



UNIVERSIDAD
POLITECNICA
DE VALENCIA



**MASTER EN COMUNICACIONES Y
DESARROLLO DE SERVICIOS MÓVILES**
2009-2010

INTERFACES GRÁFICAS MULTIMEDIA
Filosofía de ActionScript





Contenido



MUCOM 2009-2010

- Metodología Orientada a Objetos
- Independencia de la plataforma
- El recolector de basura
- Rendimiento
- Seguridad
- El formato SWF

INTERFACES GRÁFICAS MULTIMEDIA



Este tema versa sobre la filosofía de ActionScript, un lenguaje de programación muy parecido a Java en la actualidad. Como veremos, los principios básicos son esencialmente los mismos salvando ligeras diferencias.



Contenido



MUCOM 2009-2010

- Metodología Orientada a Objetos ►
- Independencia de la plataforma
- El recolector de basura
- Rendimiento
- Seguridad
- El formato SWF

INTERFACES GRÁFICAS MULTIMEDIA





Metodología Orientada a Objetos



MUCOM 2009-2010

- **Datos y código se combinan en objetos**
Un objeto contiene estados (propiedades) y comportamientos (métodos).
- **Escalabilidad: reutilización del código**
Mejora la gestión del software (grandes proyectos).
- **Problemas:**
El diseño de objetos realmente genéricos es complicado.
La reutilización normalmente va acompañada de modificaciones.
- **AS1 era un lenguaje basado en objetos**
AS2 permite programación tanto orientada como basada en objetos.
AS3 está totalmente orientado a objetos.

INTERFACES GRÁFICAS MULTIMEDIA



La metodología OOP se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para explicar este concepto, una primera idea es diseñar el software de forma que los distintos tipos de datos que use estén unidos a sus operaciones. Así, los datos (propiedades) y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene “comportamiento” (el código) y “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de software complejo. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando así su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas que permitan la reutilización del código entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo a una reducción drástica del tiempo de desarrollo.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y a menudo falta una metodología para conseguir una reutilización eficiente – muchas veces se acaba por modificar las clases de un proyecto a otro. Algunas comunidades Open Source ayudan en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos y librerías reutilizables.



Contenido



MUCOM 2009-2010

- Metodología Orientada a Objetos
- **Independencia de la plataforma ►**
- El recolector de basura
- Rendimiento
- Seguridad
- El formato SWF

INTERFACES GRÁFICAS MULTIMEDIA





Independencia de la plataforma



MUCOM 2009-2010

- **La portabilidad se consigue gracias al bytecode**
Código a medio camino entre el código fuente y el código máquina.
- **La máquina virtual interpreta el bytecode**
Está escrita en el lenguaje nativo del dispositivo (el cual entiende su hardware).
Programas algo más lentos.



INTERFACES GRÁFICAS MULTIMEDIA



Los programas escritos en ActionScript se ejecutan exactamente igual en cualquier tipo de plataforma. Es lo que significa ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java: *write once, run everywhere*.

La portabilidad es técnicamente difícil de lograr. Java cuenta con un gran éxito en las aplicaciones en el entorno del servidor, como los Servlets o los Java Beans. ActionScript “gana” la partida en el entorno del cliente – tiene un 98% de penetración de mercado en cuanto a Web se refiere.

Para asegurar la independencia de la plataforma, se compila el código escrito, generando un código conocido como *bytecode* – instrucciones máquina simplificadas. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode, que recibe este nombre porque cada código de operación tiene una longitud de 1 byte, es ejecutado entonces en la máquina virtual (VM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por la VM.

La primera VM de ActionScript (AVM1) es una máquina virtual interpretada, para conseguir la ansiada portabilidad. Esto hace que los programas se ejecuten comparativamente más lentos que aquellos escritos en C o C++. Las implementaciones recientes de la VM (AVM2) dan lugar a programas que se ejecutan considerablemente más rápido, aumentando así la potencia de este lenguaje. En Flash Lite, por el momento, sólo está disponible la AVM1. La AVM2 se emplea en aplicaciones escritas en ActionScript 3.



Contenido



MUCOM 2009-2010

- Metodología Orientada a Objetos
- Independencia de la plataforma
- **El recolector de basura** ►
- Rendimiento
- Seguridad
- El formato SWF

INTERFACES GRÁFICAS MULTIMEDIA





El recolector de basura



MUCOM 2009-2010

- **Gestión automática de la memoria**
El desarrollador se olvida de las tareas a bajo nivel.
- **Gestión del ciclo de vida de los objetos**
Consumo de recursos eficiente.
- **Proceso totalmente transparente**
No guarda necesariamente relación con las acciones que realiza el código fuente.



INTERFACES GRÁFICAS MULTIMEDIA



Un argumento en contra de lenguajes como C es que los programadores se encuentran con la carga añadida de tener que administrar la memoria de forma manual. En C, el desarrollador debe asignar memoria en una zona conocida como *heap* para crear cualquier objeto, y posteriormente desalojar el espacio asignado cuando desea borrarlo. Un olvido a la hora de desalojar memoria previamente solicitada, o si no se hace en el instante oportuno, puede llevar a una fuga de memoria, ya que el sistema operativo piensa que esa zona de memoria está siendo usada por una aplicación cuando en realidad no es así. De esta forma, un programa mal diseñado podría consumir una cantidad desproporcionada de recursos. Además, si una misma región de memoria es desalojada dos veces el programa puede volverse inestable y llevar a un eventual cuelgue.

