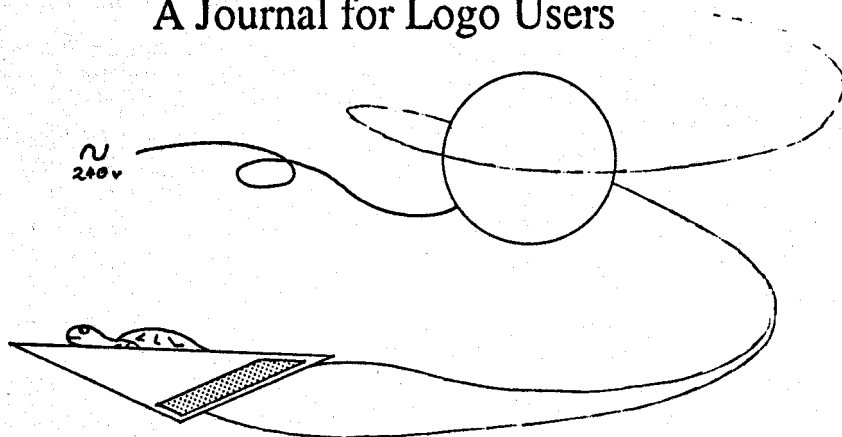


POALL

A Journal for Logo Users



ACEC'91, Extending LogoWriter

Volume 6 Number 2, October 1991

Top marks to CEGQ for the success of ACEC'91. You can read a little about the Logo part of the conference in this issue.

The main theme of this edition is extending LogoWriter. Earlier versions of Logo could be extended with buried packages of utility procedures to draw circles, drive hardware Turtles and so on. The tradition continues with LogoWriter, which makes some extensions easier, through the use of the screen as data, others more difficult, because primitives cannot be redefined.

The article by Bill Kerr I've hijacked from material for a future edition of the CEGSA *Quarterly*. The Constructionism/Instructionism debate is an important current issue, and the ideas deserve to be widely spread.

The next edition may be ready as soon as next month. There's a major ACEC'91 piece ready to go; all it needs now is your contribution.

Peter

Peter J Carter, Editor

POTS

TC Logo Emulator	2	Resources	13
Hypertext Tools	7	Missing out on the Mindstorms	15
ACEC'91	10	Computing at Entropy House	20

Extending LogoWriter 1, a TC Logo Emulator

As the power and memory of machines increases, so do the features of LogoWriter: Features that were once in separate versions, like the primitives in LogoExpress, are incorporated to make LogoEnsemble. Given time, the LEGO primitives will be integrated too. In the meantime this set of procedures, used as a tools page, will allow some control from the Apple version of LogoWriter¹.

There are some limitations: SETPOWER, FLASH, TIMER, TIMER#, RESETT and RESETT# are not implemented, and COUNTER works only for port 6². (The port 7 counter could be used as a random number generator.) If you turn a port off, you will have to TALKTO it again before you can turn it on once more.

Because STARTUP doesn't work with tools pages, you must call LegoInit to initialise the system. You could call it from the STARTUP of your page. The address of the interface is kept as a global variable. As shown here, it's for slot 2. For slot 4 it is 49344 (the formula is :portAddress := 49280 + slotNumber * 16).

```
LEGO TC Logo emulator
P J Carter, 1987..91
```

```
TO LegoInit
Globals
.DEPOSIT :portAddress + 3 1
.DEPOSIT :portAddress + 2 63
.DEPOSIT :portAddress + 1 0
.DEPOSIT :portAddress + 11 32
.DEPOSIT :portAddress 0
AO
SHOW "|Interface initialised.
END
```

```
TO Globals
MAKE "current [1]
MAKE "inBit 6
MAKE "0 0
MAKE "1 0
MAKE "2 0
MAKE "3 0
MAKE "4 0
MAKE "5 0
MAKE "ports [A B C]
MAKE "A "O
MAKE "B "O
MAKE "C "O
MAKE "oldA "F
MAKE "oldB "F
MAKE "oldC "F
MAKE "portAddress 49312
END
```

To turn things on or off, ports must be addressed with TalkTo, eg TalkTo [a 3 5] or TalkTo "B ...

```
TO TalkTo :bits
MAKE "current Check Caps :bits
END
```

¹ If you're a PC expert, please revise these procedures for the MS-DOS world and send us a copy for publication.

² Someone who knows a lot more about the 6522 than I do might like to try writing these.

```
TO TTo :bits
TalkTo :bits
END
```

```
TO Check :bits
IF EMPTY? :bits [OP []]
IF "A = FIRST :bits
  [MAKE "A :oldA
  IFELSE :A = "F
    [OP SE 0 Check BF :bits]
    [OP SE 1 Check BF :bits]]
IF "B = FIRST :bits
  [MAKE "B :oldB
  IFELSE :B = "F
    [OP SE 2 Check BF :bits]
    [OP SE 3 Check BF :bits]]
IF "C = FIRST :bits
  [MAKE "C :oldC
  IFELSE :C = "F
    [OP SE 4 Check BF :bits]
    [OP SE 5 Check BF :bits]]
OP SE FIRST :bits Check BF :bits
END
```

```
TO Caps :bits
IF EMPTY? :bits [OP []]
OP SE UpperCase FIRST :bits Caps BF :bits
END
```

```
TO UpperCase :char
OP IFELSE (ASCII :char) > 92
  [CHAR (ASCII :char) - 32]
  [:char]
END
```

