Apéndice B

Introducción al lenguaje Lingo

Apéndice B. Introducción al lenguaje Lingo

B.1. Lingo y guiones

Lingo es el lenguaje de programación que usa Director. Este lenguaje constituye la herramienta más potente a la hora de realizar los proyectos multimedia, ya que se pueden programar acciones que en el entorno gráfico de Director no son posibles. Aunque en sus principios era un lenguaje muy simple orientado casi exclusivamente a controlar animaciones, actualmente es un lenguaje completamente orientado a objetos, pudiendo llegar a compararse en potencia de programación tanto a C y Pascal como, sobre todo, a los lenguajes orientados a Internet como el Java.

Lingo se concibió en las primeras versiones de Director como una forma de controlar su entorno, es decir, la Partitura (Score), el Reparto (Cast) y el Escenario (Stage); al escribir cualquier código en Lingo estamos creando un guión, y cada uno de estos guiones constituye un miembro del Reparto, como también lo son las imágenes o los vídeos. Concretamente, un guión es un miembro del Reparto que contiene una pieza de texto que es válida en código Lingo.

Una vez escrito, un guión se puede asociar a cualquier otro miembro del Reparto, como por ejemplo imágenes o sonidos, aunque también se puede situar en la Partitura, asociado a un Fotograma (Frame) determinado.

Este lenguaje tiene la particularidad de que casi se puede leer como si fuera el idioma Inglés, lo que lo hace más sencillo tanto de comprender como de aprender, ya que hoy en día el Inglés no debería suponer un obstáculo al nivel que nos movemos. Esta (relativa) facilidad de aprendizaje se debe a que todos los comandos, funciones y otras palabras claves en Director son palabras, grupos de palabras o abreviaciones en Inglés.

Para crear guiones se puede partir de varios puntos: partir de cero mediante la Ventana de Guiones (Script Window, figura 1), usar el Inspector de Comportamientos (Behavior Inspector, figura 2) para seleccionar un evento y una acción, o podemos usar un comportamiento ya elaborado desde la Paleta de la Biblioteca (Library Palette, figura 3), bien en su forma original o bien modificándolo. En las siguientes figuras se ven estas tres posibilidades. Sobre ellas intentamos señalar los botones más interesantes a la hora de escribir nuestros guiones.

