive systems, the amplification of small differences.

Seymour Papert

Seymour set out to describe a general perspective of Logo, as an antidote to the algorithmic process, and enabling the exploration of wider fields. Children were to be software designers, rather than simply programmers, and the programming process itself was a part of conceptual development. In the backgound lurked the conflict between Constructionism and Instructionism.

As an example of children designing software, Seymour described how a group designed programs to explain fractions. One child ('Debbie') started out with this concept:




She later discovered that '1/2 hour is half an hour', and then realised that fractions were everywhere. Knowledge had come, not from teachers, but from experience.

The presentation was followed by lively questions.

## Secondary: Problem Solving

Joan Wilcox: Programming and Problem Solving

Unfortunately this one clashed with Mitch Resnick *et al.* The abstract:

> 'Programming and Problem Solving is the name of a strand of the Post Graduate Diploma in Educational Studies (Computer Education) offered to practising teachers by Sydney University's Institute of Education. This strand was introduced in 1989 to replace an introductory course on BASIC programming- the emphasis being on programming concepts, good design and problem-solving applications, rather than the features of a particular language. Logo is the language which has been chosen to illustrate most concepts. The course attracts teachers from all levels of schooling and all disciplines, and has no computing pre-requisites, so an attempt has been made to find an approach which will, if possible, be new to all participants, and one which will be relevant both to those whose interests lie in enriching their own classroom teaching and to those aiming to become computer consultants or teachers of computing studies. This paper outlines the content of the strand, giving a justification for each aspect as well as an analysis of its difficulties and its successes.'

---

[2] Logo on a Connection Machine seems appropriate. The machine's designer, Danny Hillis, like Brian Silverman, grew up in the MIT Logo culture.

## The Logo Galah Session

In the 'chair' was Gary Stager, of ISTE SIGLogo, and present were Seymour Papert, Dan and Molly Watt, Brian Silverman, Tom Lough and a great number of others. Rather than pass round the usual sheet of paper, Gary used a laptop, only to find that people were slow typists.

There was general discussion for a while, with various people discussing how Logo was used in their state/country: Rumen Nikolov from Bulgaria, Wal Parker and David Williams of Victoria, Paul Dench from Western Australia, to name only a few. Two questions seemed to stand out: 'How do we make Logo work in schools, especially in senior secondary schools (although there seems to be more happening in Australia than elsewhere in that regard)?', and 'How do we find out what other people are doing?'

Mention was made of a paper by Brian Harvey, 'Symbolic Programming and the AP', which so clearly makes the point that Pascal is out of touch that even Pascalites agree. It may appear as an LCSI Memo.

One occasionally hears that 'Logo is dead.' The presence of so many, who, as someone put it, '... voted with their presence...' proves that it is not, and is in fact alive and well, and living in many places.

## Rainbow: Logo 1

Rob Pits: Logomate: A Facilitator of Logo Learning for Primary and Early Secondary Students

There were to have been three presentations at this session, but the people from Turkey had been unable to attend, and the presenter from Zimbabwe had been unable at the last minute to obtain foreign currency, so Rob had the floor to himself, which did make my task of chairing the session easier. Had the speaker from Zimbabwe been present there would certainly have been some interesting questions from the floor, for the paper was an account of the Roy Pea kind of research.

Logomate is an environment built on LCSI Logo II on the Apple. In a way, it is the old Instant procedures developed into a complete environment, in fact a set of environments from single key control through to procedure writing and editing. Students are spared the intricacies of editing and file management, and have constant feedback. As they work in immediate mode, the procedure is constructed, and the system allows backtracking for corrections.

Logomate is now available commercially through Dataworks, priced at $85.

## Rainbow: Logo 2

Garry Chapman: Creating LogoWriter Databases

Garry admits to being a newcomer to computing and LogoWriter, but has developed LogoWriter as a presentation vehicle, almost a mini-hypertext. The example he showed was about A. B. (Banjo) Paterson, describing his life and work. Pages were illustrated with a portrait of Paterson, produced with 11 redefined Turtle shapes (the method is to draw the picture on graph paper, then edit the `shapes` page).

