Online Lingo Resources

This document and the accompanying movies and HTML pages provide a resource for anyone learning to use the Lingo programming language embedded in Macromedia™ Director™.

Author	Ian Phillips DipAD, PGCE, MSc iphillips@patrol.iway.co.uk http://www.i-way.co.uk/~iphillips/home.htm

Funding	Advisory Group on Computer Graphics http://www.agocg.ac.uk/
Date	April 1998
Edition	1.0

about... /2

About this document

This document provides a free-standing companion to the online Lingo programming resources hosted on the AGOCG website. It should be used in conjunction with the accompanying Director movies to support classes and workshops in Lingo programming.

About the author

Ian Phillips is a freelance learning materials consultant and trainer. From 1991 to 1997 he was Principal Lecturer in Computer-Aided Art and Design at Coventry University.

He taught postgraduate students and had a research interest in the teaching of computer programming. Since leaving teaching he has worked on interactive learning materials for Oxford University Press and the University of Westminster, among others.

Acknowledgements

Thanks to: John Jostins for advice on Director project planning; Mark Peden for lots of help with Lingo; Simon Turner for occasional but valuable help; Tab Julius and all contributors to the lingo -l mailing list; Rosey Bennett for editorial input, and teachers such as John Vince, Simon Ritchie, Garry Bulmer and Alan Chantler who showed me that programming could be explained.

about... /3

Funding
This work was funded by the Advisory Group On Computer Graphics (AGOCG):
`http://www.agocg.ac.uk/`
which is an initiative of the Joint Information Systems Committee (JISC):
`http://www.jisc.ac.uk/`
and the Research Councils in the United Kingdom.

Feedback
The author welcomes constructive criticism of these materials, which should be emailed to:
`iphillips@patrol.i-way.co.uk`
Requests for programming support or further training can only be met as a freelance.

Print record
This document was printed on 1 May 1998.

about... /4

Contents

about... /5

SpriteControl

SpriteControl01	Control by button 1
SpriteControl02	Control by button 2
SpriteControl03	Control by button 3
SpriteControl04	Control by button 4
SpriteControl05	Control by moving another sprite
SpriteControl06	Control by dragging 1
SpriteControl07	Control by dragging 2
SpriteControl08	Control by dragging 3
SpriteControl09	Control by events 1
SpriteControl10	Control by events 2

Navigation

Navigation01	Defining rectangular hotspots 1
Navigation02	Defining irregular hotspots 1
Navigation03	Defining rectangular hotspots 2
Navigation04	Using framelabels
Navigation05	Mapping the cursor position
Navigation06	Calculating the cursor position

about... /6

about... /7

How to use the resource

Locate a topic of interest, pick a movie from the topic group list, read more about it in the "Details" page and download it. When you've downloaded and unpacked the movie (Director 5 remember!), study the code and movie organisation and adapt it to your own requirements. It's as simple as that!

Great care has been taken to make the movies as small as possible, while keeping them informative, to minimise your download times. However, if you want more than two or three movies from a group it will probably be quicker to download the whole group archive. If you have a fast connection (or are not paying the connection charges), you may prefer to download the whole project archive (approx 1.5MB).

If you have any constructive comments about the resource, please mail me at:

iphillips@patrol.i-way.co.uk

but remember that I won't be answering individual scripting questions. For that, I advise you to subscribe to a specialised discussion list such as 'Lingo -l', run by author and programmer Tab Julius.

You can find information on how to subscribe at the Penworks Corporation website, which is listed in the Bibliography.

About the resource

This resource is based upon earlier work producing tutorial movies for art and design students learning Lingo. The downloadable movies are all cross-platform. They are currently in Director 5 format and have been tested with Version 5.0.1 for MacOS and Version 5.0 for Win95. There are no plans for Win3.1 versions.

The movies incorporate a number of features to make them easy to understand. For example, they contain heavily-commented Lingo code and castmembers named and organised to minimise confusion. However, they are designed as a learning resource for those following some course of instruction rather than as a complete open learning package. Please bear this in mind.