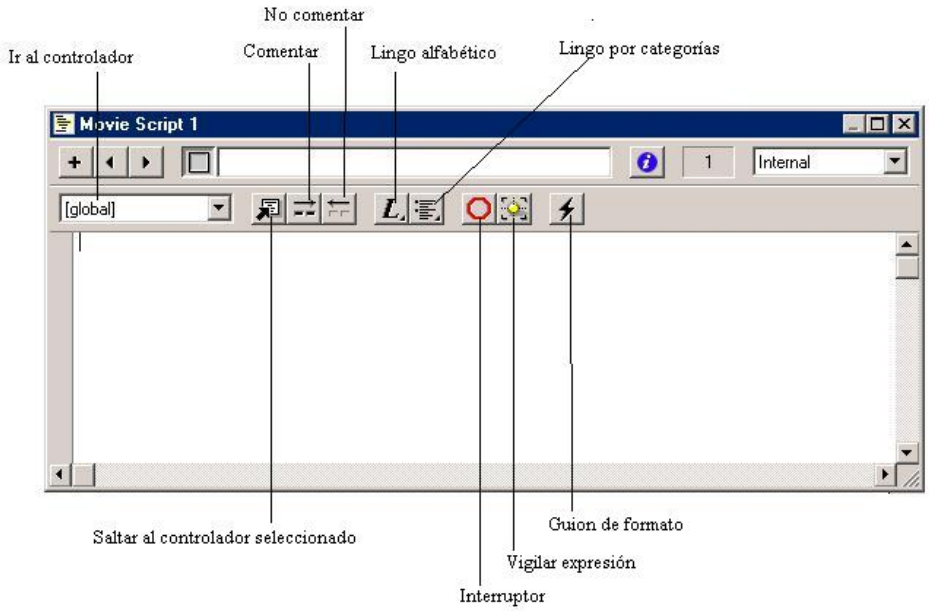

Figura 1. La Script Window

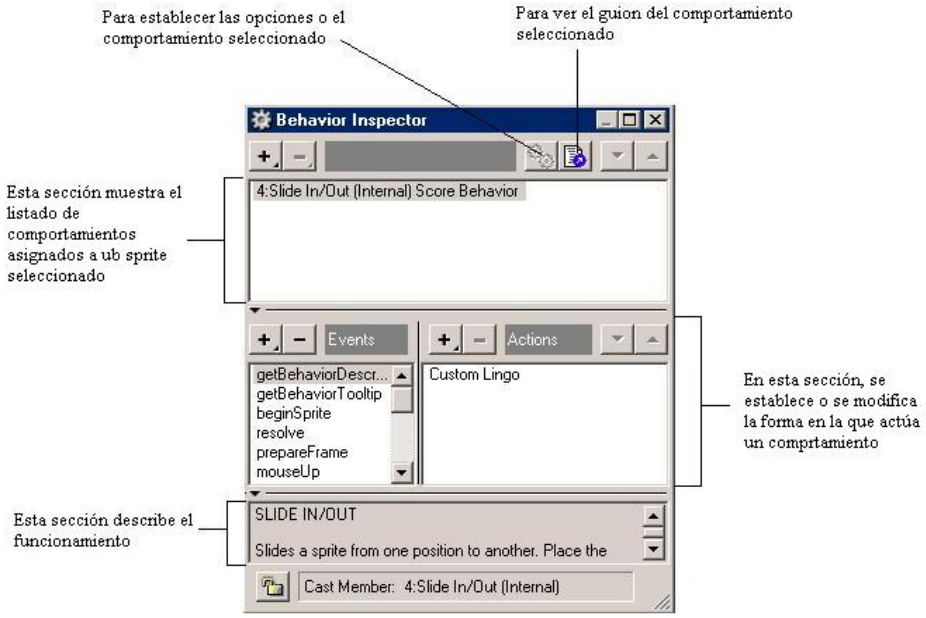

Figura 2. El Behavior Inspector

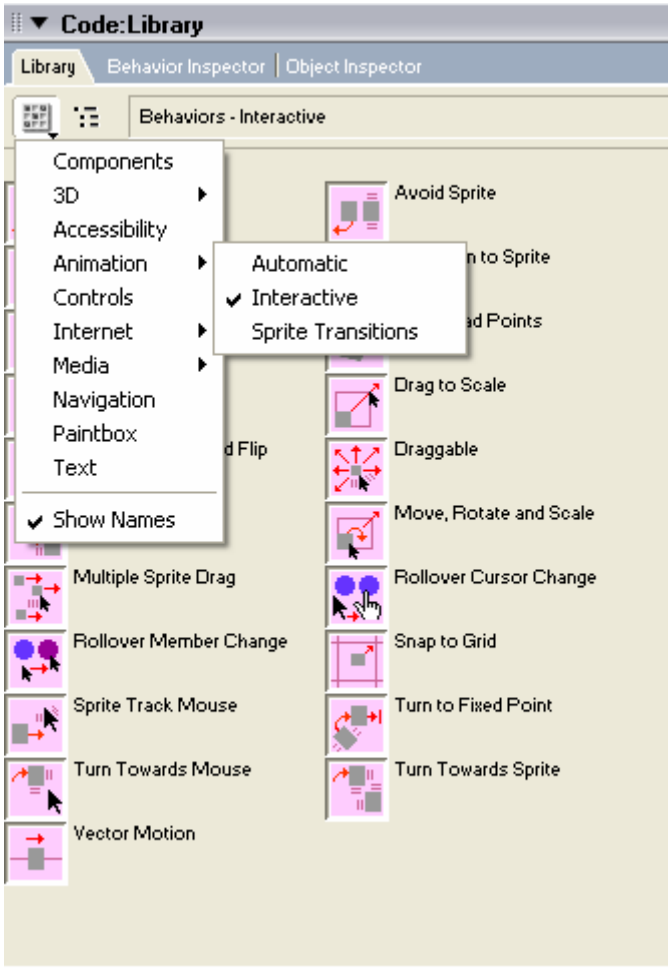

Figura 3. La Library Palette

B.2. Tipos de guiones

Respecto a los tipos de guiones de Lingo existen discrepancias entre distintos autores. Existe un grupo de éstos que consideran que las diferencias entre un guión y un comportamiento (Behavior) son meramente históricas, y que a partir de la versión 8 de Director dichas diferencias han desaparecido. Podría considerarse esto como cierto, ya que tanto guiones como comportamientos son instrucciones escritas en Lingo.

Sin embargo, los autores más detallistas consideran que sí hay diferencias, y exponen que todos los comportamientos son guiones, pero no todos los guiones son comportamientos. La principal diferencia es que los comportamientos deben escribirse según las reglas de la programación orientada a objetos, y debe tratarse como un objeto que se puede utilizar y personalizar de nuevo, dotado de parámetros que puedan ser configurados de forma independiente para cada instancia del mismo.

Esta segunda distinción parece más acertada para niveles avanzados de programación, que serán en los que nos movamos. Además, como se muestra en la figura 4., Director diferencia claramente a la hora de elegir el tipo de guión entre los tres que describiremos, entre los que se encuentra el behavior. Así, los guiones pueden ser de tres tipos: de película, de comportamiento o padre.

