



UNIVERSIDAD
POLITECNICA
DE VALENCIA

MetaCard: entorno de trabajo

Apellidos, nombre	Agustí i Melchor, Manuel (magusti@disca.upv.es)
Departamento	Departamento de Informática de Sistemas y Computadores
Centro	Universidad Politécnica de Valencia



1 Resumen de las ideas clave

MetaCard [1] (en adelante MC) es un ejemplo de herramienta de autor multiplataforma para desarrollos multimedia. Esta aplicación ha sido el germen de otras dos: *RunRevolution* y *LiveCode* [2]. Aunque sus sucesoras han crecido en el número de asistentes para realizar las más actuales tareas. Es esta una pieza muy didáctica que permitirá iniciarse en este tipo de herramientas que pertenecen a la metáfora de tarjetas y guiones. Por ello presentamos aquí los elementos centrales de su interfaz. Hay dos elementos a destacar de esta aplicación:

- La facilidad de uso del interfaz para el no iniciado. Dado que no hay muchas opciones a la vista, sino diferentes “ventanas especializadas” (paletas), que se irán abriendo cuando se requiera.
- La separación, desde sus inicios, de sus dos elementos centrales. Por una parte el motor (*MetaCard Engine*) y por otra el interfaz que constituye el entorno integrado de desarrollo (*MetaCard Integrated Development Environment* o MC IDE). Este segundo está enteramente escrito en *MetaTalk*, el lenguaje de script que soporta el motor y su código fuente es visible (y modificable) por el usuario.

Desde la aparición de *RunRevolution* (de *Runtime Revolution Ltd*) y la adquisición de MC, el motor ha ido siendo actualizado de forma independiente del interfaz. MC se ha mantenido disponible como entorno de desarrollo y está, desde entonces, disponible como *open source*. El entorno está formado por la barra de herramientas (*mctools.mc*), la ayuda (*mchelp.mc*) y la barra principal (*mchome.mc*). Para poderlo ejecutar hay que descargar el motor de *Revolution* o su última versión, conocida como *LiveCode*.

Puede descargar las más recientes versiones del entorno de desarrollo gratuitamente desde el *MetaCard IDE*¹, un grupo de trabajo albergado por *Yahoo!* para mantener y mejorar el IDE. También tiene copias de las versiones anteriores en nuestro servidor², que le permitirán instalar un sólo fichero (el *MetaCard Starter Kit*) para empezar a experimentar y las instrucciones para su instalación. Esta versión de MC es totalmente funcional con un límite de diez instrucciones en la longitud del código que se puede asociar a cada objeto. No tarde en instalarlo, que empezamos.

2 Objetivos

La finalidad de este artículo consiste en describir los elementos principales del entorno de trabajo para dar pie a la autopresentación, animada e interactiva, que la propia herramienta proporciona y a los tutoriales que incluye. Una vez que el alumno lo haya leído con detenimiento y experimente con las propuestas que realizamos, será capaz de:

- Identificar y familiarizarse con los elementos que componen el interfaz de MC.
- Navegar por la jerarquía de ventanas que constituyen el interfaz.

¹ La URL del grupo es <http://groups.yahoo.com/group/MC_IDE/>.

² En <http://web-sisop.disca.upv.es/~smm/public_xml/Software/MetaCard/indexMetaCard.html>.

Está a punto de iniciar un breve viaje por los elementos que componen MC para familiarizarse con ella, describir sus elementos y funciones y que conozca algunas posibilidades de la misma.

3 Introducción

Esta herramienta está en la línea de la que fue, posiblemente, la primera herramienta de autor: *HyperCard* (en adelante *HC*). Esta nació para plataforma *Macintosh* allá a finales de los setenta y sentó las bases de lo que con el tiempo serían las aplicaciones de carácter visual que impulsarían el desarrollo de aplicaciones con interfaces gráficas que han permitido a un gran número de personas acercarse al mundo de los computadores.

La forma de trabajar con esta herramienta está basada en el diseño de pilas de tarjetas, donde se describe explícitamente la apariencia de los objetos (tamaño, color, etc.) y su disposición en pantalla; no limitándose a trabajar en modo de pantalla completa. Para completarse con la programación (código que será interpretado por el soporte de ejecución) asociada que describe el comportamiento de los objetos y su interacción con el usuario.