... and then turned On or Off, or OnFor a specified time:

```
TO On
Set :current 1
Out
END
```

```
TO OnFor :time
On
WAIT :time
Off
END
```

```
TO Off
IF OR ( AND :0 = 1 :A = "F )
  ( AND :1 = 1 :A = "R )
  [MAKE "oldA :A MAKE "A "O]
IF OR ( AND :2 = 1 :B = "F )
  ( AND :3 = 1 :B = "R )
  [MAKE "oldB :B MAKE "B "O]
IF OR ( AND :4 = 1 :C = "F )
  ( AND :4 = 1 :C = "R )
  [MAKE "oldC :C MAKE "C "O]
Set :current 0
Out
END
```

```

TO Set :current :value
IF EMPTY? :current [STOP]
MAKE FIRST :current :value
Set BF :current :value
END

```

```

TO Out
.DEPOSIT :portAddress Convert 0 1
END

```

```

TO Convert :bit :placeValue
IF :bit > 5 [OP 0]
OP ( THING :bit ) * :placeValue + Convert :bit + 1 :placeValue
* 2
END

```

To switch everything off, AO for all off (You might even like WHEN "Q [AO] in LegoInit):

```

TO AO
.DEPOSIT :portAddress 0
Set [0 1 2 3 4 5] 0
TalkTo "
MAKE "A "O MAKE "B "O MAKE "C "O
END

```

To make things start in a particular direction...

```

TO SetEven
IF NOT 0 = :A
[IF :A = "R [Swap "A "F 0 1]]
IF NOT 0 = :B
[IF :B = "R [Swap "B "F 2 3]]
IF NOT 0 = :C
[IF :C = "R [Swap "C "F 4 5]]
END

```

```

TO SetOdd
IF NOT 0 = :A
[IF :A = "F [Swap "A "R 1 0]]
IF NOT 0 = :B
[IF :B = "F [Swap "B "R 3 2]]
IF NOT 0 = :C
[IF :C = "F [Swap "C "R 5 4]]
END

```

..or reverse while running:

```

TO RD
Reverse :ports
END

```

```

TO Reverse :ports
IFELSE :A = "F
[Swap "A "R 1 0]
[IF :A = "R [Swap "A "F 0 1]]
IFELSE :B = "F
[Swap "B "R 3 2]
[IF :B = "R [Swap "B "F 2 3]]
IFELSE :C = "F
[Swap "C "R 5 4]
[IF :C = "R [Swap "C "F 4 5]]
Set :current 1
Out
END

```

```

TO Swap :port :newDir :newBit :oldBit
MAKE :port :newDir
MAKE :oldBit 0
MAKE "current Exchange :newBit :oldBit :current
END

TO Exchange :new :old :object
IF EMPTY? :object [OP "]
IF :old = FIRST :object [OP SE :new BF :object]
OP SE FIRST :object Exchange :new :old BF :object
END

```

Sensors are 'woken up' with ListenTo, either 6 or 7. If you just want to sense rather than count, 7 is preferable.

```

TO ListenTo :bit
MAKE "inBit :bit
END

```

```

TO LTo :bit
ListenTo :bit
END

```

Sensor? reports either a TRUE or FALSE

```

TO Sensor?
OP IFELSE 7 = :inBit [Bit??] [Bit6?]
END

TO Bit6?
MAKE "bits .EXAMINE :portAddress
IF :bits > 127
  [OP IFELSE ( :bits - 128 ) > 63
    ["TRUE] ["FALSE]]
OP IFELSE :bits > 63 ["TRUE] ["FALSE]
END

TO Bit??
OP IFELSE
  (.EXAMINE :portAddress ) > 127
  ["TRUE] ["FALSE]
END

```

Counter reports a list, eg [24 347]. Use only the first number (port 6), and reset the counter with ResetC.

```

TO Counter
OP SE 65535 -
  ((.EXAMINE :portAddress + 8) +
  256 * (.EXAMINE :portAddress + 9))
  (.EXAMINE :portAddress + 4) +
  (.EXAMINE :portAddress + 5)
END

TO ResetC
.DEPOSIT :portAddress + 8 255
.DEPOSIT :portAddress + 9 255
END

```

With this next procedure you can write expressions like WaitUntil [(FIRST Counter) > 36]

```

TO WaitUntil :condition
IF RUN :condition [STOP]
WaitUntil :condition
END

```

Should you buy LogoWriter or TC Logo? Yes. Don't expect one to do the work of the other. LogoWriter has all the graphics and can be extended, as here, to drive external devices. TC Logo does the best job of driving the LEGO hardware, but is missing some graphics primitives. To get the best of both worlds you need both.



The 6522 VIA

The 6522 VIA (Versatile Interface Adaptor) is a flexible interface chip with two input/output ports, two 16 bit timers and serial to parallel/parallel to serial shift register, 16 registers in all.

Each bit of the two 8 bit I/O ports can be used for either input or output, depending on the settings of the data direction registers. On the LEGO interface, like the BBC Micro User Port, I/O register B is the one used, and the data direction register is set with the bit pattern 00111111 (63₁₀), 0 for input, 1 for output.

The timer used by these procedures counts down from 11111111111111₂ (65535₁₀)

A LogoWriter Timer

These procedures come from Adam Smith at MLC in Melbourne, and are for the PC version of LogoWriter. Note the method of adding leading zeros:

```

to reset.timer
.deposit 1132 0
end

to timer
make "ns int((.examine 1132) / 18.2042)
make "m int(:ns / 60)
make "s :ns - :m * 60
if (count :s) = 1 [make "s word "0 :s]
if (count :m) = 1 [make "m word "0 :m]
op (se :m " : :s)
end

```



The Bug stops here

- 1 The 'Queensland state school' mentioned on p 15 of our last issue should have been Coombabah. Apologies to Karen Hallett who pointed it out, *et al*.
- 2 Trouble with LogoExpress autodialling? (When is Hayes compatible not Hayes compatible?) To ATDP (pulse) or ATDT (tone) you need to add B9 for 2400 bps or B7 for 1200, like this: ATB9DP or ATB7DT, or whatever combination you need.