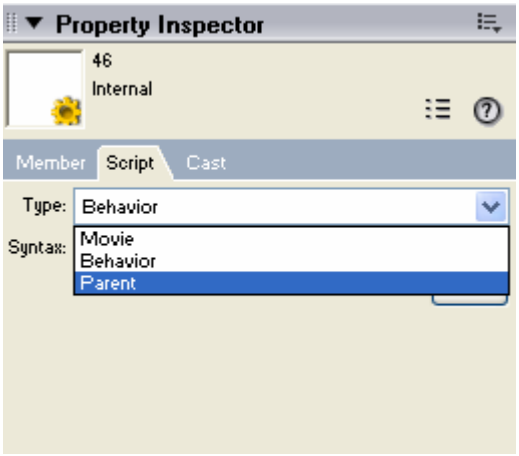

Figura 4. Property Inspector

Además, otros miembros del Reparto, como las imágenes de mapa de bits, o cualquier otro tipo de miembros, pueden tener guiones dentro de sí mismos. Estos son llamados normalmente guiones de Reparto.

La diferencia entre todos estos tipos de guión no está en qué aspecto tienen o cómo se comportan, sino en cuando actúan.

Un guión de película, por ejemplo, es una presencia global en una película. Si escribimos un guión de película que produce un pitido de sistema cuando se hace clic sobre un botón del ratón, este guión da el pitido cuando se hace clic sobre el ratón en cualquier momento. De ahí el nombre de "guión de película".

Un guión de comportamiento es similar en concepto a los comportamientos. No hace nada hasta que se sitúa en un sprite o en un canal de guión de fotograma. Si se sitúa sobre un sprite, los comandos Lingo dentro del guión están activos sólo en lo que al sprite se refiere. Si tiene un comportamiento que hace sonar un pitido cuando hace clic sobre el botón del ratón, y aplica ese comportamiento a un sprite, el pitido suena sólo cuando el usuario hace clic sobre ese sprite. Los guiones de comportamiento son a veces llamados guiones de Sprite o Partitura por esta razón.