El lenguaje de programación que soporta *MC* se llama *MetaTalk*, que forma parte del grupo de lenguajes *xTalk* (*HyperTalk*, *SuperTalk*, etc.), integrado por los que han definidos las herramientas derivadas de *HC*. Este lenguaje puede ser utilizado como lenguaje de *script* en aplicaciones que no necesiten un interfaz gráfico, como es el caso de la programación de acciones en servidores de páginas Web (CGI) o bien para automatizar procesos diarios en su máquina, sobre todo si necesita analizar contenidos de ficheros. Sí, sí, no es una novedad, pero ¿por qué no aprovechar que conoce un lenguaje para realizar otras tareas?. Además, en este caso, no hay limitación de instrucciones por objeto.

Para el recién llegado, deseoso de experimentar y dispuesto a conocer la aplicación: diríjase a la página web de MC y mientras la descarga le propongo que lea las siguientes líneas. En el transcurso de la edición de este documento la versión que se emplea es MC 2.4.3 sobre plataforma GNU/Linux, no hay impedimento alguno con utiliza otra versión y/o plataforma de trabajo.



Figura 1. El entorno básico de MC: presentaciones de las últimas versiones.

Por si acaso todavía no ha instalado MC, este es el momento. El orden en que se muestran los diferentes elementos que componen la aplicación es el mismo en que podría verlos aparecer en pantalla en cuanto ejecute su propia copia de MC,

lo primero que verá es la ventana de presentación que aparece mientras se realiza la carga de la aplicación y que puede verse en la fig. 1, para diferentes versiones de MC.

4 Desarrollo

Empezaremos con la parte visual, el interfaz desde la primera ventana que se abre al iniciar la aplicación (apartados 4.1 al 4.6): qué nuevas ventanas van apareciendo, cuál es su misión y su contenido. Terminaremos con el editor de código (apartado 4.6).

4.1 El motor de MC y el *Home*

Este es el corazón de la aplicación: el que interpreta el lenguaje MetaTalk. Es el soporte necesario, en tiempo de ejecución, de aquellas aplicaciones que cree Ud. En ese caso es cargado en memoria en el momento de iniciar el primero de sus componentes. El *Home* está concebido como una aplicación más, la que MC carga por defecto.

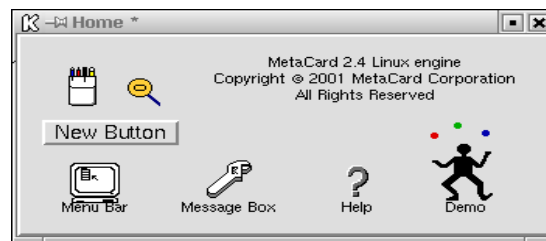


Figura 2. Mi MC: mi HOME, el HOME de MC.

Al cerrarse la ventana de presentación queda en pantalla la ventana que muestra la fig. 2, con cuatro iconos en su parte inferior, que vamos a visitar en un momento. Bueno, en realidad, lo que aparece en la figura es la apariencia que tiene mi versión de MC: puesto que se deja modificar al gusto del usuario, en la parte superior izquierda hay dos iconos de aplicaciones que uso habitualmente en mis desarrollos junto a un “New Button” que acabo de añadir sin ninguna funcionalidad ... todavía.

4.2 La barra de menú: *Menu Bar*

Lo usual, dentro del entorno es crear aplicaciones, para lo cual es casi inmediata la necesidad de abrir la paleta de herramientas, opciones y controles. Esta se sitúa en su propia ventana denominada *MenuBar*, que puede observarse en la fig. 3. Este es el gran centro de control de la aplicación, que no le despiste su simplicidad: hay muchas cosas serías detrás. No tenga la necesidad de aprenderse ahora mismo todas estas y de dónde aparecen, el uso hace esto la parte más fácil.