En ActionScript, este problema potencial es evitado en gran medida por el recolector automático de basura (*garbage collector*). El programador determina cuándo se crean los objetos y la aplicación, durante el tiempo de ejecución, es la responsable de gestionar el ciclo de vida de los objetos. El programa u otros objetos pueden tener localizado un objeto mediante una referencia a éste (que, desde un punto de vista de bajo nivel es una dirección de memoria). Cuando no quedan referencias a un objeto, el recolector de basura borra el objeto, liberando así la memoria. En definitiva, el recolector de basura permite una fácil creación y eliminación de objetos, frecuentemente más rápida que en C, y ofrece mayor seguridad, ya que es la VM la encargada de gestionar la memoria.

La recolección de basura es un proceso totalmente transparente. El programador no tiene conciencia de cuándo tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente. Debe tenerse en cuenta que la memoria es sólo uno de los muchos recursos que deben ser gestionados.



Contenido



MUCOM 2009-2010

- Metodología Orientada a Objetos
- Independencia de la plataforma
- El recolector de basura
- **Rendimiento** ►
- Seguridad
- El formato SWF

INTERFACES GRÁFICAS MULTIMEDIA





Rendimiento



MUCOM 2009-2010

- **Eficiencia de la máquina virtual**
Al ser interpretado el bytecode, los programas son más lentos.
Flash Player consume poca memoria (tanto dinámica como en disco) y mucho procesador.
- **El recolector de basura añade sobrecarga**
Es el precio que hay que pagar por la gestión automática de memoria.



INTERFACES GRÁFICAS MULTIMEDIA



El rendimiento de una aplicación está determinado por multitud de factores, por lo que no es fácil hacer una comparación que resulte totalmente objetiva. En tiempo de ejecución, el rendimiento de una aplicación ActionScript depende más de la eficiencia de la VM, que de las propiedades intrínsecas del lenguaje. El bytecode puede ser interpretado bien en tiempo de ejecución, bien al cargarse el programa, o bien durante la propia ejecución (JIT), para generar código nativo que se ejecuta directamente sobre el hardware. Lo que está claro es que si es interpretado, será más lento que usando el código nativo de la plataforma destino. Y si es compilado, durante la carga inicial o la ejecución, la penalización está en el tiempo necesario para llevar a cabo la compilación.

El uso de un recolector de basura para eliminar de forma automática aquellos objetos no requeridos añade una sobrecarga que normalmente afecta al rendimiento.



Contenido



MUCOM 2009-2010

- Metodología Orientada a Objetos
- Independencia de la plataforma
- El recolector de basura
- Rendimiento
- **Seguridad** ►
- El formato SWF

INTERFACES GRÁFICAS MULTIMEDIA





- **Flash Lite está muy limitado a nivel de hardware**
Estas limitaciones permiten la portabilidad y añaden "seguridad" a las aplicaciones.
- **Modelo de seguridad SandBox**
Recursos muy estrictos y limitados disponibles para cada aplicación.



ActionScript ofrece sobre todo seguridad y portabilidad, por ello no ofrece acceso directo al hardware de la arquitectura ni al espacio de direcciones. Tampoco soporta expansión de código ensamblador, aunque las aplicaciones pueden acceder a características de bajo nivel usando librerías nativas.

Flash Player usa un modelo de seguridad *sandBox*, lo que significa que las aplicaciones Flash que están ejecutándose disponen de recursos muy estrictos y limitados. Dichas aplicaciones, por ejemplo, no pueden leer archivos del disco duro o memoria del dispositivo (excepto los SharedObjects, tipo cookies) y, a partir de la versión 7 del reproductor, sólo pueden comunicarse con su dominio de red, a menos que sea permitido explícitamente por otro(s) dominio(s).



Contenido



MUCOM 2009-2010

- Metodología Orientada a Objetos
- Independencia de la plataforma
- El recolector de basura
- Rendimiento
- Seguridad
- El formato SWF ►

INTERFACES GRÁFICAS MULTIMEDIA





El formato SWF



MUCOM 2009-2010

- **Objetivo: archivos multimedia muy ligeros**
Optimización del ancho de banda y memoria.
Formato binario que usa compresión Zlib.
- **Formato totalmente abierto**
Desde la creación del Open Screen Project (Mayo de 2008).



INTERFACES GRÁFICAS MULTIMEDIA



El objetivo principal del formato SWF es crear archivos pequeños pero que permitan interactividad y funcionen en cualquier plataforma, aún sobre un ancho de banda reducido. El plugin que permite reproducir ficheros SWF es gratuito y está disponible para todos los navegadores y sistemas operativos más conocidos.

El formato es bastante simple, si bien es cierto que al ser binario no es de lectura accesible, al contrario que el SVG – estándar abierto basado en XML, recomendación del W3C. SWF ha utilizado la compresión Zlib desde 2002, con lo que los datos se almacenan usando el menor número de bits, minimizando la redundancia.

Pese a ser un formato propietario de Adobe, todas las especificaciones están completamente disponibles desde la creación del Open Screen Project en Mayo de 2008.

Por otro lado, la creación de software que genere archivos SWF está permitida con la condición de que el archivo resultante pueda ser ejecutado sin problemas por la última versión pública del reproductor de Adobe.