Carolyn Dowling: Exploring the Unpredictable: The Computer as a Catalyst for Creative Thinking in the Classroom

Carolyn sees the computer as a tool for the empowering of the individual, as an intellectual partner, and that problem solving in the classroom should be open ended rather than product oriented. The problem in the classroom is often that certain ends be reached, for assessment if nothing else. The value of the computer is that it offers opportunities for undirected exploration, for messing about, and the encouragement of divergent and creative thinking.

Gary Stager: Developing Scientific Thought in a Logo-Based Environment

How best can students learn 'scientific method' and learn mathematical and problem solving skills? Not by passive rote fact learning from textbooks, but by actively pursuing knowledge in a dynamic environment. 'Science is something you *do* as well as *study*.'

Gary describes how Logo environments meet those requirements, by putting the students in control, solving real problems. At the end of the presentation Gary was surrounded by people buying a set of papers and notes he had prepared. Two of those appear on other pages.

## Rainbow: Logo 3

Peter Carter: Step By Step: LEGO Legged Locomotion

Wheels are useful, but not in mud or rocks, and machine walking is being researched worldwide. Biped walking requires more sensing and control than we have in LEGO/Logo, therefore we use static stability, with hexapod machines. The presentation was rushed, crammed into 20 minutes, but the audience laughed at the right places.

Arthur Tatnall: LEGO Logo for Big Kids: Control Technology and Senior Secondary Curricula

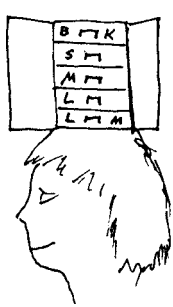
Most LEGO/Logo work has been with young children, but it has a real place at higher levels. Arthur sees value in LEGO/Logo in a range of subjects:

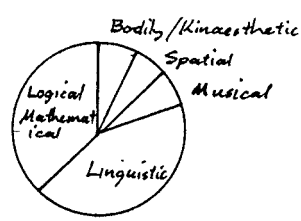
- Problem Solving: algorithmic vs heuristic
- Electronics: interfacing
- Computing: control programs
- Mechanics
- Cybernetics
- Science: sensory feedback, monitoring, control
- Mathematics: design and geometry of robot arms
- History: mechanisation, robots
- English: robots in fiction
- Social Science: robots and people
- Economics: the commercial imperative

Ole Moller: Bricks for Brains: A Hands- and Heads-on Approach to Scoence and Technology

'Why do not robots wear shoelaces?' Try explaining how to tie shoelaces in writing, and the answer becomes obvious. Yet schools try to teach most things that way, through the abstract rather than actively.

How can we understand bar code readers? By reading about them, or making a LEGO/Logo model? What about record players, CD players, IR remote control? Models again. Ole listed 5 intelligences: bodily/kinaesthetic, spatial, musical, linguistic, and logical/mathematical. On the left, a representation of a Danish child's brain, on the right, the usual time allotment in schools for the different intelligences.



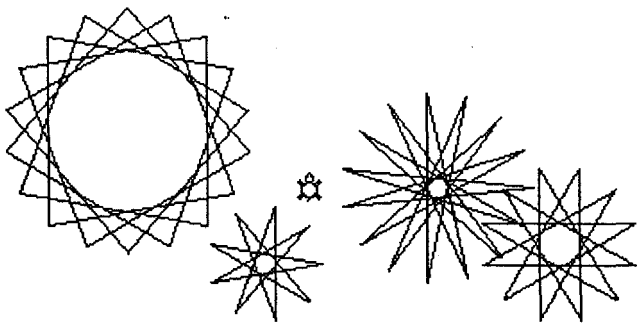


## Rainbow: Logo/Games

Linda Coyle and Jocelyn Rees: Computers Enable New Strategies and New Topics: An Example from Mathematics

The new topics are star polygons, formed by connecting equally spaced points in a circle. It can be done by hand, but the process is tedious and error prone. With Logo, three inputs to a procedure. Factoring, complements, simplest form, and some ideas from geometry form the mathematics.