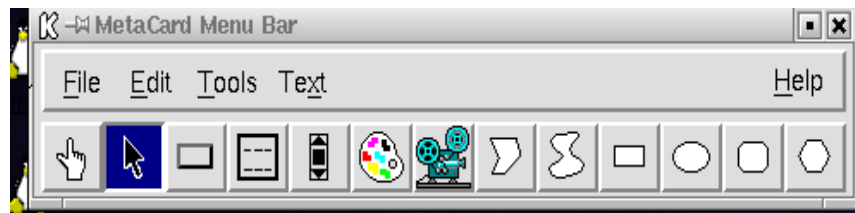


Figura 3. La MenuBar de MC incluye la barra de menús en la parte superior y la de herramientas en la inferior.

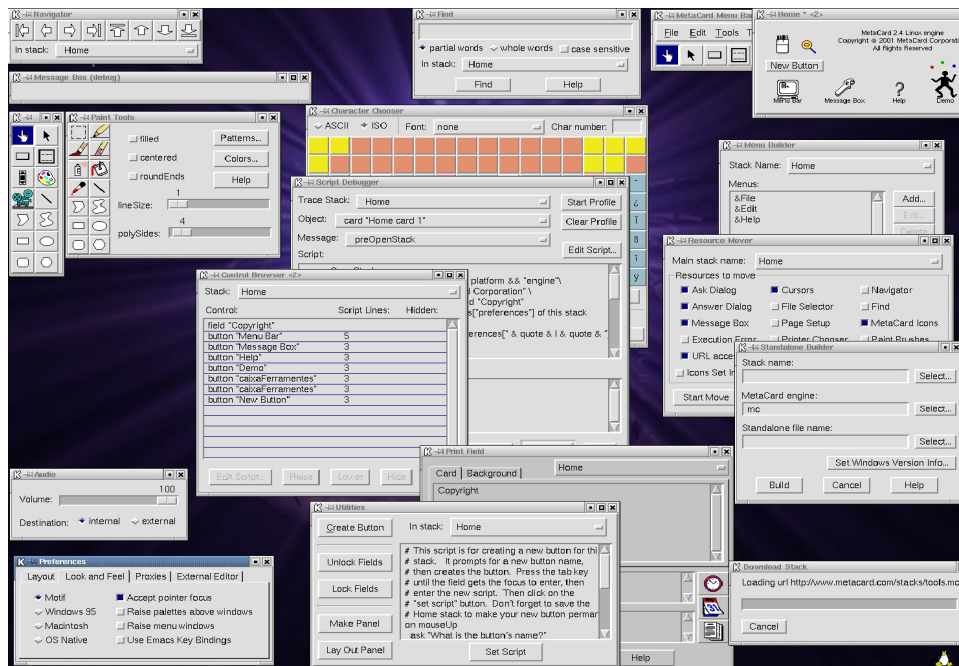


Figura 4. Las opciones de la entrada Tools de la MenuBar desplegadas.

En la parte superior está el menú del entorno de desarrollo, compuesto por:

- **File**, es el punto inicial para cargar una aplicación existente o para crear una nueva. Contiene las típicas opciones que ya habrá visto en otros programas. La curiosidad es quizá la opción de *Import...*, aunque no es una opción recomendable cargar los ficheros de imágenes, audio o vídeo con la aplicación cuando prolongan la carga de la aplicación. Con ella podrá fácilmente “recortar” un pedazo de la pantalla que aparecerá en la aplicación que está creando.
- **Edit** es todavía más previsible que el anterior, hace uso de la nomenclatura propia de MC así que de momento tampoco debe preocuparnos más, ya la irá visitando.
- **Tools** es la que oculta un mayor número de herramientas que ya se irán haciendo patentes con el uso. La fig. 4 las muestra todas desplegadas sobre el escritorio.
Entre todas las herramientas que componen la entrada *Tools* cabe destacar el *Control Browser*, que permite ver los controles o



componentes de una tarjeta de la pila escogida y nos informará de su estado: en cuanto a líneas de código, orden de apilamiento y si está visible o no. También para tener una visión global de la aplicación que se está construyendo o examinando, es recomendable usar la paleta *Components* a la que se puede acceder desde las propiedades de la pila.