The following assumptions are made about the user and her learning environment:

1. User is familiar with Director terms and operation.
2. User has done some basic Lingo scripting.
   If not, she should check out “Learning to Program in Lingo” (660K), the sixty page handbook aimed at novice Lingo programmers which accompanied the original AGOCG tutorial movies.

3. User has access to a licensed copy of Director 5 or 6 and the manuals.
4. User is familiar with downloading and unpacking Internet archives.
5. User's machine can display thousands of colours at 640x480 pixels.

I will respond to email about problems with the site but I cannot answer individual scripting questions.

About the movies

The movies are in thematic groups but be aware that there is no hard-and-fast division between these groups. Any Lingo project will probably involve concepts and techniques from more than one group.

There is also no need to progress in linear fashion from the first group to the last when downloading and studying the movies, though this is advisable.

movies... /10

Basic Lingo

In this group you will find movies demonstrating:

- execution of statements and flow of control
- simple use of local and global variables
- basic text handling
- simple use of lists

There are several examples in some categories. A number of the examples use the same graphic sprites in an attempt to focus on the differences in the Lingo constructs. These differences will only become clear when you download and open the movies.

BasicLingo01 - Sequential execution of statements

Lingo executes individual program 'statements' sequentially, in common with most other programming languages. In this movie, four statements are attached to buttons designed to be clicked in sequence.

A 'Reset' button executes all four statements in reverse order. Additional feature is status messages about each button click.

movies... /11

BasicLingo02 - Conditional execution of statements 1

Lingo can execute statements conditionally, by 'branching'. The simplest form of branch is:

```
IF <condition> THEN
  <action(s)>
END IF
```

In this movie, an alarm sounds IF a clockface shows a particular hour. A 'Reset' button resets the time. Additional features are: an alarm cutoff button and a status message sent when Director enters the branch.

BasicLingo03 - Conditional execution of statements 2

Lingo can execute statements conditionally, by 'branching'. This movie illustrates the two-way branch:

```
IF <condition> THEN
  <action(s)>
ELSE
  <alternative action(s)>
END IF
```

In this movie, an alarm sounds IF a clockface shows a particular hour. An alert dialogue is displayed when the alarm is switched off: it is worded one way IF user's machine is a Mac ELSE if user's machine is a PC, it is worded differently. A 'Reset' button resets the time. Additional features are: an alarm cutoff button and a status message sent when Director enters the branch.

movies... /12

BasicLingo04 - Conditional execution of statements 3

Lingo can execute statements conditionally, by 'branching'. A multiple branching logic is more efficient than repeated use of 'IF...THEN...' statements. This movie illustrates the multi-way branch:

```
CASE <case condition> OF
  <expression1: action(s)>
  <expression2: action(s)>
  <expression3: action(s)>
  <...>
  <[otherwise action(s)]>
end CASE
```

In this movie, an alarm sounds IF a clockface shows a particular hour. An alert dialogue is displayed after the alarm is switched off: it is worded differently according to the current day of the week. A 'Reset' button resets the time. Additional features are: an alarm cutoff button and a status message sent when Director enters the branch.

BasicLingo05 - Repeated execution of statements 1

Repeated execution of sequential statements is more efficient and less tedious to code (or read) if Lingo's 'repeat loop' capabilities are employed. There are four constructs:

```
REPEAT WITH <counter> = <start> TO  <finish>
  <action(s)>
END REPEAT
```

```
REPEAT WITH  <counter> = <start> DOWN TO <finish>
  <action(s)>
END REPEAT

REPEAT WITH <variable> IN <aList>
  <action(s)>
END REPEAT

REPEAT WHILE  <testCondition>
  <action(s)>
END REPEAT
```

This movie demonstrates repeated execution of Lingo statements using the REPEAT WITH.... keyword construct. Castmembers representing the hands of the clock are swapped a set number of times. An alert dialogue gives the number of repeats performed. Additional features are: an alarm cutoff button and status messages sent just before and just after Director enters the loop.