Los guiones de comportamiento pueden ser asignados también al canal del guión de fotograma de la Partitura. Cuando se hace esto, actúan como guiones de película, pero sólo para el fotograma o fotogramas a los que son asignados. Los comportamientos usados de esta manera son llamados a veces guiones de Partitura.

Los guiones padre son un tipo de guión mucho más avanzado. No están asociados a ningún otro elemento, y funcionan como instancias de objetos independientes. Realmente no hacen nada hasta que usa algunos comandos Lingo de programación orientados a los objetos para decirles cómo y cuándo deben utilizarse.

Los guiones de Reparto, por otra parte, son fáciles de usar. Puede crearse uno seleccionando un actor, como por ejemplo una imagen de mapa de bits, y haciendo clic sobre el botón de guión en la parte superior de la Ventana de Reparto (Cast Window). Esto abre la Script Window y le permite añadir un guión a ese miembro del reparto en particular.

Los guiones de Reparto actúan solamente en ese miembro del Reparto. Si coloca un guión con un miembro del Reparto que hace al sistema soltar un pitido cuando el usuario hace clic sobre el ratón, esa acción afecta solamente a ese miembro del Reparto cuando está sobre el Escenario. Si usa a ese miembro del Reparto más de una vez en la Partitura, el guión con él esta activo en todos esos lugares. Sin embargo, no debe confundirse un guión de miembro de reparto con un comportamiento: un miembro de reparto puede aparecer muchas veces en la película, e incluso puede haber varias instancias de ese miembro al mismo tiempo en el escenario (cada una de estas apariciones será un “sprite”). El guión de miembro de reparto se aplica siempre a ese miembro, mientras que el comportamiento se aplica sólo a una aparición de este (es decir, a un sprite).

Para crear o modificar guiones se usa la Script Window; la importancia de esta ventana a la hora de programar en Lingo merece que expliquemos un poco sus partes (ver figura 1.).

La parte superior tiene los típicos botones de miembro del Reparto de Director que nos llevan hacia atrás y adelante en el Reparto, nos permiten añadir un nuevo miembro del Reparto, y nos permiten cambiar bibliotecas de Reparto. El resto de botones tratan con funciones más avanzadas de Lingo tales como Correctores, inserción de breakpoints (puntos donde se parará la ejecución del guión, para facilitar su depuración), recompilación del guión, etc.

Los dos botones llamados Alfabetical Lingo y Categorized Lingo nos permiten buscar e insertar automáticamente en un guión cualquier palabra clave de Lingo. Son muy útiles cuando no podemos recordar el nombre de un comando, o cuando no recordamos la sintaxis apropiada para usar comandos Lingo.

Por último decir que mediante el Property Inspector podemos cambiar el tipo del guión (figura 4.).

B.3. Controladores y eventos

Lingo está diseñado para responder a eventos en el entorno de Director. Prácticamente todas las acciones que tienen lugar en una película son eventos y mientras se reproduce una película de Director, éste emite notificaciones internas llamadas mensajes, indicando los eventos que han tenido lugar. Esto da lugar a que prácticamente podamos intervenir en cualquier momento de la película en función del tipo de evento que se produzca.

Los eventos pueden ser de varios tipos, como por ejemplo de reproducción (startMovie, enterFrame, etc.), de respuesta del usuario (keyDown, mouseDown, etc.), de tiempo (idle, timeOut) o de ventana (acivateWindow, openWindow, etc.). Una lista completa de los eventos pueden verse en el manual de Director o en la ayuda.

Los controladores de eventos o handlers (los llamaremos a partir de ahora sólo controladores) se pueden considerar como hojas de instrucciones dirigidas a realizar varias tareas. Los controladores están asociados a los eventos, es decir, cuando se produce un evento, actuará el controlador asociado a dicho evento.

La estructura de un controlador es la siguiente:

- Comienzan con la palabra on seguida por un evento.
- Incluyen una serie de acciones que han de producirse cuando el evento tiene lugar.
- Terminan con la palabra end.