Extending LogoWriter 2, Hypertext Tools

The Mac version of Gary Stager's LogoWriter hypertext tools. The Apple/PC version was published in *Sunrise Notes*, Vol 1 No 4.

**Bug-Fixes for Gary S. Stager's LogoWriter
HyperText Program Designed for Mac
LogoWriter.**

Put the following procedures on a page named,
HYPERTEXTTOOLS.

Hit shift-option-Z while on a word to link to a page

Hit shift-option-Q to go to the first page of your hyper-
text/database

Hit shift-option-P to go to the previous page

Hit shift-option-X for a list of these keys

GP "HYPETOOL to begin a new hypertext document

When you would like to work on an already started hy-
pertext document

YOU MUST always begin working on your story/
hypertext/database from the first page you created.

(C) 1990-91 Gary S. Stager

```
to cursunder
cf select cb
OP selected
end
```

```
to findspace.b
if (or (textpos = 0) (cursunder =
char 32) (cursunder = char 13) punc-
tuation?) (make "word.start textpos +
1 stop)
unselect cb
findspace.b
end
```

```
to findspace.f
if (or (textpos = textlen) punctua-
tion? (cursunder = char 32) (cursun-
der = char 13)) (make "word.end text-
pos stop)
unselect cf
findspace.f
end
```

```
to punctuation?
op member? cursunder [ . , ? ! " ]
end
```

```
to getword
findspace.b
cf
findspace.f
unselect
settextpos :word.start
```

```
select
settextpos :word.end
cut
end
```

```
to allcaps
cut
insert cap clipboard
end
```

```
to cap :word
if empty? :word (op " )
if (ascii first :word) > 90 (op word
(char (ascii first :word) - 32) cap
bf :word)
op word first :word cap bf :word
end
```

```
to link
getword
allcaps
make "previous :current
if member? first8 clipboard pagelist
[gp first8 clipboard make "current
clipboard stop]
newlink
end
```

```
to newlink
savepage cp
np first8 clipboard
make "current clipboard
end
```

```
to ready?
CC
type [Hit a key when you are done
reading the instructions]
ignore readchar
cc
end
```

```
TO ADD.STARTUP
IF FRONT? [FLIP]
PR [TO STARTUP]
PR SE [MAKE "FIRSTPAGE] word ""
:CURRENT.
PR [GETTOOLS "HYPERTEXTTOOLS]
pr [END]
pr []
FLIP
END
```

```
to new
cc
getpagename
end
```

```

to getpagename
type (se char 13 [What would you like
to name this page?] char 32)
make "current first readlistcc
if member? :current pagelist [type se
char 13 [Already used... please
choose another name] getpagename
stop]
np first8 :current
make "firstpage :current
cc
ADD.STARTUP
end

```

```

to startup
if not page = "hypertexttools [stop]
ready?
newpage
gettools "hypertexttools
setevents
new
end

to ?
cc
type se [shift-option-Z = Link shift-
option-P = Previous Page shift-
option-Q = First Page shift-option-X
= Help] char 13
end

```

```

to previous
make "last :current
gp :previous
make "current :previous
make "previous :last
end

```

```

to setevents
when "p [previous]
when "z [link]
when "q [gp :firstpage]
when "x [?]
end

```

```

to procedure? :thing
if front? [flip]
top
search (word "to char 32 :thing)
flip
op found? :thing
end

```

The first8 procedure was written for DOS versions of this program and actually limits the name of the page to 15 characters in the Macintosh version. I was just too lazy to change the procedure name.

```
to first8 :word
```

```

if (count :word) < 15 [op :word]
op first8 bl :word
end

```

```

to ignore :thing
end

```

A Talking Alphabet

An example of a Mac LogoWriter Hypertext Program in which you can teach the computer to say the name of a letter when clicked on by the mouse.

If the computer does not recognize the word, you can teach it and it will remember.

This program requires sound import capability on your Macintosh.

```

to go
pu
setpos mousepos
if button? [check]
go
end

```

```

to check
if colorunder = 4 [play "blue stop]
if colorunder = 5 [play "red stop]
if member? charunder soundlist [play
charunder stop]
learn
end

```

```

to learn
if charunder = 32 [stop]
play "question
record charunder 2
end

```

Type GO to begin the program and click on letters on the page.

You may also wish to click on colored "buttons" to perform a desired effect.

A Talking Wordprocessor

This program will read the text on the LogoWriter page aloud and will ask you to teach it any word it doesn't recognize.

Be sure to record a sound, called QUESTION, prior to running this program. The sound will be stored with the page.

The QUESTION sound should say, "Please read me the name of the word to the left of the cursor."

You also need to teach LogoWriter how to pronounce each letter in the alphabet by recording each letter and then naming the sound with the name of the letter. You may use the LEARN.LETTERS (A B C D E F G...) procedure to record the alphabet. Be sure to wait for the second beep after you have recorded "A" in order to record all of the letters properly.

Type READ to begin reading the text.

```
to read
top
make "theword "
top go
end
```

```
to go
if textpos = textlen [say :theword
stop]
select
cf
ifelse selected = char 32 [unselect
say :theword make "theword "] [play
selected make "theword word :theword
selected unselect ]
go
end
```

```
to say :theword
if member? :theword soundlist [play
:theword stop]
play "question
record :theword 4
end
```

---Extra Junk---

```
to learn.letters :list
if empty? :list [stop]
cc show first :list
record first :list 2
wait 3
learn.letters bf :list
end
```

Jukebox [list] will play all of the sounds in the list. Try typing JUKEBOX SOUNDLIST.

```
to jukebox :list
if empty? :list [stop]
cc show first :list
play first :list
jukebox bf :list
end
```

```
to erase :list
```

```
if empty? :list [stop]
ersound first :list
erase bf :list
end
```

©1991 Gary S. Stager
Computer Education Consultant
N.A.M.E. - Fallon Center
51 Clifford Drive
Wayne, NJ 07470 USA
voice: (201) 633-3121
fax: (201) 628-8837

LogoExpress: GARYS
BITNET: K0331@Applelink.apple.com
Applelink: K0331
Compuserve: 73306.2446



ACEC'91

CEGQ hosted the Ninth Australian Computers in Education Conference at Bond University, 23..25th September. The venue was ideal (if unfinished in places), and as for the weather, like stepping out of midwinter into midsummer, complete with bushfires in the background.

