



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

# Controles audiovisuales de *MetaCard* para el desarrollo de aplicaciones multimedia

<b>Apellidos, nombre</b>	Agustí i Melchor, Manuel (magusti@disca.upv.es)
<b>Departamento</b>	Departamento de Informática de Sistemas y Computadores
<b>Centro</b>	Escola Tècnica Superior d'Enginyeria Informàtica Universitat Politècnica de València



## 1 Resumen

En este artículo y desde la perspectiva de una herramienta de autor multimedia como es *MetaCard* [1], se va a explorar el uso de los elementos que provee para la utilización de imagen, audio y vídeo.

## 2 Objetivos

El objetivo central consisten en exponer la forma de uso de los elementos multimedia diferentes del texto, que provee *MetaCard*, esto es:

- Los estáticos como una imagen de tipo mapa de bits, utilizando el objeto *image*.
- Los media dinámicos, como el audio y el vídeo, soportados por el reproductor incorporado, en su versión más simple (con la orden *play*) o bien utilizando el objeto *player* y sus propiedades y órdenes asociadas.

En este documento se exponen pequeños listados de código relacionados con las diferentes órdenes que se exponen para dar pie al usuario a que investigue su uso, ejecutando la aplicación, probando el código y consultando ayuda incluía dentro de la propia herramienta. No espere largas y detalladas explicaciones del código: sea activo, pruébelo.

## 3 Introducción

En el presente documento ilustrará con breves ejemplos el desarrollo de partes de aplicaciones que utilicen estos recursos multimedia, atendiendo a las características multiplataforma que tiene *MetaCard*.

## 4 Mapas de bits (*image*)

A la hora de manejar representaciones visuales estáticas , en principio, se puede recurrir al uso de imágenes en su sentido más tradicional dentro del ámbito de los computadores: como un conjunto de puntos o *mapa de bits*.

Es posible llevar a nuestra aplicación diferentes contenidos en formato de mapa de bit sin más que crear un contenedor de tipo "image" y asignarle un fichero desde su caja de propiedades (fig. 1), o desde el código (véase listado 1) utilizando la propiedad **fileName**, que puede consultarse y modificarse como cualquier otra propiedad de un objeto con la orden **set;** volcar (con **put**) una imagen desde un fichero (o URL) sobre este contenedor o capturar una zona de la pantalla. Además es posible llevar a fichero los contenidos de uno de estos controles con la orden **export**.

### #Desde fichero

set the filename of image "nombreDelObjeto" to "rutaHastaLaImagen"

### # Desde una URL

put "http://laURL/de\_la\_imagen" into image "nombreDelObjeto"

### #Directamente de la pantalla

import into image "nombreDelObjeto"

Listado 1: Ejemplos de asignación de contenido a un objeto de tipo mapa de bits.

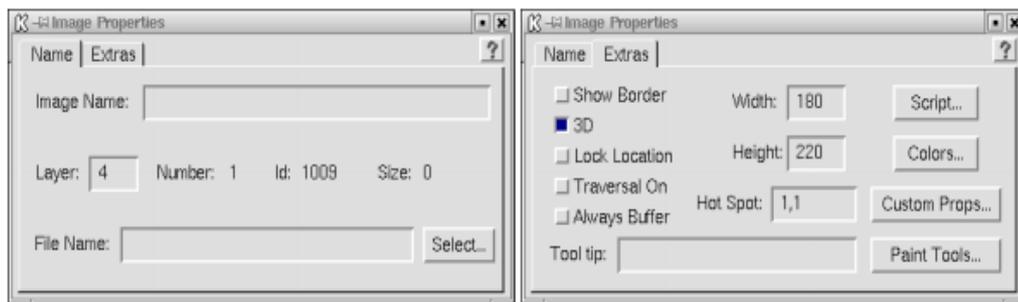


Figura 1: Aspecto y propiedades de un objeto tipo "image" (barra de desplazamiento).

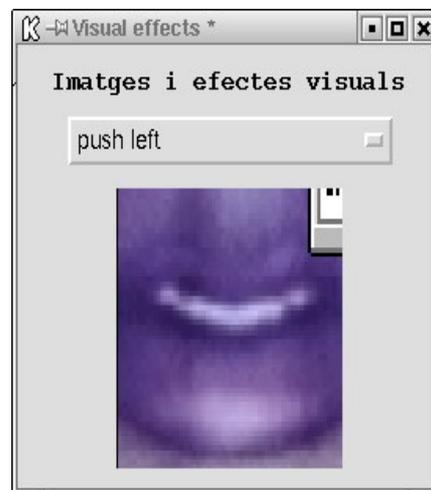


Figura 2: Las operaciones típicas sobre un objeto de tipo "image".