Sample star polygons look like these...



... and the procedure, at least as I work it out:

```
TO StarPoly :totalPoints :points :size
FD :size
RT 360 / :totalpoints * :points
StarPoly :totalpoints :points :size
END
```

It could do with a stop rule. (How would you implement one?)

New strategies include a shift in the locus of control from teacher to student, immediate feedback, the possibility for exploration to greater depth, and the ease of formulating and testing hypotheses. Students enjoyed the 'neat shapes', liked the computer as a tool, and needed little teacher support.

Some questions that might be asked are 'Why are `StarPoly 9 4 50` and `StarPoly 9 5 50` alike?' and 'Why are `StarPoly 12 4 50` and `StarPoly 12 5 50` so different?'

Tom Lough: Logo in Secondary Education

Tom began with an overview of Logo as a language, then described how Logo could be used in mathematics and science for: vectors (with Turtle as the arrow on the end), linear motion, planar motion, gravitation, electrostatics, algebra, trigonometry, precalculus, and differential equations. There were some concerns: teacher training, materials and support, curricular responsibilities, hardware limitations, and perceptions of Logo (ie. as a 'toy'. In fact it's so powerful that it can be configured as a toy.) There were also some encourageing signs: teacher preparation was now improving, Logo use was becoming more widespread, students were using it from primary grades through to secondary, and there was a growing awareness of its power and value.

## Research: Twilight Panel

Working with Teacher Researchers in the Field of Computers in Education

Among the speakers were Dan and Molly Watt, who outlined their system of action research, which involved teachers attending regular meetings and closely observing two students for a year. Dan and Molly are excited with the potential of Logo, but see most research into it, in a country where research and evaluation are almost industries, as disappointing.

## Research: Problem Solving

Kwok-Wing Lai: Problem Solving in a LEGO/Logo Environment: Cognitive and Metacognitive Outcomes

This is perhaps the first paper on the outcomes of LEGO/Logo to be published, and it is most encouraging. Kwok-Wing, from New Zealand, was clearly excited by it, but frustrated at having to confine himself to 20 minutes.

In many ways the research was conventional, with pre- and post-intervention interviews and questionnaires, together with videotapes and logs. Unlike other studies, this one did not measure capacity to perform arithmetical operations or whatever, but concentrated on childrens' attitudes about what they were doing, and their awareness of how they were learning. A significant overall improvement in metacognitive awareness was noted, showing that children can be made aware of and develop their own metacognitive skills. They recognised that if they received no help, they 'had to learn ourselves.' They were also clearly pleased with their LEGO/Logo work.

Anne McDougall: Children, Recursion and Logo Programming...

It's difficult to explain recursion, Anne observed, without lots of waving of hands. She managed to avoid that in describing aspects of her research with two young children in showing that they could recognise many examples of recursion around them, and use the concept in their own Logo programming.

## Closing Keynote Speaker 2

Seymour Papert: The Perestroika of Epistemological Politics

Two themes seemed to pervade: the collapse of hierarchical, bureaucratic systems, and the conflict between instructionism and constructionism. The former seeks to improve schooling by better instruction, the latter by allowing children to construct, to build and to make, not just LEGO or Logo models, but the essential mental models.

How can we that that children are learning when they just appear to be playing? Playing is hard work, even Nintendo games are hard work, and the reward is even harder tasks. Children have been heard describing Logo as 'hard fun.' Research has so far concentrated on statistics: insert some Logo, leave everything else the same, and measure at the end. Culture is not like that, Logo must pervade. Logo must, if necessary, be a Trojan horse, carrying with it the seeds of change into the system.

The future computing teacher will not be a servant, but a philosophical leader.

# Resources

Came across several new, well, to me, items in the conference exhibition. A new book has arrived from MIT, and there's the promised review of another.