With both Fred D'Ignazio and Jan Hawkins as keynote speakers, multimedia was a key theme. Fred's presentation especially was a 'bells and whistles' demonstration of aspects of the multimedia systems, orchestrated by LinkWay, the latest MS-DOS HyperCard clone.

Before the conference itself were some LEGO Logo workshop sessions, which I missed because of the ACCE Board meeting, and there were other sessions and the Coombabah team in attendance throughout.

What follows is a brief précis of each of the Logo or Logo related sessions, based largely on the published abstracts. A P indicates that the full paper appears in the proceedings, which are available from CEGQ.

Anne McDougall: Programming as teaching the computer P
(A case for the revival of programming)

'This paper examines the metaphor of programming as teaching the computer noting the importance of programming languages for describing processes and other ideas not easily expressed in everyday language, and relates the precision of specification and description needed in programming to the skills of communication and explanation needed for teaching, arguing that the exercise of such expression for communication increases an individual's understanding of the conceptual domain being considered. Based on this a case is presented for providing opportunities for students to develop sufficient skills in programming to be able to use programming to be able to use programming to support their learning and problem solving in other curriculum areas.'

Several of the questions at the end revealed that the questioners misunderstand the reason for Logo, as a notation for the process of problem solving. One person even went as far as to suggest that students should be using CASE tools!

Liddy Nevile: What would Piaget say? P

'Only awareness is educable' Gattegno
'Logic should be on tap, not on top' Papert

'In this paper there is some discussion of the ideas of epistemological pluralism and imperative knowledge. It is argued that it is time for a reconsideration of how students are encouraged to learn and in what form they should expect to find and represent knowledge. It is suggested that it is time for teachers to take advantage of the programming facility of computers to provide for the high proportion of bricoleurs among the students.'

The presentation was based on an idea from Holt, that letting some piece of knowledge, or skill, go by was like leaving something at McDonald's and having to go back for it.

David Mitchell: Logo without houses

'The aim of this presentation was to provide ideas for the use of Logo and LogoWriter in mathematics which do not rely on the teacher possessing advanced expertise with Logo or having access to a large amount of commercially prepared

material. The majority of the activities do not require any programming skills and a minimum knowledge of Logo commands.'

Dave demonstrated the use of LogoWriter for exploring mathematical expressions like³:

```
PR (4 * 100) + (8 * 10) + (3 * 1) + (0 * 0.1) + (5 * 0.01) + (6
* 0.001)
PR 23.781 / 0.3 = 237.81 / 3
PR 2 * 3 * 2 * 2 = 24
```

There followed some demonstrations of student written programs for converting fractions to decimals, interest and other percentage calculations, dice rolling, UTC⁴ clock etc.

Jenny Betts:

The creation of databases using LogoWriter

'Permitting the children to initiate their own accumulation of knowledge was the original intention of this project in Social Studies and so the idea of these students creating individualised databases to suit their needs was introduced. Even though their programming methods may have appeared crude, together the children and the presenter made some startling discoveries. Not only did this project allow for the development of many programming skills applicable to all curriculum areas, but it fostered the accumulation of an extraordinary amount of knowledge of the processes involved in the gathering, synthesising and presentation of information.'

Jenny professed to know no more Logo than HideTurtle when the project began, although that had clearly changed by the end. The students' skills in 'gathering, synthesising and presentation of information' had also developed considerably.

The presentation included demonstrations of a number of student programs. Among the problems they needed to solve were those of allowing the user control of scrolling text, and the choice, through menus, of successive pages. The eventual method of scrolling was to use the arrow keys to control CursorUp and CursorDown.

The biggest problem was searching for words in several pages of text. The solution was to use LoadText and Search.

Perhaps the most impressive piece of programming was a currency converter. The screen showed a map from PC Globe. As an arrow (redefined Turtle) was moved about the screen, the name of the country it crossed was printed in the command centre. The secret? Beneath the map was a screenful of text, in colour 0, invisible. CharUnder did the rest. No lists or coordinates, the screen as data. Neat, and devised unaided by Warwick Mitchell, a Year 7 student⁵. More on this in the next edition.

Karen Hallett

Unravelling Rules and Exceptions

P

'How is our language structured? Are there simple sets of rules which determine how words and sentences can be created, modified and built into larger units? While the complexity and eccentricity of English may be obvious to adult language users, these questions represent a real challenge to a student who is beginning

³ The sample expressions were stored on the flip side of a page and were copied to the command centre for scrolling to in turn.

⁴ Universal Coordinated Time, what used to be known as GMT. Why are the initials in the wrong order? Ask the French.

⁵ Why is it that 12 year olds can run rings, figuratively speaking, around Year 12 students who seem hardly able to program themselves out of a wet polygon? Perhaps incentive has something to do with it. Perhaps curriculum has something to do with it too.

to use the language extensively. Perhaps just as challenging is the organisation of instruction so that students can untangle, and make explicit, the structure of the language in an exciting, constructive and rewarding manner. How can a student 'play' and solve problems with language apart from reading, writing, listening and speaking?

This paper will look at some ideas that grew from one small part of a professional development experience recently held at the Queensland Sunrise Centre, Coombabah State School. A number of perspectives will be offered, including those of adult users of language, teachers of language, as participants in a research project and as tool constructors in Logo.'