BasicLingo06 - Repeated execution of statements 2

There is a companion construct to 'REPEAT WITH...': 'REPEAT WITH... DOWN TO...'. This movie demonstrates use of this construct to run a clock backwards. An alert dialogue gives the number of repeats performed and the (slightly unusual) construct used. Additional features

are: an alarm cutoff button and status messages sent just before and just after Director enters the loop.

BasicLingo07 - Working with variable values

We often wish to work with values that we do not know in advance. Such values can be stored as 'variables' and Lingo has two kinds: local and global. Here is an example use of a local variable:

```
set myName = "Ian"
put "Hello" &&myName
```

The double ampersand concatenates (joins) the text in quotes and the variable with a space between, producing the output:

```
Hello Ian
```

Local variables do not have to be declared but the value of a local variable is not transmitted beyond the handler containing it. If that facility is required, then a global variable must be used. Global variables have to be declared. Here is an example use of a global variable:

```
global gMySpecialSprite

on mouseDown
  set gMySpecialSprite = the clickOn
  puppetSprite gMySpecialSprite, TRUE
end
```

```
on mouseUp
  set x = random(100)
  set y = random(100)
  set the loc of sprite gMySpecialSprite = point(x,y)
  updatestage
end
```

Both 'mouseDown' and 'mouseUp' handlers have access to the global variable's value. Note that the 'mouseUp' handler also uses two local variables, 'x' and 'y'. Any sprite with this script attached would be moved around the stage when clicked. This movie demonstrates use of local and global variables within repeated execution of Lingo statements. It is based on the 'REPEAT WITH...' demo movie, 'BasicLingo05' but here the clock animation loop uses a local variable for the castmember required and sets a global variable recording the number of repeats. Additional feature: the ALERT message giving the number of repeats performed takes the number from the value of the global variable set by the animation repeat loop, instead of having it 'hard-coded' into the relevant button script.

BasicLingo08 - Storing and retrieving data with lists

Data can be stored in 'lists', of which the simplest form is a linear list. Here is an example of a linear list:

```
set myList = ["jack","jill","fred"]
```

Here the list is set up with three text entries. The entries could equally

well be numbers, symbols (starting with the hash character '#') or variables. Lists are manipulated with special commands such as:

```
getAt(myList,2)
```

which will retrieve "jill", the second entry in the list, and:

```
add myList,"sue"
```

which will add "sue" at the end of the list.

A list would typically be intitialised empty and entries would be added later, perhaps using a repeat loop. For example:

```
set myList = [ ]
repeat with i = 1 to the number of members of castLib
"Internal"
  add myList,(the name of member i)
end repeat
```

This loop will cycle through the internal cast, adding an entry consisting of the member name each time.   This movie demonstrates use of linear lists, using loops similar to the above to build two lists. The button 'List Entire Cast' does just that, calling a handler which produces a list containing one entry for each castmember name. The button 'List Specified Cast' builds another list, using an 'IF THEN' construct with two conditions: that the entry contains both "Countdown' and 'BMP'. These characters are found only in those members used in the clock hands animation. Additional features are: feedback messages in status field and two buttons, 'List Entire Cast' and 'List Specified Cast'.

movies... /17

BasicLingo09 - Repeated execution of statements 3

There is a companion construct to 'REPEAT WITH...': 'REPEAT WITH <variable> IN <aList>'. The syntax is:

```
REPEAT WITH <variable> IN <aList>
  <action(s)>
END REPEAT
```

This assigns successive values from the specified <aList> to the <variable>. It may be a more convenient REPEAT construct if the loop control or actions are related to data best held in a list. It is necessary to have some familiarity with variables and lists to use this form: these topics are introduced in earlier movies. This movie demonstrates use of this construct to create the clockface animation by looping through a list of the relevant bitmap castmembers. An alert dialogue gives the number of repeats performed, as in earlier REPEAT movies. Again as in an earlier movie, this number is taken from the value of the global variable set by the animation repeat loop, instead of having it 'hard-coded' into the relevant button script.

movies... /18

BasicLingo10 - Repeated execution of statements 3

The last form of repetiton in Lingo is REPEAT WHILE <condition>. This movie uses the number of the last animation castmember as the loop test condition. In other respects, the Lingo is very similar to that in the REPEAT WITH... movie. An alert dialogue gives the number of repeats performed, as in earlier REPEAT movies. Again as in an earlier movie, this number is taken from the value of the global variable set by the animation repeat loop, instead of having it 'hard-coded' into the relevant button script.