- *Text* estará deshabilitada hasta que no haya creado un control de texto y está poniéndolo a prueba en ejecución, le permitirá decorar el texto, cosa que será útil en las aplicaciones que necesiten muchos efectos puntuales y visuales en estos componentes.
- *Help* permite acceder a los componentes de documentación que incorpora la propia aplicación. Únicamente mencionaría que la sintaxis del lenguaje y las órdenes (funciones y propiedades) disponibles, están siempre a mano en este apartado. No hay que memorizar, lo único necesario es perder el miedo a navegar por ella.

Los botones de la parte inferior de la “*Menu Bar*” ofrecen el acceso a los controles más usuales en el proceso de creación de una aplicación con MC. De izquierda a derecha los podemos agrupar en dos bloques. Los dos primeros permiten iniciar el modo de ejecución (**browse**) y pasar al modo de edición (**pointer**).

Los restantes están destinados a la construcción de los diferentes tipos de elementos que componen el interfaz de la aplicación de forma visual: **button**, para introducir botones; **field**, para crear un campo de `texto`; **scrollbar**, que crearía una barra de desplazamiento; **image**, es un contenedor para imágenes de tipo mapa de bits; **player**, es un contenedor de propósito variado donde se puede asignar desde una secuencia de vídeo a un fichero de audio y las primitivas gráficas en los últimos seis controles, tipo **graphic**, que debido a la abierta naturaleza de este tipo de gráficos vectoriales lo que se proporciona es una serie elementos característicos como *polygon*, *curve*, *rectangle*, *oval*, *roundrect* y *regular*.

4.3 La caja de mensajes: *Message Box*

Esta es la herramienta imprescindible para trabajar con MC. A pesar de su simpleza de apariencia que se puede ver en la captura que muestra la fig. 5. En su interior, un campo de texto espera a que se teclee una línea de caracteres. De forma que al pulsar *Enter*, se ejecutará por MC esta línea como si fuese una del código de un objeto de una aplicación en ejecución. Existe un “histórico” de órdenes tecleadas, así que con el cursor puede recuperar y editar rápidamente una orden anterior.

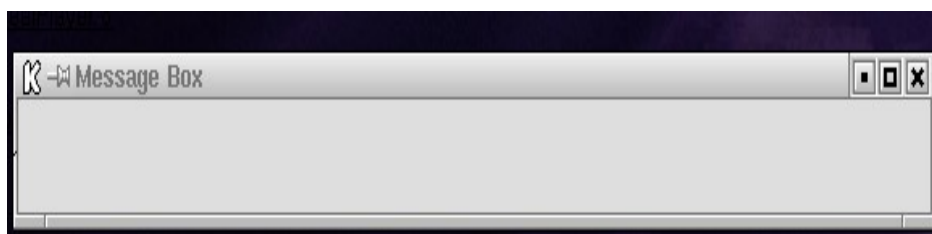


Figura 5. *Message Box*: el banco de pruebas rápido y el lugar más cómodo donde dejar mensajes mientras se desarrolla.

¿Qué se ejecuta desde ella? Cualquier orden de *MetaTalk* en cualquier momento. Sólo cabe recordar que no hay ningún recuerdo de lo que ha ejecutado en la orden anterior, así que las variables tienen una corta existencia. Por ese motivo es un sitio ideal para, por ejemplo:

- Comprobar la sintaxis o el funcionamiento de una orden sin necesidad de crear una aplicación para ello: put "Hola, Mon!"
- Acciones puntuales: move this stack rel 0,30

Observe que la acción realizada se lleva a cabo sobre la ventana activa, por lo que facilita la comprobación o actuación sobre los elementos de la aplicación que se está construyendo

- Comprobaciones o acciones rápidas y puntuales de órdenes que involucran el uso de variables put "hola, Mon!" into varAux; put varAux
- Acciones repetitivas de ordenación de parámetros de definición de una serie de objetos en pantalla: repeat with i = 1 to the number of fields; set the height of field i to 20; end repeat
- Alterar la secuencia habitual de mensajes que desencadena una aplicación para poder editar y corregir un comportamiento: lock messages; go to next card
- Tiene todo el potencial de MC, así que puede hacer acciones "algo más complejas: answer information "Tria un fitxer"; put url ("file:" & it)

Por supuesto, si es necesario la puede incluir en su aplicación, como una componente más o reinventándola. ¿Cómo? Simplemente es un campo de texto, cuyo comportamiento consiste en el mecanismo para guardar el histórico de órdenes, más la ejecución del texto introducido mediante la orden **do**.

