Lingo

Director's Scripting Language

Multimedia – CMO340
Tutorial Slides
By: Mohyuddin

Basic Commands

Commands: Instructions for the movie to do sth.

Play Command:

•Play next

•Play previous

•Play "scene1"

•Play done

•Play "scene1" of movie "movie1"

Go to Command:

•Go to 1

•Go to frame 10

•Go to frame "scene1"

•Go to "scene1"

•Go to the frame

Put Command:

Displays or prints the value on the stage / message window.

• Put the mouseH

• Put the mouseV

• Put the mouseLoc

• Put the soundLevel

Set Command:

Sets the values of the variables.

• Set the volume of the sound 1 to 100

• Set the stageColor=Random(255)

Scripts:

1. Behaviours Scripts:

•Sprite Behaviour / Script:
 Attached to sprites.

•Frame Behaviour / Script:
 Attached to frames on the channels.

•Apply using drag and drop from cast members or choose from the pop up menu.
•Can be attached to more than one location in the score.

2. Movie Scripts:

•Available to the entire movie.
•Examples:
•Available to mouse and key board events
•Available to start, stop and pausing a movie.

3. Cast Scripts:

•Attached to cast members.
•Independent of score.
•Available to sprites assigned from cast members.

4. Parent Scripts:

•Special scripts that contain lingo to create child objects.

Types of Events:

•User feedback events: Mouse and Keyboard events

•Playback events: Movie and Frame events

•Time events: Idle time, timeout events

•Window events: Multiple director window events

Custom Handlers:

•Units of code which execute on event activation.

•Written for events other than specified

•Invoke the name of the custom handler by call in sprite script

•Specify the handlers in the start of movie

•Example: Changing background, Menu

Writing Multiple Scripts for the same object:

•Sprite Vs Cast Scripts

Data Types:

Integers: Whole numbers
Example: 2, -3

Floats: Decimal point numbers
Example: 1.7, -0.8989

Booleans: *Example:* TRUE or FALSE, 1 or 0

Strings: Combination of alpha-numeric or numeric characters in quotes
Example: "dean", "Multimedia design"

Symbols:
- User defined constants, start with #, more faster.
- Can't contain spaces or punctuation.
Example: #dean, #123
abc=#symbol
abc="symbol"

x= string(#abc)
put x
-- "symbol"

Lists: Holding multiple pieces of data at once, which can be integers, floats, strings, symbols or a combination of all.

Example: [1, 2, 3,4], ["text", 1.22, #name1],
CourseList = ["Multimedia", "Databases Management ", "Artificial Intelligence"]

Functions:

getAt(list, postion):
This function returns the value on the list at the position indicated.

getOne(list, value):
This function returns the position of the given value on a linear list.

addAt(list, position, value):
This function adds an entry at a specified position of a linear list.

setAt(list, position, value):
Adds an entry at a specified slot, overwriting the contents of that location.

add(list, value):
Will simply add another entry. If the list is unsorted, the addition will be at the end, if sorted, it will go into its proper location in the alphanumeric order.

append(list, value):
Adds the specified value at the end of the list even if it is sorted.

getLast (list): Returns the last value on a list.

count(list): Returns the number of items on a list.

sort(list): Sorts a list in ascending order.

Variables:

Variables Naming Conventions:

- Single words, without spaces.
- Not starting with a number.
- Can't have ? character.

Types of Variables:

*Local Variables***:**

- Hold data available in current handler being executed.

- Used within holder where they are defined.

Example:
Movie Script:
```
on setLocalVar
    localVar = "Local Variable"
    put localVar
end
```

Message Window:
setLocalVar

Result:
-- "Local Variable"

***Global Variables*:**
- Hold data whose value can change and be accessed throughout the movie.
- Value accessible from one handler to another, and from one script to another.
- Global variables must be declared using the *global* keyword.

Example:
Movie Script:
global gGlobalVar
on setGlobalVar
    gGlobalVar = "Global Variable"
end

on getGlobalVar
    setGlobalVar
    put gGlobalVar
end

Message Window:
getGlobalVar

Result:
-- "Global Variable"

Property Variables:
- Used in Sprite Scripts.
- Contains different values for different sprites.
- Hold data only for the particular object it is attached to.
- Like global, properties need to be declared, using the *property* keyword.