Sprite Control

In this group you will find movies exploring sprite control:

- with buttons
- by moving another sprite
- by dragging a sprite
- by external events (key press, mouse click etc)

There are several examples in some categories.

SpriteControl01 - Control by button 1

Sprites can be controlled by designating them as puppets and changing one or more of their properties. This movie demonstrates control of a property with buttons. The property changed is 'the locH of sprite'.

A global variable is used to identify the puppet sprite and a local variable to modify the horizontal location. There is no limiting or error-checking code.

movies... /20

SpriteControl02 - Control by button 2
In this movie, the following properties of a sprite are changed:

a. foreColor of sprite
b. stretch of sprite
c. height of sprite and width of sprite

There is no limiting or error-checking code. Additional feature: status messages about current values of properties; Reset button.

SpriteControl03 - Control by button 3
In this movie, the following properties of a sprite are changed:

a. locH of sprite
b. locV of sprite
c. foreColor of sprite
d. backColor of sprite

There is no limiting or error-checking code. Additional feature: status messages about current values of properties; Reset button.

SpriteControl04 - Control by button 4
In this movie, the horizontal location of the sprite is changed with two arrow buttons. The sprite moves continuously while mouse button is held down. Limiting and error-checking code is introduced to provide the following additional features:

a. movement is constrained to invisible 'play area'
b. movement is reversed if sprite hits limit
c. aural feedback if sprite hits limit

The 'continuous operation' code is adapted for the movie 'SpriteControl05'

SpriteControl05 - Control by moving another sprite
This movie demonstrates control of a gauge sprite with a slider constrained to move only between the top and bottom of a track. The 'height of sprite' property of the gauge sprite is changed by moving the slider, with the height value being calculated from the vertical position of the slider.   This technique can be applied to the control of sound volume or the playing of digital video. Digital video members can of course be 'sound-only'. Additional feature: display of code fragment, containing a 'repeat while the stillDown' loop, which controls both slider and gauge.

SpriteControl06 - Control by dragging 1
This movie demonstrates mouse control of a sprite constrained within a non-rectangular path. Additional features are:

a. visual and text feedback if illegal move is made
b. visual, text and aural feedback if sprite is moved to end of path

c. a score conditional on number of illegal moves
d. elapsed time readout seconds

SpriteControl07 - Control by dragging 2
This movie demonstrates the checking of a moveable sprite's location, using the comparison operators 'Sprite...Intersects' and 'Sprite...Within'. The different results obtained using COPY and MATTE inks for the sprites can be observed. Additional features are:

a. visible bounding boxes round the fixed and moveable sprites
b. ability to switch between COPY and MATTE ink frames, to observe effect on truth of comparison.

There is also a status message giving the position of the square relative to the circle.

SpriteControl08 - Control by dragging 3
This movie is another demonstration of the use of the comparison operators 'Sprite...Intersects' and 'Sprite...Within'. When used in this way, there is obviously a need for some additional code to prevent the 'ball' being moved into the 'water' through the side of the 'bucket'.

SpriteControl09 - Control by events 1
This movie demonstrates sprite control through response to events: mouse clicks in this case. A list of four sprite properties is created on

startup and each click on the ball sprite selects a property at random. A new value for the selected property is also generated; at random but within fairly small limits.

Use of modifier keys to select a particular property is suggested as a possible development. Additional feature: a 'panic reset', triggered by holding down the CONTROL key while clicking anywhere on-stage, for use if any operation renders the sprite invisible.

SpriteControl10 - Control by events 2

This movie demonstrates sprite control through response to events: key presses in this case. A handler determines which of the four arrow keys has been pressed and another handler then moves the sprite in the appropriate direction. The moving sprite is constrained within a non-rectangular path. Additional features are:

a. visual and text feedback if illegal move is made
b. visual, text and aural feedback if sprite is moved to end of path
c. a score conditional on number of illegal moves being below a set limit
d. sprite location readout in the form point(x,y)

There is also a status message about the direction of travel.

movies... /24

Navigation

In this group you will find Lingo-based approaches to such navigational aids as defining hotspots and referencing framelabels. The hotspot techniques can employed to create toolbars or navigation buttons in the manner of of a WorldWideWeb image-map.