Si un guión incluye varias instrucciones, se van ejecutando progresivamente una después de otra hasta llegar a final del controlador. Veamos un ejemplo sencillo:

```
on mouseUp me
    beep
end
```

Este controlador indica que cuando se pulse el ratón sobre el elemento al que está asociado, suene un pitido. Como puede apreciarse, puede leerse fácilmente el significado, pues

Lingo intenta imitar el idioma inglés (excepto por la palabra “me”, que es un parámetro de la función indicando que el objeto sobre el que se aplica la función es el mismo guión al que pertenece).

El programador puede fabricar sus propios controladores, que no respondan a ningún evento. En ese caso, al controlador también se le puede llamar función. Los controladores en ese caso pueden ir acompañados por argumentos, también conocidos como parámetros, los cuales representan valores suplementarios que se transmiten al controlador cuando se le invoca y que suministran la información adicional que necesita la función para completar la tarea para la que se programado. Estos parámetros se sitúan detrás del nombre de la función, como el parámetro “me” en el ejemplo anterior. Veamos un ejemplo en el que se ve fácilmente la utilidad de estos parámetros: al llamar a la función "sumaDosNumeros" e introduciéndole como parámetros esos dos números, nos devolverá su suma:

```
on sumaDosNumeros num1,num2
        suma=num1+num2
        return suma
end
```

Si ahora usamos la siguiente expresión en cualquier momento de la reproducción:

Variable = sumaDosNumeros(2,4)

nos devolverá 6, siempre que el guión esté disponible durante toda la película, es decir, que sea un guión del tipo película. Este método de crear controladores propios con parámetros tiene innumerables aplicaciones.

Ahora bien, a una función sólo se le puede llamar desde el ámbito desde el cual esa función es visible: desde la misma instancia del mismo objeto al que pertenece (bien sea un guión de comportamiento, o bien sea un guión padre). La única forma de tener una función a la que se puede llamar desde cualquier tipo de objeto es colocarla en un guión de película. Pero si queremos comunicarnos con una función que está en un objeto determinado y que no es visible, aún nos queda otra posibilidad: podemos mandar un mensaje hacia ese objeto.

Para terminar de hablar de los controladores y eventos, sólo hay que decir que a la hora de ejecutarse, los guiones siguen una jerarquía definida. Es decir, al producirse un evento

determinado, es importante conocer a qué controlador va asociado, y si tiene que ejecutar algún conjunto de instrucciones.

Supongamos que se produce un evento mouseDown (es decir, si se pulsa el botón izquierdo del ratón); el primer lugar en el que este evento se tendría en cuenta es en el sprite sobre el que se ha pulsado el ratón. Si dicho sprite tiene asociado un controlador con el evento en cuestión, lo recibe y realiza la acción correspondiente que le indique el controlador. Si por el contrario no tiene ningún comportamiento con un controlador asociado al evento mouseDown, continuará buscando un controlador receptivo.

El siguiente sitio donde mira es en los miembros del Reparto, es decir, chequea el Reparto para ver si algún controlador lleva asociado el evento mouseDown, y en tal caso lo ejecuta. Si tampoco lo encuentra aquí, busca en los guiones de fotogramas y finalmente, si esto también falla, se va a los guiones de película. Si sigue sin encontrarlo, desecha el evento y sigue con la reproducción de la película.

B.4. Elementos de Lingo

Las instrucciones incluidas en un controlador de Lingo están formadas por una combinación de elementos, de forma idéntica a las palabras de una frase en una lengua. Se pueden clasificar según su función y las reglas determinan cómo se pueden combinar.