## LogoExpress

My first glimpse of LogoExpress was in someone's hotel room before the conference. There it was running on a laptop, trying, through a modem with a flat battery, to contact MIT in Boston, or LCSI in Montreal, via RMIT in Melbourne. No joy.

The official launch in Australia took place on the Thursday morning of the conference, on a launch trip on the harbour. The weather was decidedly dreary, with tall buildings disappearing into the overcast, and the boat felt very different from my usual craft (sea kayak), but with croissants and champers/orange juice, and sparklers, LogoExpress was formally released.

What is it? LogoExpress is a communications package, built on the LogoWriter environment. There's no Turtle, no graphics at all in fact, but in its place the code needed to drive modems, while all the usual LogoWriter editing remains. On the flip side, one puts procedures to auto-dial, log in, send and receive mail, etc. (they can also be tool pages), while the front of the page, in 80 columns, is the screen for sending and receiving. Telecom and OTC permitting, it's all very easy, and it was there to try on the Computelec stand in the exhibition.

LogoExpress is available in Apple and MS-DOS versions, and can be bought as a single copy, as a site licence, or as a district licence. The last package includes pages to configure the system as host. Schools in a district can then easily communicate between themselves, with the ability to transmit LogoWriter pages for the first time.

Also on the Computelec stand was the MS-DOS version of the *Phantom Fish Tank*, with the author to demonstrate.

## Archimedes Logo

It's three years since I'd used Logotron Logo, and I'd never tried an Archimedes, but I was immediately at home with the new Logotron Logo on the A 3000, although I did have to look up the colour numbers for the new display modes. All the capabilities of the old version are there, with lots more primitives, property lists etc., together with lots of memory and the sheer speed of the machine: less than 4 seconds for a 200 line PolySpi.

Tried one of my old flag programs (to appear in the forthcoming *Turtle Bunting*), and it worked straight off, and looked much better in the high resolution of mode 12 than it ever did in mode 5. I don't think it would persuade me to buy an Arc, but if I had one I'd certainly go for this Logo.

There was another Logo on the Acorn stand, RoboTron. It can be used as a conventional, if limited, Logo, but its default mode is an android. The body, arms, and legs can be resized, and of course can be animated so that the thing looks as if it's walking in three dimensions. A third mode is Jet, in which the Turtle 'flies' in three dimensions. Didn't get a chance to try it. Could be fun, but the manual admits that the Logo is limited; no lists, for instance, and the maths is integer only.

## Terrapin Logo for the Macintosh

Now that MacLogo has been discontinued, Terrapin Logo is the only version available for the Macintosh. It's by virtually the same team who wrote the original

Terrapin Logo for the Apple, and for the Commodore 64. However, the verbose IF..THEN..ELSE syntax has been replaced by LCSI type IF.. [] []. It's a very comprehensive version: multiple, redefinable Turtles, p-lists, etc, etc. The manual is a loose leaf affair which will need some reinforcement in the classroom, and contains tutorial and reference sections. Page numbering is more logical than the C-64 version. I didn't get a chance to try it, but Terrapin products in the past have been robust, and it seems to be well supported. Single copies are $145, a quite reasonable figure, from Dataflow and/or JPR Software. Mac equipped high schools will find it has all the power for computing science issues.

## The Valiant Roamer

The Valiant Turtle has been around for some years now; it's the polyhedral shaped one with infra-red control. The manufactuer has now developed Roamer, a self contained Turtle-like robot that can have all manner of decorations added, hermit crab like (it comes with "Face Shapes"), as well as other devices which can be controlled via an interface module. Valiant admit the influence of BigTrak, but have sought to make Roamer as non-sexist and non-aggressive looking as possible. The appearance is certainly bland, a rounded grey dome shape, with a keypad on the top.

Programming is similar to BigTrak, CompuRobot and others. Program capacity is 60 instructions, including REPEAT and PROCEDURE facilities, and sensor devices and dc or stepper motors can be connected. The tone generator has a range of 3 octaves.