Navigation01 - Defining rectangular hotspots 1

Navigation in a Director project often relies on locating the cursor in relation to parts of the stage designated as 'hotspots'. This movie demonstrates the division of the stage into regular, rectangular hotspots or zones with invisible 'position detector' sprites made from a bitmap. Additional features are: feedback messages in status field; buttons to show the individual zones used and a button to show all zones simultaneously overlaying a map.

Navigation02 - Defining irregular hotspots 1

Irregularly-shaped hotspots are sometimes required. This movie uses a bitmap castmember for a sprite with 'Matte' ink which is overlaid on a map to create the hotspot. The function 'the mouseCast' is used instead of 'rollOver()' to detect positioning of the cursor inside the actual shape, rather than just somewhere within the bounding box. Additional features

are: invisible detector sprites dividing rest of 'stage' into rectangular zones and feedback messages in status field.

Navigation03 - Defining rectangular hotspots 2
Rectangular zones can be defined without using sprites. This movie uses a property list to define the zones. Avoiding sprites frees channels for other uses and reduces movie file size. Additional features are: a permanent crosshair marking junction of zones; feedback messages in status field.

Navigation04 - Using framelabels
Text in fields can be made clickable. This movie uses 'the labelList()' function to build a list of framelabels which is stored in a field, one label to a line. The labels can then be used to navigate a movie. The technique could be applied outside of the Shockwave environment to generate a clickable list of filenames. Additional features are: popup display of framelabels in the score; feedback messages in status field.

Navigation05 - Mapping the cursor position
Lingo provides a means of mapping one rectangle onto another. This function can be used to map the cursor position from a small 'model' rectangle to a larger 'real' one. This movie demonstrates modelling of a restricted part of the stage and use of the 'map()' function to move a

sprite around the 'stage'. Additional feature: a button to show comparative proportions of model and stage rectangles.

Navigation06 - Calculating the cursor position
Relatively simple maths can be used to calculate the proximity of the cursor to designated sprites. This movie demonstrates the application of this technique to the making of responsive hotspots. This is done by changing certain sprite properties in response to cursor movement. Additional features: display of the calculation code; feedback messages in status field. The proximity routines were devised by Mark Peden. Check out more of his work at his home page:

`http://www.mousedown.demon.co.uk`

his amoebic lifeforms at:

`http://www.mousedown.demon.co.uk/amoebic.lifeforms/start.htm`

and his online lifeforms at:

`http://www.mousedown.demon.co.uk/online.life/life.htm`

Animation

In this group you will find Lingo-based approaches to such animation techniques as colour-cycling, image-swapping, and using a list of points to position an image. These techniques can be employed to create small and efficient animations which play well in the environment of a WorldWideWeb browser.

Animation01 - Colour cycling

Colour-cycling is an efficient way to produce simple animation. There are two classic techniques: changing the colour of an object and changing the palette of colours applied to a whole scene.   Palette cycling can be unpredictable in a browser environment, since everything in the window is subject to change, so this movie demonstrates the cycling of sprite colours. These were chosen from the D5 Netscape palette. Additional features: Status messages; display of essential code fragment.

Animation02 - Image swapping 1

Traditional animation consists of a number of frames shown in rapid succession to create the illusion of movement, a technique sometimes called ‘swap-frame’ animation. To avoid confusion with Director’s use of

the term ‘frame’, the technique will be referred to as ‘swap-image’ (or image swapping) in this project.

There is a trade-off between image quality and file size: in this movie 1-bit images created in Director are used and a very small file results. Images can however be produced with any graphics software capable of saving files in PICT format.

This movie demonstrates how Lingo can be used to swap the images assigned to a sprite which stays in one position on-stage. Additional features: speed control buttons; Status field messages.