Example:
Sprite Script: Attach it to sprite 1 and sprite 2.
```
-- declaring property variable
property pPropertyVar
on setProperty me, varData
    pPropertyVar = varData
end


on getProperty me
  put pPropertyVar
end
```

Message Window:
```
setProperty sprite 1, "Property of sprite 1"
setProperty sprite 2, "Property of sprite 2"
```

getProperty sprite 1
or put (sprite 1).pPropertyVar
Result:
-- "Property of sprite 1"

getProperty sprite 2
or put (sprite 2).pPropertyVar
Result:
-- "Property of sprite 2"

Operators:

Character or word that acts or 'operates' on one or more elements.

Arithmetic Operators: (+, - , /, *)

Assignment Operators: (=, to, into)
Examples:

    x = 10
    set x to 10
     put 10 into x

Comparison Operators: (<, >, =, <>)

Logical operators: Combine Boolean values and produce a Boolean result. A Boolean value is TRUE (=1) or FALSE (=0). In Lingo, positive or negative integers is TRUE, 0 is FALSE. Operators include AND, OR, NOT.

Concatenation operators: For combining strings. These include & and &&.
&& combines the strings with a space in-between them.
For example:
put "dean" && "utian"
-- "dean utian"

Conditional Statements:

If else:

Testing and executing command based on the condition.

- If(condition) then Do something

  end if

Example:
if sprite(1).locV < 180 then sprite(1).locV = sprite(1).locV + 10
if sprite(1).locV > 90 then sprite(1).locV = sprite(1).locV - 10

- If(condition) then
          Do something
   else
          Do something else
   end if

Example:
if rollover(1) then
          member("message").text = "Activated"
else
          member("message").text = "Deactivated"
end if

Using or:
Example:
if rollover(1) or rollover(2) then
  member("message").text = "Member 1 or 2 selected"
end if

- If(condition) then
  Do something
 else if (condition)
  Do something else
 end if
 End if

Using *case:*
Multiple branching logic structure.

- case expression of
  value1 : statement
  value2 : statement
 otherwise:
  statement(s)
 end case

Example:
```
case the rollover of
        1: member("message").text = "Selected Sprite - 1."
        2: member("message").text = " Selected Sprite - 2."
        3: member("message").text = " Selected Sprite - 3."
otherwise
        member("message").text = "Selected Sprite - None."
end case
```

Loops:
For repetition of script / movie.

- For repeating instructions based on condition
```
repeat while (condition)
        statement(s)
end repeat
```

Example:
```
repeat while the mouseDown
        nothing
end repeat
```

- For repeating action number of times
repeat with localVariable = startValue to endValue
        statement(s)
end repeat

Example:
repeat with x = 6 to 10
        sprite(x).visible = 1
end repeat

- Exiting a repeat loop
        exit repeat

- Skipping a particular loop
        next repeat
Example:
repeat with x = 6 to 10
        if (x mod 2) = 1 then next repeat
        sprite(x).visible = 1
end repeat

Publishing Shockwave Movies:

- Convert files into Shockwave format (DCR).
- Publishing process: Compresses data in movie and optimizes it for web playback.
- EMBED Tag: Netscape Navigator Tag, supported by all shockwave compatible browsers.
- OBJECT Tag: Internet Explorer Tag, for Active-X controls support.

Code:
```
<HTML><HEAD>
<TITLE>Shockwave Tutorial</TITLE>
</HEAD>
<BODY>
<BODY BGCOLOR ="white">
<OBJECT WIDTH="320" HEIGHT="240">
<PARAM NAME="SRC" VALUE="Shock.DCR">
<EMBED SRC="Shock.DCR" WIDTH=320 HEIGHT=240 TYPE="application/x-director" PLUGINSPAGE="http://www.macromedia.com/shockwave/download/" >
</EMBED>
<NOEMBED><IMG SRC="alt_image.jpg" WIDTH=320 HEIGHT=240>
</NOEMBED>
</OBJECT>
</BODY>
</HTML>
```