Roamer appears solidly made, and it uses 6 volt lantern batteries (including rechargables) rather than a pile of those other little things. There's a range of interfaces and other gear, as well as a video, *Objects to Think With*, available.

## The Turing Omnibus

A. K. Dewdney will be known to many as the author of the Computer Recreations column in *Scientific American*, and also an interesting sf piece, *The Planiverse*. *The Turing Omnibus* is a collection of 61 essays on a range of computing science topics: AI, simulation, machine vision, cryptography and the like, with numerous references to Turing machines and like automata. The book's name comes from the idea of a touring omnibus, a red double-decker, as seen by Dewdney during a sabbatical in Oxford, and it was written, like most issues of *POALL*, bit by bit over a period of time.

In the first chapter, Dewdney discusses algorithms, and, like other authors, uses an example from the kitchen, before moving on to wallpaper (p 3):

```
'       WALLPAPER

1. input corna, cornb
2. input side
3. for i ← 1 to 100
  1. for j ← 1 to 100
    x ← corna + i * side/100
    y ← cornb + j * side/100
    c ← int(x^2 + y^2)
    if c even
     then plot (i, f) '
```

That translates into Logo (Dewdney uses Pascal) as:

```
TO Wallpaper :cornerA :cornerB :size
CS HT
MainLoop 1
END
```

```
TO MainLoop :l
IF :i > 100 [STOP]
Plot :i 1
MainLoop :i + 1
END

TO Plot :i :j
IF :j > 100 [STOP]
MAKE "x :cornerA + :i * :size / 100
MAKE "y :cornerB + :j * :size / 100
MAKE "c INT ( :x * :x + :y * :y )
IF Even? :c [DOT SE :i :j]
Plot :i :j + 1
END

TO Even? :no
OP IFELSE 0 = REMAINDER :no 2 [TRUE] [FALSE]
END
```

Be warned, it's slow.

Every chapter has a set of problems, and a set of references. *The Turing Omnibus* is not for the school student, but for tertiary students and others looking for a readable introduction to the foundations of computing science, it should be on the list.

## Approaching Precalculus Mathematics Discretely

Philip Lewis teaches mathematics and computer science at a high school in Massachusetts, and *Approaching Precalculus Mathematics Discretely* is the latest in the Exploring with Logo series from MIT. Discretely? Lewis explains (p 201):

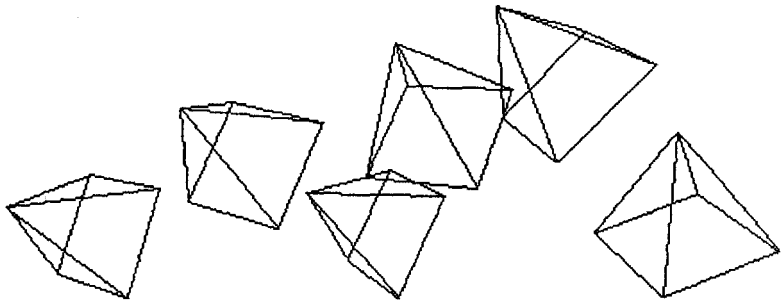
> 'The mathematics [in this book] is a discrete mathematics. This means that it deals with the finite: finite sets of points determined by finite numbers of calculations together with the algorithms for performing them.'

Lewis uses Logo as a mathematical language, showing how it can be used to explore and apply much of the material that is normally dealt with abstractly in high school mathematics. The book is an excellent model of integration of Logo into the curriculum.

In the first chapters, Lewis introduces Logo as a means of expressing mathematical functions. The Logo is straightforward, and well explained. Lewis moves on to vector operations in the plane, and then to linear transformations and matrix operations to develop a system to plot geometric objects on the screen.

In Chapter 5 there is as good an explanation of recursion and mathematical induction as you will find anywhere. Lewis is conventional in using factorials and the Tower of Hanoi in his explanations, but approaches from the mathematical rather than the programming aspect.