Animation03 - Image swapping 2

This movie demonstrates application of the image swapping technique to creating the illusion of motion. 32-bit images have been imported from a graphics program and downsampled to 8-bits in Director, using the Netscape palette. This improves image quality at the expense of file size. Additional features: speed changing buttons; Status messages.

Animation04 - Using a list of points 1

It is obviously better to move a small image against a static background than to swap large images, if possible. In this movie, the location of the green telltale is set successively to points stored in a list. Note that points may have to be tweaked to get alignment with other sprites, as has been done in this case.   The technique can produce very fast animation: this example is deliberately slowed down to improve

visibility. Image quality can also be higher as far fewer castmembers are required. The two used in this movie are 32-bit. Additional features: speed changing buttons; Status messages.

Animation05 - Using a list of points 2

In this movie, the 'point-list' method of changing the location of a sprite is combined with two methods of hiliting other sprites based on that location. The technique is used to illustrate the flow of control in a REPEAT WHILE... loop.

The animation in this movie was originally produced as a Score animation and took 404 frames: this version uses only 2 frames. The original Score animation was used as source for a QuickTime movie which was 332K: this movie is 195K.

The main problem with the technique when used to produce Shocwave movies is that animations have to be artificially slowed to be visible at all. This can produce the rather jerky results evident in this example when viewed on some systems. Compare the animation in the downloaded movie with what you see here. Additional features: status messages; aural feedback on START and 'passTestCondition'.

Object-Oriented Programming

In this group you will find just two movies. One demonstrates the application of object-oriented programming techniques to the creation of utilities. Once you have understood the issues and possibilities, you can use this movie as a basis for making your own utilities. The second movie demonstrates the oop approach to animating sprites.

You are strongly advised to consult 'Lingo Sorcery: The magic of lists, objects and intelligent agents', SMALL [1996] for more detailed information and examples.

OOP01 - Utility objects

This movie demonstrates an object-oriented approach to programming utilities and interacting with them. The approach outlined can be applied to offline tasks such as managing files and folders in a CD-ROM project. SMALL [1996] explores the issues in considerable depth. Additional features: text input field; custom message fields to replace 'alert' dialogues; custom messages.

OOP02 - Animated objects

This movie (by Mark Peden) demonstrates object creation and control in Director through the animation of sprites. Additional feature: extensive

tutorial comments in the code. Study Mark's great tutorial comments in the downloaded movie carefully... Check out more of his work at his home page:

`http://www.mousedown.demon.co.uk`

his amoebic lifeforms at:

`http://www.mousedown.demon.co.uk/amoebic.lifeforms/start.htm`

and his online lifeforms at:

`http://www.mousedown.demon.co.uk/online.life/life.htm`

NetLingo

In this group you will find movies demonstrating two aspects of Shockwave-specific Lingo:

a. Using one movie to retrieve another
b. Storing and retrieving data locally

You are strongly advised to consult 'Shocking the Web', CLARKE & SWEARINGEN [1997] for more detailed information and examples.

NetLingo01 - Covering download times

This link loads one small animation which in turn loads a larger movie. The first movie ('NetLingo01a') demonstrates application of the point-list animation technique introduced in the 'Animation' group to the creation of a Shockwave load-time cover movie. This movie preloads the second one. There are no additional features.

The second movie ('NetLingo01b') appears only when loading is complete, minimising the time spent looking at a blank screen. This movie could be anything at all: in this case we use a cut-down version of 'NetLingo02', which loads a previously-saved data file to create a drawing. Note that you have to load 'NetLingo02' at least once and save a drawing before anything will appear in this movie.

The first movie in the sequence is 18K; the second 95K. Note that

movie file size has no impact on the time taken to load the plugin. This is dependent upon the user's platform (the Win95 implementation is appreciably faster than the MacOS 7 one) and machine configuration. There is therefore a limit as to how much benefit you can derive from the 'load-time cover' technique.

NetLingo02 - Storing data on user's machine

This movie demonstrates use of the Shockwave-specific Lingo elements 'setPref' and 'getPref' to store and retrieve local data. Note that most Lingo facilities for file operations are disabled or frowned upon in Shockwave, in line with general thinking on security.