Most of the work centred around pluralising nouns, a far from trivial task in English⁶. Karen demonstrated a number of student written procedures, in particular, one by Margo Lamond with a number of logical rules and a long list of exceptions. This is how it fared with words suggested by the audience:

<i>Singular</i>	<i>Plural</i>
radius	radius
ox	oxen
cannon	cannon
salmon	salmon
ovum	ovums

Along the way we were introduced to sniglets, words that don't exist but ought to. Some samples:

cheertrainaway	farewell unwanted visitors
honourschool	praise a school
layrabbitdose	a dose of medicine which lays a rabbit out for 5 days

Gary Stager New Environments for Intellectual Expression: Mac
LogoWriter and LogoEnsemble
(or, Over an hour of hyperactive hysterics by the past president of the Wayne Hills
High School Computer Club)

'This presentation will showcase two powerful new Logo-based environments, LogoWriter for the Macintosh and LogoEnsemble. This session will provide examples of how these two new software environments may be used across the curriculum and provide strategies for teachers interested in making Mac LogoWriter and LogoEnsemble an integral part of their school. There will also be a demonstration of how interactive multimedia (including laserdiscs) and hypertext projects may be created with this software.'

The features of Mac LogoWriter came first, with the hypertext tools, the talking word processor (the screen as data again) and a videodisc demonstration. A quick introduction to LogoExpress followed, based mainly on a graphing package written during the conference, apparently the result of a 'dare' by some other software exhibitor. It quite happily graphed mixed data (eg. 1.6 metres and 10 cm) from the database 'side' of the page. Would it replace AppleWorks? Perhaps not, but it was more functional, and in any case, AppleWorks had simply produced 'fifth graders with secretarial skills.'

ACEC'92

Next year's ACEC will be in Melbourne, July 5..8th, with the theme

'Computing the Clever Country?'

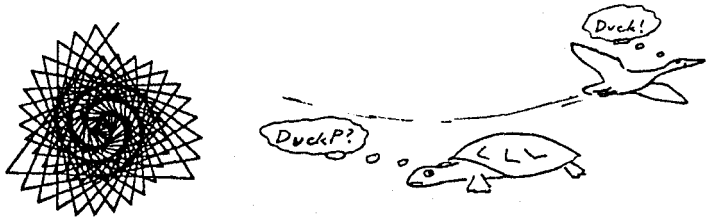


⁶ For an extended discussion of this, and other grammatical matters, see Goldenberg, E P and Feurzeig, W *Exploring Language with Logo*, MIT Press, 1987

Resources

Object Logo 2.51 rev2

Received just as the previous issue went to press was the latest version of Object Logo. This release has a couple of new features, fixes a couple of bugs, and is System 7 compatible. Among the new features is the ability to cut and copy from a Turtle window (and paste directly into something else, like this, without saving the screen and going via a graphics package):



There are new primitives to handle background patterns and colours, and a 'new Turtle initlist keyword, DuckP.' It's there to cover the case of multiple Turtles in the one window slowing each other's movements.

Announced with the upgrade is a series of applications and utilities disks, written by users, a 'Frequent Promoter Program' as an incentive to 'sell' Object Logo, and the news that Brian Harvey's *Computer Science Logo Style* programs will be available on disk.

Paradigm Software is in Cambridge MA (where else?), and Australian agents are Oxford Systematics (03) 459 9761,

LogoEnsemble

Expected 'real soon now' is the latest from LCSi. It's a combination of LogoWriter, LogoExpress, and a simple, but effective, database. The first version is for the Apple IIGS, but an MS-DOS version will come in about 18 months.

The database forms a third side to a page (the page metaphor is now being stretched a bit), and consists of cells, something like a spreadsheet. New primitives allow access to the data in cells, rows and columns, including the ability to total and average numeric values. Other new primitives include logarithm, power, exponent and pi. Word processing, with bold (which shows in yellow on screen), *italic* and underline, can be in 80 or 40 columns. Turtle shapes are part of the page, instead of being on a separate shapes page, and the shape editor is greatly improved. Hit π -7 and you can print the shapes. The mouse is used for page flipping, scrolling, and for drawing like a paint program.

The usual LCSi site licences and lab packs will be available, and the package comes with four sample databases with project material: Nutrition, Planets, Countries and Olympics.

More when it arrives.

The Nurnberg Funnel: Developing Minimalist Instruction for Practical Computer Skill.

With some experience, most of us can cope with new versions of Logo and other packages, new operating systems, and the like. But for the newcomer the computer represents an almost alien intelligence, with manuals and tutorials written in an incomprehensible foreign language. The technical writing profession has made great strides in recent years to make computers more accessible.

John Carroll is Manager of User Interface Theory and Design at the IBM Watson Research Centre. His book documents a number of experimental approaches to developing instructional materials, based on the 'minimalist' philosophy, that is, providing only the minimum of information, and putting the learner firmly in control.

Much of the book is taken up with descriptions of the approaches taken by Carroll and his team, and discussions of the results and their statistical analysis. Beginning with a simple comparison between a minimum manual and the full system documentation (for the IBM Displaywriter), their materials evolved through guided exploration to 'training wheels', cut down systems with careful error trapping⁷. The consequences of errors were not confusing tangles, but, with appropriate feedback, opportunities to learn what was and was not appropriate. Carroll is well aware of the value of, and need for, errors by the learner: 'If errors are never made or if they are made but not clearly analyzed, the learner will lack the raw material to construct an understanding that discriminates between errors and nonerrors.' (p 87)

Carroll found that tutorials were often ineffectual, and in one case his most successful learners were two who did not use a tutorial but simply explored and learned by trial and error. Like most such tutorials, it, rather than the user, was in control.