Así, es posible poner una imagen en pantalla, cargándola desde un fichero y mostrándola con algún efecto de transición de los que proporciona la orden **visual effect**. ¿Se lo imagina? La fig. 2 le propone un posible interfaz y el código de los controles asociados está desglosado en el listado 2). Todo empieza cuando el usuario pulsa sobre el objeto imagen, permitiendo que se cambie la imagen mostrada y cuando se escoge un efecto desde el botón



desplegable se aplica a la imagen mostrada. Se ha pretendido con ello que sea posible la experimentación de lo expuesto. De hecho el código de la tarjeta está ahí únicamente con la idea de ilustrar cómo se puede inicializar el contenido de uno de esos objetos (un botón) para un caso similar.

```
#Objeto imagen
on mouseUp
  answer file "Tria una imatge"
  put "(1)" & it && "(2)" & the result
  if not(the result is "Cancel")
  then
    set the fileName of me to it
  end if
end mouseUp

#Botón desplegable
on mouseUp
  hide image 1
  -- wait 10
  do "show image 1 with visual effect" && the effective label of button 1
end mouseUp

#La tarjeta
on openCard
  set the text of button 1 to "barn door close" & return & "barn door open" & return &
  "checkerboard" \
    & return & "dissolve" & return & "iris close" & return & "iris open" \
    & return & "zoom out"
  # Hay más efectos, pero abreviaremos aquí
end openCard
```

Listado 2: Código del ejemplo de operaciones típicas sobre un objeto de tipo "image".

Venga, hagamos algo un poco más difícil ... o quizá no. Hagamos una versión básica de una aplicación al estilo álbum de fotos, protector de pantalla o catálogo de productos que se basan principalmente en mostrar contenidos visuales. Se deja a la imaginación del lector crear una de estas aplicaciones que



muestre una imagen y por lo menos un texto asociado (el nombre del fichero), de forma que la cantidad de estos elementos no esté fijado de antemano.

¿Ya lo tiene? Como ejemplo de partida se plantea el del listado 3. En este se muestra el código que toma del directorio actual los nombres de los ficheros de imágenes que encuentra y muestra, de forma aleatoria, uno de ellos. La primera parte debería realizarla la pila en el momento de inicializarse y la segunda parte (la que contiene el ejemplo) la realiza el propio objeto de tipo "image". La última instrucción es únicamente para mostrar un sencillo mensaje de confirmación. Se han evitado las cuestiones referidas al control de errores para facilitar la lectura en el momento actual.

```
#En la pila
global llistatFitxers

on openCard
  put the files into llistatFitxers
end openCard

on mouseUp
  set the fileName of me to any line of llistatFitxers
  put the filename of me
end mouseUp
```

Listado 3: Código del visor de imágenes básico.

```
# Control de tipo image
on mouseUp
  if the repeatCount of me is 0
  then
    set the repeatCount of me to -1
  else
    set the repeatCount of me to 0
  end if
end mouseUp
```

Listado 4: Código del visor de imágenes con "animaciones" (GIF89a).

Para cerrar este apartado, cabe mencionar que es posible manejar una imagen en formato "GIF89a". Esto es, actuar sobre la cadencia en que se muestra la secuencia de imágenes que contiene un fichero de este formato. Para ello se utilizan, entre otras, las propiedades **currentFrame**, **frameCount**, **constantMask**, **palindromeFrames** y **repeatCount**. El listado 4 muestra un



ejemplo de uso de estas propiedades: en concreto, permite la visualización de la secuencia de imágenes que contiene un fichero de tipo GIF, o pararlo si está en "marcha".

## 5 Vídeo y sonido (*play* y *player*)

El mecanismo básico de utilización de archivos de audio y animación está soportado por la instrucción **play**, Para ver el funcionamiento de este objeto recurriré a un extracto del ejemplo didáctico, que sobre *MetaCard* realizó A. Serrano y sobre el que añadido algunos detalles. La fig. 3 muestra una sencilla botonera para controlar el vídeo que se reproduce en el centro de la tarjeta.



Figura 3: Utilización de "play VideoClip".

### # La tarjeta

```
global nomvideo
on preOpenCard
  hide group aviso
  put " " into field vol
  put "video1.avi" into nomvideo
  put " " into field instruccion
end preOpenCard

on closeCard
  play stop videoClip "videos/" & nomvideo
end closeCard
```

Listado 5: Asignación del vídeo a la apertura de la tarjeta.