4.4 La ayuda y la documentación: *Help*

La ayuda está condensada en un pequeño panel de opciones que se muestra en la fig. 6a y que permite acceder a una panoplia de "armas" que le vendrá bien tener a mano durante sus desarrollos.

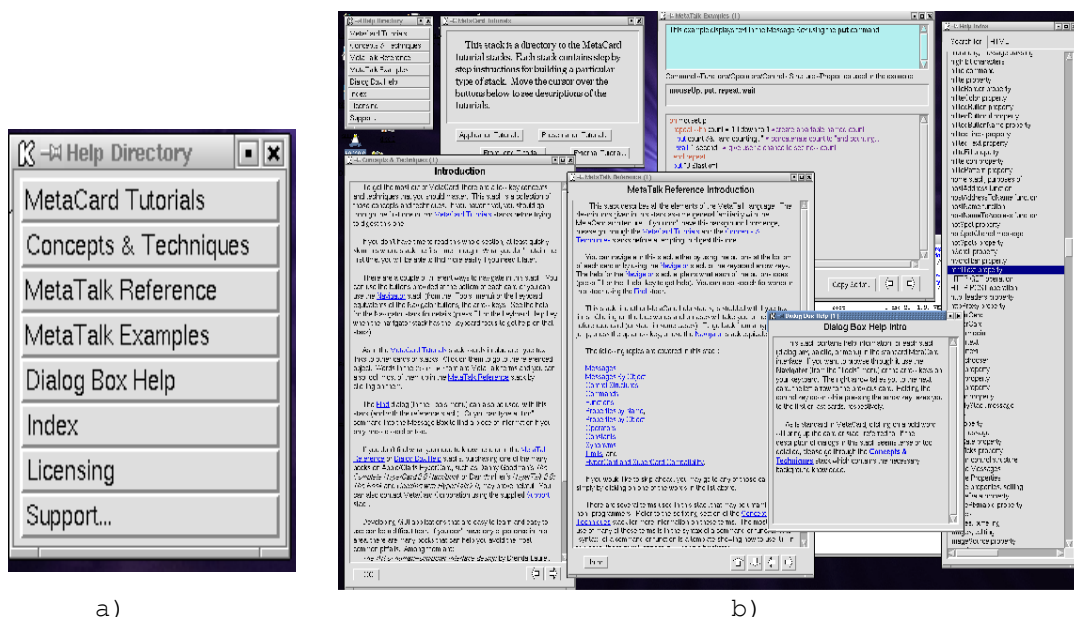


Figura 6. La pila de ayuda (a) y los contenidos de esta desplegados (b).

Los contenidos de este “directorío”, como reza en el título de la ventana y que puede ver en la fig. 6b, son: los tutoriales de MC para empezar dentro de la propia aplicación; la guía del usuario, *Concepts & Techniques*; el manual del programador o *MetaTalk Reference*; los ejemplos de utilización en el *MetaTalk Examples*; *Dialog Box Help* guarda la explicación de cada una de las herramientas y paletas incluidas en MC; *Index*, que permite la opción de búsqueda ordenada de funciones, propiedades, órdenes y eventos.

De todos estos elementos, cabe destacar la importancia de poder disponer siempre a mano del índice. Este muestra, ordenados alfabéticamente, los nombres de funciones, propiedades y eventos disponibles en el lenguaje y a los que se puede acceder de forma manual o introduciendo lo que se anda buscando y dejando que sea MC el que actualice una lista de contenidos que se corresponden con la cadena de texto que se está tecleando. A partir de aquí puede acceder a la página de ayuda correspondiente. Desde esta se puede comprobar su sintaxis, significado, algún ejemplo y, lo más interesante, órdenes y otros elementos relacionados con el que ha dado pie a la búsqueda.

4.5 La “demo”

La tarjeta de visita, la “demo”, es una sencilla clase de carácter ameno que le guiarán por los contenidos de esta herramienta y por la forma de trabajar con ella, sin perder tiempo en la realización de los elementos de naturaleza visual que utiliza. En cuanto pueda, ejecute por lo menos la primera parte de esta demostración, la fig. 7 muestra algunas instantáneas de la misma. Recomiendo disfrutar de la demostración en directo, así conocerá a la mascota y más cosas.