This, of course, is the crux of the matter, well summarised in a paragraph I found in my first cursory flip through the pages:

'It is sometimes said that the unmotivated learner cannot be helped and the motivated learner cannot be stopped. In designing training, it is critical to recognise and to bear in mind always that adults learning to use computer equipment, often in the context of the workplace, already have goals. They do not come to computer applications to be drilled, led step by step through prerequisites, or tested for mastery of trivial skills. They want to get something done. Instruction that fails to support this motivational orientation fails as instruction.' (p 78)

True not only of adults learning new skills and techniques in the workplace, but also of students in classrooms. Yet so often teachers do little more than give step by step drills of trivial skills, the kind of approach that gives rise to test questions like: 'When inserting a row into a [Works] spreadsheet, does the new row appear above or below the cursor?' (You can think of a Logo example.) The kind of approach that Carroll shows to be ineffectual.

It is perhaps rare to find learning theory discussed in a computing book, but this one could well be set as reading for educational psychology or instructional design courses. Chapter 4, entitled 'The Paradox of Sense-Making', is a well written critique of the systematic drill and practice approach, contrasted with constructionism. A number of prominent educationalists appear in the 24 page bibliography: Ausubel, Bloom, Bruner, Gagne and Piaget are there, and the list includes a well known book by Papert. (Another to catch my eye was one by P. Smith; *Insult to intelligence: Bureaucratic invasion of our classroom.*)

There is no Nürnberg Funnel, the legendary device through which knowledge could be poured to make one wise very quickly. Knowledge and skills are not liquids. They cannot be passed as though they were parameters between the procedures and functions of a program. They are structures in the mind, built piece by piece as ideas and hypotheses are tested against perceived realities as the learner experiences the world.

Carroll's book is a very welcome one, valuable to the technical writing profession for which it was written⁸, and valuable to all teach with computers. The details: Carroll, J. *The Nürnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*, MIT Press, 1990



⁷ As a cyclist, I don't believe in training wheels, but that's another story.

⁸ This (abridged) review was written for the S. Aus Society for Technical Communication newsletter.

Missing out on the Mindstorms

Bill Kerr

Before I read *Mindstorms* and had only read about *Mindstorms* I gained the impression that Papert's educational philosophy was open ended discovery learning and that was about it.

Some of the articles that I have since read about Logo or about Papert's philosophy convey just this sort of impression, They talk vaguely about the 'Logo philosophy' and about how some teachers who use Logo are aware and others are not aware of it.

It could be that either some Logo commentators do not understand Papert or, alternatively, they water him down so as to make him appear more respectable. I don't think that this is right. If Papert's ideas are important then we ought to find out what he is on about and if they inspire us, passionately propagate them. After all, ideas when they are put into practice do change the world, either for better or for worse.

At best, some writers about Logo talk about the importance of Logo to problem solving, debugging (children reflecting constructively about their 'mistakes') and using Logo to develop learning about learning and all that guff. In other words, the sort of reflections on Logo that often pass for informed educational comment are so consistent with current modern educational thinking they would scarcely cause a ripple in the mind of the informed teacher. No *Mindstorms* here!

In my view the central tenets of Papert's thesis are educationally, socially and politically somewhat more radical. So, what is Papert really on about?

Constructionism

Papert's beliefs are rooted very firmly in Piaget's findings about children's learning. Papert worked with Piaget for five years, applying his own expertise in maths to help build Piaget's theories. Two points from Piaget stand out:

- Children build or construct their own intellectual structures.

From this point arises the obligation of the modern teacher to restructure traditional subjects such as maths to fit the child. Hence, Papert has restructured maths by inventing the computing language Logo to fit the natural development of the child.

- Children build on what they know. Piaget's term for children's continual balancing of existing cognitive structures with new experiences is equilibration.

From this point arises the obligation of the modern teacher to investigate the cognitive structures of their students and to interact with those cognitive structures in a subtle, not a heavy handed manner.

Piaget found that incredible amounts of learning occur without formal teaching. In his work, Papert tries to discover and promote the factors that are causing this 'hidden' learning and also asks 'Why is it that learning often does not occur with formal teaching (and often does occur without formal teaching)?'

Mathsland: Restructuring traditional knowledge

Piaget was not an educational psychologist but a genetic epistemologist. These obscure words are highly significant. Papert had recently moved to a new lab at MIT which had been named the Learning and Epistemology Group. Clearly, epistemology is central to the concerns of Piaget and Papert. So, what is epistemology and what is genetic epistemology?

Piaget has recognised it as a mistake to separate the learning process from what is being learned. The study of what is being learned is epistemology. Hence, a genetic epistemologist is a person who investigates the evolution of the structure of knowledge in the minds of young people.

This is a much more dynamic conception than a traditional psychology of the learning process which passively accepts the traditional structure of knowledge as a given. Piaget and Papert are suggesting that there is a dialectical relationship between knowledge and people. Papert quotes Warren McCulloch tellingly to make this point: '*What is a man so made that he can understand number and what is number so made that a man can understand it.*' (*Mindstorms*, p 164)

In looking at learning it is not enough to look at 'learning how to learn' (ie. concentrate on the learner) but we need to study the basic structure of the subject itself. Papert investigates the basic structure of mathematics in some detail including a critique of the formal logical thinking emphasised in Bertrand Russell's *Principia Mathematica* and the 'new math' of the 1960..70s. In Piaget and Papert's view the basic structure of maths is derived from the thinking of the Bourbaki school: order, proximity (topology), combination (algebra). In Papert's view it is not natural that advanced maths ideas are inaccessible to most. What Papert has tried to do is restructure maths so as to accommodate the natural tendencies of the child. Instead of mathophobia Papert hopes to create a mathsland where it will be natural to learn maths, like learning to speak French in France.

Logo was designed with this philosophical/mathematical background in mind. Logo was developed as a language so that mathematically naive users could learn how to program and control the computer as well as more sophisticated users.

Tools, culture and people

Change is inevitable but widespread change will only occur when there are significant changes in the wider culture. This applies to both social change and change in patterns of intellectual development.