### **# Botón "play"**

```
global nomvideo
on mouseUp
  play videoClip "videos/" & nomvideo
  put "play videoClip <nombre>" into field instruccion
end mouseUp
```

### **# Botón "looping"**

```
global nomvideo
on mouseUp
  play videoClip "videos/" & nomvideo looping
  put "play videoClip <nombre> looping" into field instruccion
end mouseUp
```

### **# Botón "stop"**

```
global nomvideo
on mouseUp
  play stop videoClip "videos/" & nomvideo
  put "play stop videoClip <nombre>" into field instruccion
end mouseUp
```

### **# Botón "pause"**

```
global nomvideo
on mouseUp
  play pause videoClip "videos/" & nomvideo
  put "play pause videoClip <nombre>" into field instruccion
end mouseUp
```

### **# Botón "resume"**

```
global nomvideo
on mouseUp
  play resume videoClip "videos/" & nomvideo
  put "play resume videoClip <nombre>" into field instruccion
end mouseUp
```



```
# Botón "back"
global nomvideo
on mouseUp
  play step back videoClip "videos/" & nomvideo
  put "play step back videoClip <nombre>" into field instruccion
end mouseUp

# Botón "forward"
global nomvideo
on mouseUp
  play step forward videoClip "videos/" & nomvideo
  put "play step forward videoClip <nombre>" into field instruccion
end mouseUp
```

#### Listado 6: Operaciones sobre el vídeo con la orden **play**.

Obsérvese que el nombre del fichero de vídeo se ha fijado en el código, pero no se ha importado en la pila, si no que se incorpora desde el sistema de archivos en disco en el momento de la ejecución, lo que permitirá cambiarlo entre dos ejecuciones diferentes. Esto se muestra en el listado 5.

Los controles específicos del vídeo son implementados con una instrucción que es copiada en un campo de mensajes para apoyar la parte didáctica de la aplicación. Se muestran en el listado 6.

Ah, permítame una observación: las barras ("/") aseguran que el producto final es multiplataforma. La utilización de rutas relativas permite también un fácil transporte de la pila de una ubicación a otra en el sistema de archivos de la máquina. Incluso llevarlo a otro sistema operativo.

El uso de un fichero de audio se realiza de forma análoga, indicando a la orden **play** el parámetro **audioClip**. La única diferencia con el ejemplo anterior de **videoClip**, es que no tiene una referencia visual como es el caso del vídeo.

La complejidad de los formatos y los algoritmos de compresión que utilizan la mayoría de estos archivos, junto a la necesidad de mantener el tamaño del motor de una herramienta de autor en unos valores mínimos hacen recomendable poder descargar esta tarea en aplicaciones externas.

Así, con el devenir de las versiones de MetaCard hemos visto aparecer el objeto "player", que permite dejar esta tarea en manos de aplicaciones específicas: *Xanim* en plataforma *Unix*<sup>™</sup>, *Media Player* o *QuickTime* en *Windows*<sup>™</sup> y *Macintosh*<sup>™</sup>. Con lo cual ahora es un control más, que dispone de una barra de control estándar, pero sobre todo que se puede mover, ocultar, hacerlo transparente, ... y recibir eventos como cualquier otro control.

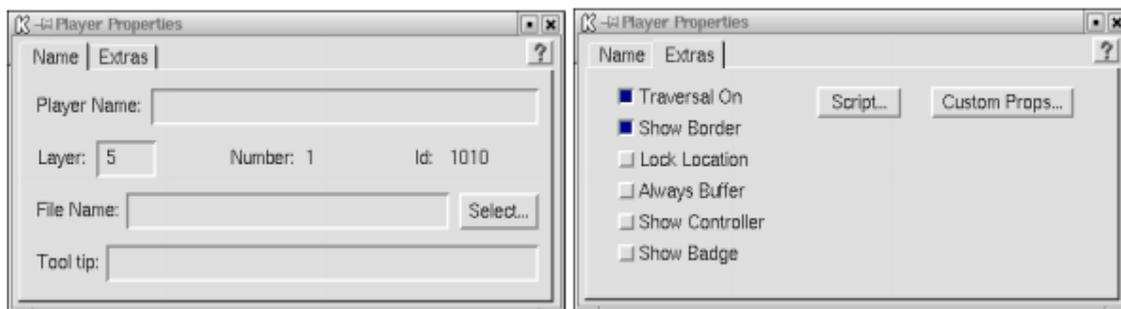


Figura 4: Aspecto y propiedades de un objeto tipo "player".

```
# Botón
if quinBoto es 1
then
  set the filename of player "x" to fld "y"
  start player "y"
else
  stop player "y"
end if
```

Listado 7: Código del ejemplo del objeto "player".

La definición básica de este contenedor la podemos ver en la figura 4. Trabajar con estos controles es tanto o más sencillo que con la instrucción **play**. Ahora tenemos la posibilidad de fijar su posición en una tarjeta y los controles son más precisos. Un ejemplo simple de aplicación que dispone de un campo de texto (de nombre "y", un **player** (llamado "x") y un botón, cuya pulsación pone en marcha o para la reproducción del vídeo. En función del botón utilizado como se muestra en el listado 7.

## 6 Conclusiones

A lo largo de este documento hemos visto el uso de los controles más recurridos a la hora de realizar aplicaciones multimedia. Así hemos abordado el uso de imágenes estáticas (en mapas de bits) y el del audio y vídeo en *MetaCard*.

La exposición de sus tipos y modos de operación se ha acompañado de ejemplos breves para que el lector haya podido llevarlos a cabo sin mucha demora en su elaboración.

## 7 Bibliografía

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