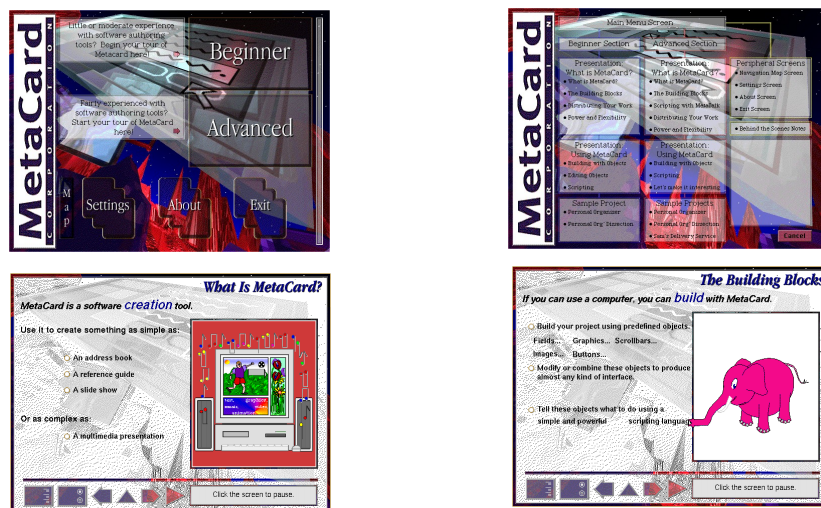


Figura 7. La “demo” en ejecución. Capturas del índice y del mapa de contenidos de la aplicación (en la primera fila) y dos ejemplos de la ejecución (en la segunda fila).

Esta demostración está compuesta por dos grandes apartados: *Beginner* para aquellos que se acercan a MC por primera vez o bien aquellos que tengan poca experiencia en el manejo de una herramienta de autor y *Advanced* para los que ya han utilizado una herramienta de este estilo. Ambas aproximaciones permiten ver:



- Qué es MC en tanto a qué elementos lo componen, cómo se puede distribuir las aplicaciones realizadas con ella y unos apuntes sobre sus capacidades.
- Cómo realizar una presentación utilizando esta herramienta; desde el uso de los controles que incorpora hasta una breve introducción al lenguaje de programación que incorpora.

Cuando tenga un poco más de tiempo véala al completo, le mostrará incluso algunos de los secretos (bueno, en realidad si quisieran que lo fueran no los contarían, ¿verdad?) y de cómo echarle un vistazo al código de la misma.

4.6 El editor de código

El *Script Editor* es la herramienta para la edición de código dentro del entorno de MC. Se puede acceder a él desde las propiedades de cualquier objeto y también desde el *Control Browser*. La fig. 8 muestra un ejemplo del mismo en acción mostrando el código de la *Menu Bar* que ha sido ejecutado desde la *Message Box* con la instrucción:

edit the script of stack "MetaCard Menu Bar"

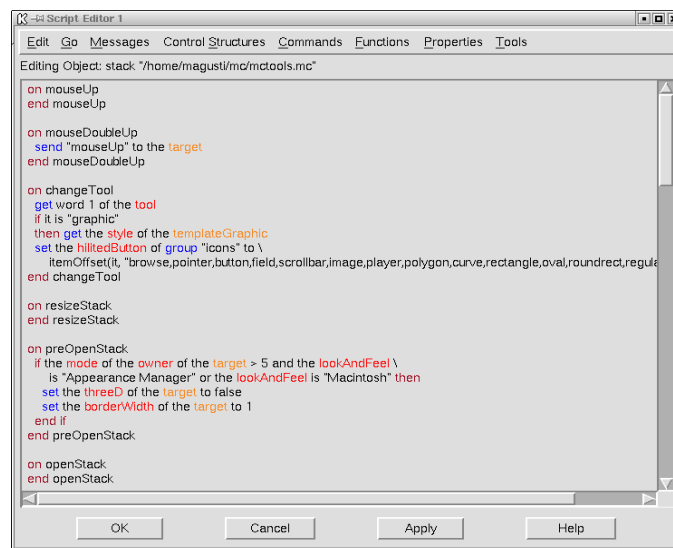


Figura 8. El editor integrado de MC con la característica de colorear el código activa (Tools\Colorize Script).