Los elementos de Lingo se agrupan en las siguientes categorías:

- Comandos: Indican, en una película, una acción. Por ejemplo, *go to* ordena al cabezal de reproducción que se desvíe hacia un fotograma concreto.
- *Funciones*: Devuelven un valor. Por ejemplo, *date* devuelve la fecha establecida en el ordenador. Existen muchas funciones en Lingo, además de las que pueda escribir el programador.
- Términos *clave*: Son palabras reservadas que tienen un significado especial en Lingo. Por ejemplo, todas las propiedades de Lingo y muchas propiedades y funciones de los sprites deben ir precedidas por el término clave *the*. Este término clave distingue el nombre de la propiedad de una variable o del nombre de un objeto.
- Propiedades: Son atributos de un objeto. Por ejemplo, *the colorDepth* es una propiedad del monitor del ordenador referente a la definición del

color. Las propiedades también pueden venir expresadas con sintaxis de punto, como por ejemplo *lista.count* (número de elementos de una lista). En este caso, lista es un objeto, y count, una propiedad suya.

- Operadores: Son términos que modifican o cambian uno o varios valores. Por ejemplo, los operadores + o <.
- Constantes: Son elementos que no cambian. Por ejemplo, *TRUE* siempre tiene el mismo significado, el valor 1.
- Variables: Son fragmentos que se reservan en la memoria para guardar ciertos valores. Las dos cualidades más importantes de las variables son su tipo (entero, punto flotante, carácter, cadena de caracteres, lista, etc), y su ámbito. En cuanto al ámbito, las variables pueden ser locales (sólo accesibles en la función en que fueron declaradas), globales (accesibles en toda la película) y propiedades (accesibles por todas las funciones en una instancia de un objeto, bien sea comportamiento, bien guión padre).

B.5. Primeros pasos: la ventana Message

Director envía mensajes cada vez que tiene lugar un evento en la película (por ejemplo, pulsar el botón del ratón). A partir de los eventos, Director desencadena una serie de acciones que previamente se le han asignado mediante un controlador.

Estos mensajes son recogidos por Director en la ventana Message, que no es más que un intérprete de Lingo, es decir, toma comandos de Lingo y los evalúa de la misma forma que los guiones. Por tanto, haciendo uso de esta ventana podemos visualizar la forma de trabajar de Director. Esto supone una forma sencilla de comenzar a aprender el funcionamiento de Lingo y Director.

Tenemos dos posibilidades a la hora de trabajar con la ventana Message. Si el botón Trace está pulsado y comenzamos a reproducir nuestra película, comenzarán a aparecernos los distintos mensajes que envía Director, así como las líneas de guión que ejecuta; esta es una forma de visualizar lo que ocurre durante el transcurso de la película, ayudando notablemente en la realización de proyectos tanto a nivel de principiante como a nivel más avanzado. Por el contrario, si el botón Trace no está pulsado, podremos practicar tal como escribiríamos los

guiones en Lingo, para ver los errores que cometemos y poder subsanarlos. Esta es una forma sencilla de probar y mejorar guiones.

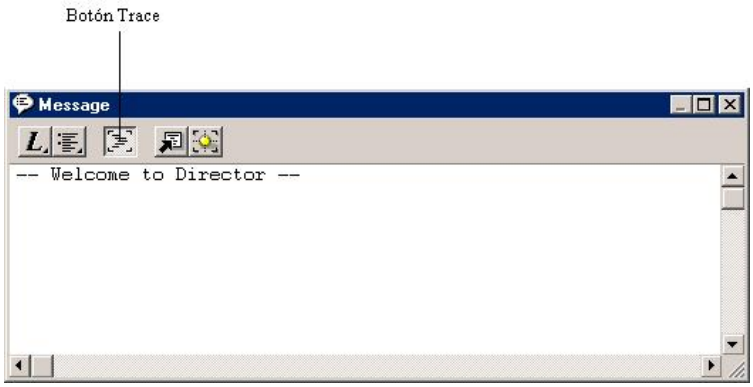

Figura 5. La Message Window

Para terminar, solamente comentar que el uso de la ventana Message es de gran utilidad para comenzar a aprender Lingo, y que es recomendable su uso para iniciarse en este lenguaje de programación.