The two dimensional transformations are now extended to three dimensions, with a program to draw pyramids:



Lewis used Object Logo on the Macintosh to develop final versions of the programs, but they should work with other dialects since he makes no use of objects or other special features. I also used Object Logo after MacLogo gave up[3], but I found one procedure duplicated, and others had parentheses that Object Logo complained about. In Object Logo the program works well, but Lewis warns readers that it can be slow on other systems.

The remaining chapters contain a general graphing system that is used to explore various functions and the idea of a limit. From there, Lewis moves on to derivatives and integrals of polynomials, the precursors of calculus.

*Approaching Precalculus Mathematics Discretely* is both good mathematics and good Logo, and will be an inspiration to all mathematical Logophiles.

## Tortoises of Australia

This little book is in print again. It's a well illustrated (92 colour photographs) description of the Australian species, more than twenty, together with advice for feeding and care for those who keep specimens. Tortoises[4] (and turtles) are threatened in many areas as wetlands are drained, streams polluted etc. Several times while canoeing Murray backwaters I've come across the remains of tortoises that have been trapped in fish nets and then simply thrown up on the bank, or, in one case, crushed. Cann's book is informative, and encourages care and conservation. My old copy is paperback, the new printing is hardcover and priced at $17.95.

## The Books:

Dewdney, A. K. *The Turing Omnibus* Computer Science Press (W. H. Freeman), 1989

Lewis, P. *Approaching Precalculus Mathematics Discretely* MIT Press, 1990

Cann, J. *Tortoises of Australia* Angus and Robertson

[3] The procedures filled MacLogo's workspace, and refused to run. There seems no way of telling MacLogo that it's running on a 4 Mb machine, and not the 512 K that it was originally written for.

[4] Australian tortoises are all aquatic, but the name 'turtle' is usually reserved for marine species.

# LogoWriter in Latin America

From *LCSI Logo Link*

Michael Tempel

Project Genesis began in Costa Rica a little over two years ago. By the end of the current school year (Which began in March 1990), labs of IBM computers running LogoWriter will be in 210 schools serving half hte elementary school children of the country.

This past January, the first steps were taken to build a similar project in Venezuela. A three-week workshop was sponsored by the IBM Latin American Education Research Centre. A team of nine instructors, including myself, from Venezuela, Costa Rica and the United States worked with the entire teaching staff of one school, Escuela Simon Bolivar. The school, with 400 students in grades kindergarten through nine, is operated by LagoVen, one of Venezuela's oil companies, for the children of some of its thousands of employees at the huge Amuay refinery on the remote Paraguana Peninsula.

LagoVen also provides housing and recreational facilities in a community that is vaguely reminiscent of a typical North American suburb. (It was built by Americans prior to the nationalisation of Venezuela's oil industry in 1975.) Of course, it is diffucult to maintain a green lawn in the face of a hot desert climate, six-inch long grasshoppers, and herds of wild goats and donkeys that graze freely about.

By the time I joined the workshop for the final week, the teachers had gotten quite far with some very elaborate projects. A wide variety of curriculum themes found expression in illustrated and animated reports and tutorials. Teachers worked in groups according to grade level or subject. There were projects on history, geography, math, music, physics, and physical education.

In many ways, this workshop was similar to others that I have been involved in in the United States. The room was set up with 30 computers, one for each participant. Twenty of these were to remain in the school permanently. The others were provided by IBM for the duration of the workshop. Most of each 7 am to 5 pm day was spent in hands-on project development with the instructors providing assistance as needed. There were also several small group sessions each day devoted to topics such as mathematics, using LogoWriter with very young children, Logo grammar, and LogoWriter language activities. Teachers attended when the subjects were of interest to them. At the end of each day, there was a large group discusssion, usually having to do with how LogoWriter would be incorporated into the life of the school.

A noteworthy aspect to this workshop was that *all* the teachers were involved. This was not a workshop for computer teachers. There are no computer teachers at Escuela Simn Bolivar, just teachers using computers along with many other materials and tools. This is surely the direction in which things are headed in North America as well, but in this respect, Latin America may be more advanced.