The printing press on its own did not create poetry, but by spreading poetry around it helped to create new poets. The steam engine on its own did not create the industrial revolution. Tools are made by people and when tools call out for revolution they will speak through people.

Computers will not create an educational revolution- Forget about computers (for a minute); culture is central to change. Papert is not a mechanical technological determinist. He is concerned with reconceptualising traditional subject domains and using, in this instance, the computer as a tool to help do this.

This is a vitally important point when we come to evaluate the effectiveness of Logo for if Logo is implemented as a technical act (in a formal, teacher centred, Instructionist classroom) then obviously Papert's beliefs are not being given a fair trial. Papert has clearly rejected this technological determinism:

'Technocentrism refers to the tendency to give a ... centrality to a technical object - for example computers or Logo ... (this) betray(s) a tendency to reduce what are really the most important components of educational situations - people and cultures - to a secondary, facilitating role. The context of human development is always a culture, never an isolated technology ...' (Papert, quoted in Solomon, p.128)

Since culture is central to change then it follows that a teacher ought to aspire to be an anthropologist. The computer is merely one important recent addition to the cultural landscape. The question that the anthropologist/teacher ought to focus on is which cultural materials are relevant to intellectual development.

The computer will not replace the teacher. On the contrary, teachers will have to become more skilled to incorporate the new technology into the overall educational context:

- Skilled in modern learning theories and psychology
- Skilled in relating to a variety of children
- Skilled in detecting new, important elements of their student's culture
- Skilled in cross curricular applications
- Skilled in computing
- Able to apply a variety of skills creatively

These skills are necessary for a modern educational system. Currently, one of the main problems with regard to developing creative applications of computers in education is training teachers with these skills. But let's not blame the teachers for this when education departments and governments are not providing the time, the infrastructure or the educational insights to make it all possible.

Papert has proposed a new field of teacher training called humanistic computer studies, where:

'In my vision of this field its professionals will need special combinations of competences. Apart from a foundation in scientific knowledge and technological skill they will need high degrees of psychological sensitivity and artistic imagination. For the ones who will make the greatest social contribution will be those who know how to mold the computer into forms which people will love to use and in ways which will lead them on to enrichment and enhancement.' (from Solomon, p.133)

The role of the computer

If culture is central then what is the role of the technology? The new technology provides the underlying basis for a radical change in the educational and social system. Computers are obviously an important new part of our popular cultural landscape (just like Bart Simpson!) and everyone agrees that their influence will grow in the future.

However, the point is that the future possible pathways for education and society are manifold and that these decisions will be made in the cultural and political arenas- popular culture often determines political expediency

In today's world computers will user in new cultural change but the sort of change that occurs will be fought out socially, in the world of business (how can productivity be maximised?), in the world of institutionalised education, in schemes for alternative schools, in the home with PCs, in the arcades with the latest computer games. There is no social inevitability about the future pattern of usage of computers.

Computers may be used to increase productivity by crunching words, numbers and data. Others will use them as an expressive and creative tool to develop individuals with new insights into traditional subject domains, including human psychology. As a tool the computer is versatile enough to do both. Alan Kay has claimed that the computer can be used to simulate anything:

'... the computer is a medium that can dynamically simulate the details of any other medium, including media that cannot exist physically ... it has degrees of freedom for representation and expression never before encountered and as yet barely investigated.' (*Sunrise Notes* Number 2, June 1990, p.29)

Papert says that the role of the new technology is twofold: both instrumental and heuristic.

Instrumental simply means that as computers become cheaper, more powerful and more popular they will carry and spread the ideas and social relations embedded within them amongst larger and larger groups of people. Papert expresses the instrumental role of computers spreading ideas around very powerfully with the metaphor 'computer as a pencil'.

The heuristic influence of computers is a more complex and surprising idea.

Computing Science is not fundamentally a technical science of computers. Rather, most of it is the science of descriptions and descriptive languages. Hence computing science (especially AI research) has something to offer learning theory, since descriptive languages are used to talk about learning. At an elementary level it is clear that concepts such as input, output, feedback, subprocedures (modularisation), recursion, debugging and extensibility could provide at least part of a framework for explanations of biological and human behaviour.

Papert and Minsky argue that ideas from computing science are instruments of explanation of learning and thinking. More, they are instruments of changing, altering the way in which we learn and think. In this way computing science and AI Research has ushered in a whole new theory of human psychology as outlined by Minsky in *Society of Mind*.

Papert's critique of the education system

'Those who invented the automobile didn't do so by an in depth study of the horse and buggy.' is Papert's comment on the educational horse and buggy.

Papert is scathing of the established education system. He perceives our present schooling process as a technical act under the guiding methodology of Instructionism.

Although instruction is fine and an inevitable part of everyone's everyday learning this is different from Instructionism which is the entrenched methodology of a central person or curriculum transmitting pre-established pieces of information to an essentially passive, captive audience. Papert is against the teacher as technician under the control of the curriculum, against centralised control, against hierarchy, against the whole notion of a centralised curriculum and against accountability and national testing based on the above precepts. In short, Papert is swimming against the current tide of educational tightening up in this country but in doing so he is giving us some powerful weapons to effectively oppose the current disastrous, straight-jacketing trend. Papert's weapons are the powerful ideas outlined above plus computer software (Logo), computer hardware and other hardware like LEGO.

In a dynamic, living culture there is little place for a centralised curriculum because the culture will generate its own interesting, unpredictable challenges on a day to day basis. Attempts to impose a curriculum onto this culture would only serve to cramp the style and creative interest of those who work within the culture.

Instructionism is misguided because it treats children as empty vessels to be filled up with knowledge. Instructionism ignores Piaget who emphasises that children construct their own internal mental worlds by integrating new information with already established structures (equilibration).