Las opciones del editor, agrupadas en secciones, son las propias para retocar y facilitar la inserción de código. *Edit*, para las típicas operaciones de deshacer, cortar, pegar, seleccionar y las no tan usuales pero bastante útiles de comentar o descomentar una selección del texto. *Go*, permite ir a una línea y, cuando haya código, avanzar directamente hasta la línea donde empieza el manejador que elija de la lista (construida en tiempo de ejecución). *Messages* que ejecuta la pila *MetaCard card Messages Menu* que ofrece una lista de posibles eventos de *MetaTalk* que se pueden definir en el punto de inserción actual. *Control Structures*, facilita la escritura de estructuras propias de *MetaTalk* (*exit*, *if-then*, *next*, *pass*, *repeat*, *return*, *switch* y *try-catch*). *Commands*, *Functions* y *Properties* proponen (mediante la ejecución de las pilas *MetaCard Commands Menu*, *MetaCard Functions Menu* o *MetaCard stack Properties Menu*) una lista de, respectivamente, órdenes, funciones y propiedades que se pueden utilizar

en el punto actual de edición del código. *Tools*, reúne una serie de herramientas de las que le recomiendo sobre todo la de *Colorize Script*. La opción de colorear el código facilita su lectura y, también, la detección de errores en la escritura del mismo, para ello se utiliza la codificación que guarda la *Menu Bar*: "blue" para las instrucciones, "brown" para funciones globales, "DarkOrange" para las componentes de las estructuras de control y "red" para las propiedades.

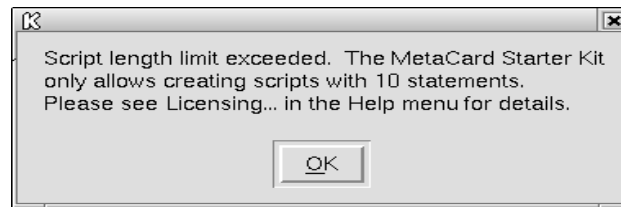


Figura 9. Mensaje de aviso del Starter Kit: Solo se pueden utilizar 10 instrucciones en el código de un objeto.

En cualquier momento de la escritura del código se puede hacer uso de los botones que hay en la parte inferior de la ventana del editor. **OK**, analiza sintácticamente el código y, en caso de ser correcto, lo actualiza en el objeto y cierra la ventana del editor. Si hubiesen errores aparecería una pequeña ventana contándole algunas cosas al respecto como (casi) el quién y el dónde se ha producido el primer error. **Cancel**, cierra la ventana sin analizar ni actualizar el código. **Apply**, es el más útil en el proceso de edición, por que como el OK comprueba y actualiza, pero no cierra el editor. **Help**, permite acceder a la ayuda al respecto del propio editor.

Puede que se encuentre con un mensaje como el de la fig. 9, por ejemplo al querer modificar el código de la pila *Menu Bar*, como se ha hecho para obtener la fig. 8. Recuerde que estamos trabajando sobre la versión *Starter Kit* y este es el aviso de que estamos intentado asignar (con *Apply* u *OK*) un código de más de diez instrucciones (que no líneas de código) a un objeto. Lo que no impide es su inspección y el aprendizaje que se puede derivar de esta actividad.

5 Conclusión

Después de realizar las acciones indicadas en este documento, el lector puede moverse en el entorno de trabajo de MC. Para que sea útil este contenido, ahora el lector ha de ser activo: vea la *demo* otra vez y láncese a realizar los dos primeros tutoriales que acompañan a esta herramienta. Le van a guiar paso a paso y, siempre, desde dentro del entorno. Durante su realización será consciente de la libertad que ha adquirido y de cómo va asimilando dónde están las cosas sin necesidad de memorizarlas.

6 Bibliografía

[1] MetaCard. Disponible en <<http://www.metacard.com/>>

[2] Run Revolution Ltd. Disponible en <<http://www.runrev.com/>>.