As the long workshop drew to a close, the teachers prepared to show off their projects to each other and to visiting oil company executives. The evening before this final session, a half dozen cars pulled up outside the company housing where the instructors were living. A couple of dozen teachers piled out, and with guitars and quatros they serenaded us. I've had posiitive participant evaluations at workshops, but this set a new standard. We were all up rather late, but still got up for the show next day.

The music teacher had programmed a local folk tune complete with animated dancers. The science teacher had created a tutorial about the physics of light. An elementary teacher had done a report on various forms of energy. The physical education teachers did animations of the various sports that the school children play. And there were many others.

The intention of these projects was not always clear. Were the teachers modeling the kinds of work they expected their students to do? Were they using LogoWriter as an authoring system to create software for use by their students? The answers to these questions will emerge as they use LogoWriter with their students over the coming months and years.

A few weeks later, this first group of teachers began to use LogoWriter with their students; half an hour per week for the kindergarten children, up to three hours per week for the ninth graders. At the same time, there was another three-week workshop underway, a hundred miles to the south, for the teachers of five more oil company schools.

How far will this go? Does the experience in a handful of very special schools provide a reasonable model for the rest of Venezuela's 18,000 schools? Will there be the same kind of national commitment to computer use as there is in Costa Rica? We don't know, but the first steps are being taken.

# What LogoWriter, LEGO TC Logo and The Phantom Fishtank[5] Contribute to Scientific Thinking

Gary Stager

1 Out of simple beginnings complex structures may emerge.

2 Each environment is child-centred. Students are motivated to think and solve problems because the problems were generated by themselves in the process of creating a project.

3 Debugging is encouraged and strongly parallels important aspects of the scientific method. Experience in debugging encourages a healthier attitude towards learning and provides students with strategies for approaching any problem.

4 Each computing environment has 'no threshold and no ceiling.' Students are free to explore, create, and learn at their own level. Students are challenged by their imaginations. They will never reach the limits of the software.

5 The three environments support the philosophy of constructivism, embraced by Piaget and Papert. Students learn new concepts by reflecting on previous experience and constructing their own intellectual models.

6 The materials discourage passivity. Learning becomes an active and exciting life-long process and students feel free to roll up their sleeves and 'mess about' with science.

7 The three computing environments are Logo-based so that the 'young scientist' need only learn one environment, set of syntax, and rules of behaviour. Fluency with the materials allow for heightened creativity and intellectual discovery.

---

[5] ie. Logo environments in general, not just LCSI products.

# Fibonacci Revisited

Leonardo of Pisa, Filius Bonaccio, Fibonacci for short, was an Italian mathematician who discovered an interesting sequence in about 1202, a sequence that turns up in botany and other places. The writer of an article in a recent *AppleSauce*, the magazine of the SA Apple Users Club, presented two Fibonacci generators whch he admitted were not elegant. They were BASICly flawed, because the Fibonacci numbers are best defined recursively...

$$Fib_0 = 0$$
$$Fib_1 = 1$$
$$Fib_2 = 1$$
$$Fib_n = Fib_{n-1} + Fib_{n-2} \text{ (where } n > 3)$$

...which translates easily into Logo as:

```
TO Fib1 :number
IF 0 = :number [OUTPUT 0]
IF 1 = :number [OUTPUT 1]
OUTPUT ( Fib1 :number - 1 ) + ( Fib1 :number - 2 )
END
```

That's certainly valid, but it's also flawed, because it's wasteful of time and memory. To calculate the fifth number in the sequence, the third must be calculated twice, the second three times, and the problem becomes worse as the number increases. (`TRACE "Fib1` and you'll see it all.) It's an example of 'tree recursion.' Nothing wrong with tree recursion in its place, but in this case it's simply inefficient.