In this example application, the two elements are rolled into handlers to store and retrieve drawings on the user's machine. Additional features: drawing area; buttons to Save and Load data; buttons to Clear or Redraw the area; Status messages.

The idea and core coding come from Mark Peden. Check out more of his work at his home page:

`http://www.mousedown.demon.co.uk`

his amoebic lifeforms at:

`http://www.mousedown.demon.co.uk/amoebic.lifeforms/start.htm`

and his online lifeforms at:

`http://www.mousedown.demon.co.uk/online.life/life.htm`

Annotated Bibliography

On Lingo:
My local branch of Blackwells listed twenty books dealing with Lingo on 30 April 1998, though some were out of date or out of print. Note that the supplied Director documentation is essential for reference, though less useful in a tutorial context. Director 6 users will already have seen the supplied tutorial movies, which are a vast improvement on Macromedia's earlier efforts and very useful.

CLARKE, C. & SWEARINGEN, L. 1997.
Shocking the Web
Macromedia Press/Peachpit Press.
£ 36.95. 447pp book and CD-ROM. An absolute goldmine for anyone developing Shockwave.

THOMPSON, J.T. & GOTTLIEB, S., 1996.
Macromedia Director Lingo Workshop (2nd ed.)
Hayden Books, Indianapolis.
Mid-price at around £50.00.
Very useful tutorial and reference material - I still use the first edition. CD packed with example movies (arranged in Chapter order).

SMALL, P., 1996.
Lingo Sorcery
John Wiley & Sons, Chichester.
£24.95. Excellent but idiosyncratic treatise on object-oriented programming in Lingo. Especially good on working with lists.

MACROMEDIA, 1997.
Lingo Authorized
Macromedia Inc., San Francisco.
£31.95. 250pp pbk & CD-ROM. Useful if a bit pedestrian at times. Step-by-step lessons that actually work.

JULIUS, T., date unknown
Lingo!
New Riders.
£41.99. Tab Julius runs the lingo -l mailing list: judging by that, the book should be worthwhile even though I believe it's now rather out of date.

ROSENZWEIG, G., date unknown
The Comprehensive Guide to Lingo
£ 36.95 700pp book & CD-ROM. Derived from Gary's excellent online tutorials, which were organised as thirty-two short chapters. He supplied plenty of good code fragments.

On-line help and information:

This project was developed and tested on my own site:

`http://www.i-way.co.uk/~iphillips/WORK/LINGO/Lingo.htm`

and you can check it out for a taste of the whole project. The URL for the complete project has not been finalised at the time of writing but I will post a link on my site. Other relevant WWW sites are:

`http://www.macromedia.co/director/`

`http://www.mcli.dist.maricopa.edu/director/`

and they have pointers to other useful sites. An old standby:

`direct-l`

is a mailing list which is a bit too generalised now. A more useful list, devoted entirely to Lingo, is:

`lingo -l`

moderated by Tab Julius. Find out more at the Penworks Corporation website:

`http:www.penworks.com/`

As always, be prepared for congestion and slow response from the WWW sites. It's also best to subscribe to the digest versions of mailing lists to avoid a flood of incoming postings.

A very active developer and distributor of Director Xtras is g/matter Inc. Their site is:

`http://www.gmatter.com/`

and they maintain a mailing list called xtras -l. Details and subscription instructions are available from the g/matter website.

Mark Peden, who collaborated on this project, has developed a number of interesting 'artificial life' animations and a colour conversion palette for Director users. See more of his work at:

http://www.mousedown.demon.co.uk/

http://www.mousedown.demon.co.uk/amoebic.lifeforms/start.htm

http://www.mousedown.demon.co.uk/online.life/life.htm

http://www.mousedown.demon.co.uk/tools/palette.html

Blackwells Online Bookshop is at:

http://bookshop.blackwell.co.uk/

The site is well-organised and delivers search results quickly.

Addresses of other sites of interest may be distributed from time to time if there is enough demand. Email me:

iphillips@patrol.i-way.co.uk

about your favourite site(s), books, or other support materials.