Hence, Piaget's findings and not computers as such are at the centre of Papert's radical critique of the education system. Papert would oppose the use of computers for Computer Aided Instruction (CAI) such as maths drill as a band-aid to patch up a basically sterile system.

In opposition to Instructionism, Papert advances the guiding principle of Constructionism for creating a humane and enriching education system. The learning environment is about building and creating things, eg. building rich cognitive structures internally and building things like LEGO machines externally. In this environment the teacher is first and foremost a fellow learner (who might spend more time instructing others simply because he/she may know more).

There is nothing new in Papert's critique of the education system up until now. In the history of education there has always been alternative schools with an emphasis on freedom. These movements have never really caught on partly because '...they were unable to handle the more formal aspects such as mathematics or grammar or many parts of science.' (Papert, address to WCCE, 1990) So, what is new in Papert's vision in the use of modern technology (computers with LogoWriter and LEGO TC Logo) to make possible interesting constructivist maths, science and grammar for perhaps the first time ever, historically.

Conclusion

Many teachers are enthusiastic to start with. Then, after ten years many teachers are burnt out Instructionist hacks, despite their best intentions. Don't blame the teacher, blame the system.

Fundamentally, Papert influenced us because, if we really listen to him, he politicises the educational debate in a highly practical way. Papert has taken the most traditional subjects, maths and science, and has begun to restructure them to fit the user. Papert and his supporters have created an interesting maths-land and science-land that are both user friendly and powerful learning environments.

Hence, Papert and the MIT group are creating conditions that make it possible for people to become passionate about educational options. LEGO TC Logo is something that you can take home and happily play with. It is hard to be passionate about maths drill and practice style textbooks, or Instructionism- broadcasting essentially the same lesson year after year, marking Common Tests, or whether Sarah

was really worth a low A or a high B. Constructionism and Logo are different. It fits the user and has no ceiling in terms of expertise.

Papert's ideas have the power to change lives and to change whole education systems (e.g. Costa Rica). Of course this will require a tremendous and possibly protracted educational/political struggle since the Instructionist model casts such a long shadow. As in all meaningful struggles the outcome is far from certain.

Since Papert's ideas are revolutionary they are not for the faint hearted. It is very difficult to mentally step outside of a system you are working in, that you are part of, that you help to reproduce by your day to day actions and then to turn around and to say that it is fundamentally at fault. It is easier for Papert to make this critique than it is for the practising classroom teacher. In the final analysis, Papert invites us to have the courage to embark on the adventure of tearing down the old ways while creating the new ways of teaching and learning.



OZLOGO

A new Logo Special Interest Group

During an ACEC'91 lunchtime an informal meeting was held to publicise and explain OZLOGO. To quote from the brochure:

'A Special Interest Group dedicated to teachers concerned with the use of Logo in primary, secondary and tertiary education has recently been formed.

The aims of the OZLOGO Group are to:

- Promote the use of Logo as a valuable educational tool.
- Meet and share ideas.
- Produce a regular Newsletter/Journal.
- Run inservices for teachers and holiday seminars for students.
- Provide a mutual support network.'

Moving force behind OZLOGO is David Rasmussen, and anyone interested should contact OZLOGO at POB 1061 Richmond North Victoria 3121, or, if you have the right connections, OZLOGO@sunrise.bf.rmit.oz.au

Wanted

A working copy of LogoWriter for the Commodore 64, a version no longer supported by LCSi. Oakbank Area School's disk refuses to do anything. Can you help?

Please contact either POALL or Richard Coyte at Oakbank Area School, Main Road, Oakbank SAus 5243



Computing at Entropy House

Bill Kerr is yet another victim of the state of state education in South Australia. He's now at Renmark High School after spending three terms at Padthaway Primary School (between Kieth and Naracoorte). Before that he was at Henley High for some years. His family still lives at home in Adelaide. The system seems to be 'fundamentally at fault.'

Apple has released the Avalanche CD-ROM, with some 90 MB of educational HyperCard material, much of it student written. Why should the HyperCarders have it all their own way? Send some Logo to the developers for the next edition

Had an interesting question the other day from a student of someone who was conspicuous by his absence at ACEC'91 (He apparently could have been heard on Melbourne radio talking football). Student wanted to write a virus in LogoWriter, inspired by this one on p 54 of *Thinking Logo...*

```
TO Replicate.0 :generation
PR SE (This is) WORD "Replicate. :generation
POTS
DEFINE WORD "Replicate :generation + 1
      TEXT WORD "Replicate :generation
ER WORD "Replicate :generation
RUN LIST WORD "Replicate :generation + 1 :generation + 1
END
```

... and he wanted it to spread, be difficult to detect, and to change procedures.

Best I could do was this:

```
to startup
if front? [flip]
top
pr (to x)
pr [top select]
pr [repeat 7 [cd]]
pr [copy]
pr [pr " paste]
pr [x]
pr [end]
x
end
```

Fortunately, or unfortunately, depending on your point of view, LogoWriter is largely immune to the kind of tricks that can be played with other Logos (and have been described in earlier issues of POALL). But my original purpose was not to be malicious, but to draw attention to questions of recursion, self-reference, and self-replication. I suggested he rethink the project.

Coomababah seems to be where the action is. While the primary crew was at ACEC, a secondary team was on the way to success at the 1991 Pedal Prix at the Adelaide International Raceway. A century after the safety bicycle took over from the ordinary (ie. the penny farthing), the bicycle and its cousin the HPV are enjoying a resurgence that seems likely to continue, with increasing traffic/fuel costs/greenhouse as incentives. Incidentally, that well known cyclist, who today would wear 'cycling clothes, fluorescent to be seen', Mulga Bill, rode a safety bicycle, not an ordinary. Banjo was among the writers to disparage the new machine.

What has that to do with Logo? Nothing, except that the Pedal Prix organisers understand constructionism. We learn about things by building them, in either hardware or software.