An iterative process is not so wasteful. We can use a pair of numbers, $a$ and $b$, initialised to 1 and 0, and apply the processes $a \leftarrow a + b$ and $b \leftarrow a$. Again, the translation is straightforward, although in Logo iteration means tail recursion:

```
TO Fib2 :number
OUTPUT Fib2Aux 1 0 :number
END

TO Fib2Aux :a :b :count
IF 0 = :count [OUTPUT :b]
OUTPUT Fib2Aux :a + :b :a :count - 1
END
```

A third way is to generate a list of the numbers by adding the sum of the first two numbers on the list to the beginning of the list:

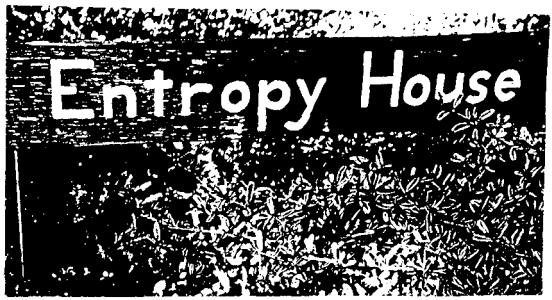
```
TO FibList :number
IF :number < 3 [OUTPUT [1 1 0]]
OUTPUT FibListAux FibList :number - 1
END

TO FibListAux :list
OUTPUT FPUT ( FIRST :list ) + ( FIRST BF :list ) :list
END
```

Our third `Fib` procedure simply outputs the first item on the list:

```
TO Fib3 :number
OUTPUT FIRST FibList :number
END
```

Much more elegant than four letter words. The list version I've snarfed from Brian Harvey's Volume 1, the iterative from Abelson, Sussman and Sussman.

# Computing at





Thought I was joking about the name of the place, weren't you. Needs repainting, and the weeds pulled from among the groundcover.

MacLogo (ie. LCSI's Logo for the Macintosh) has been hard to come by for some time, and I've had several people call me for information about it. The word is now out: MacLogo is no more. In its place LCSI will be releasing a Macintosh version of LogoWriter in the near future. It will have all the features common to other LogoWriter versions, but will make use of the Mac windows, mouse, etc. Mac equipped schools, especially primary schools, start saving now for your site licences.

There was some discussion at WCCE90 of the deficiencies and omissions from LogoWriter: THROW, CATCH, TEXT, DEFINE, TRACE, WINDOW etc. Some of them are not really lost at primary level, but for some of the work with senior secondary students they can be useful. As it is, self replicating procedures are rather difficult. DEFINE can be written, but TEXT would require some very neat work with SELECT. A project for someone?) TRACE can be very useful when dealing with recursive processes. If there are things you want to see in future versions of LogoWriter, give the people at LCSI a call.

One person who's already made a suggestion about improving LogoExpress is Brian Silverman's daughter, who wasn't satisfied with fixed Tabs and suggested Brian go to the office then and there and fix it.

Another Logo to meet its demise is Object Logo, also for the Macintosh. In this case the blame lies with Apple, who bought the Coral company that made Object Logo, and a highly regarded LISP. Seems that Apple had its eye only on the OOPS environment for use in other things, and simply abandoned the Logo. Do you want it back? Write to Apple, to John Scully in fact, or to Alan Kay, with the message 'Let our Turtle go', and suggesting that it be made available through the Apple Programmers and Developers Association. Object Logo is a powerful system, an excellent tool for serious programming, and deserves not to be simply forgotten. In the meantime, Terrapin Logo is the only Logo available for the Mac.

Seymour Papert visited the Powerhouse Museum with a group of friends, and was intrigued by the set of old railway switching levers, so intrigued it seems, that he climbed on to them to see how they worked. When a small child approached, he was firmly told by his mother 'Don't touch that thing!' There's a lesson in learning styles somewhere.

Dan and Molly Watt headed home with T-shirts bearing Aboriginal designs. Turtles of course.

Somebody rang the other day, asking about printing graphics from PC Logo. It's not one I know, so I suggested she read the manual. PC Logo is a Logo with all the works, but for a primary school, whence came the call, LogoWriter every time.