Interfaz Webcam

Gestión de contenidos mediante detección de movimiento

Esta es un aplicación que, aunque todavía no es posible aplicarla a dispositivos móviles con Flash Lite (pero sí otros dispositivos que tengan Flash Player ≥ 6), muestra parte del potencial de ActionScript.

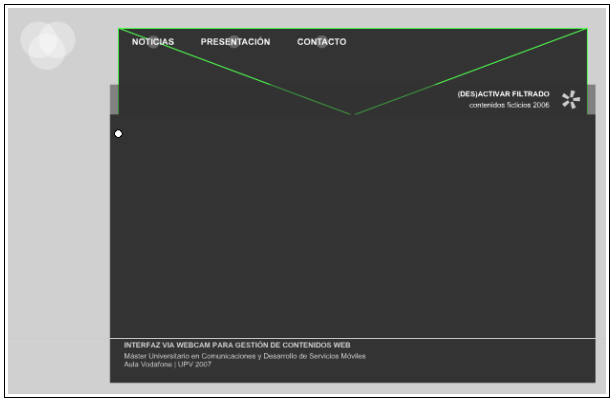
El objetivo es crear una interfaz gráfica que funcione no sólo con ratón, si no mediante webcam. La aplicación será una Web *mockup* con tres secciones, cada una con unos contenidos ficticios.

Los actores

Por un lado tendremos un menú con tres elementos para seleccionar; cada uno cargará la sección correspondiente. Al pasar el ratón por encima se crea una pequeña animación formada por tres círculos, que indican visualmente la sección seleccionada.

Por otro lado tenemos las tres secciones a mostrar, que serán tres MovieClips, llamados `secc1`, `secc2` y `secc3`, cada uno en una capa, todos dentro de una carpeta de capas.

Por último, el más importante, crearemos un MovieClip llamado `mcVideo`, el cual llevará dentro un objeto *Video*.



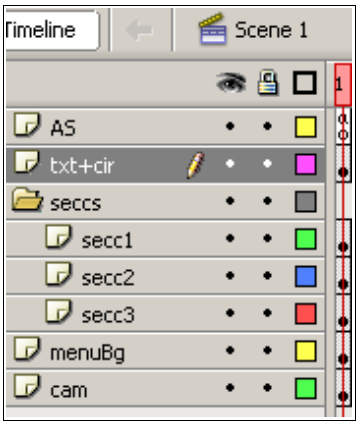
El escenario

Crearemos una película de 640x480 px a 25fps. Imprescindible indicar Flash8 y AS2 en las opciones de publicación, ya que vamos a emplear la clase `BitmapData` para monitorizar la webcam.

Las capas

Crearemos 6 capas. De inferior a superior son:

- *cam* – donde va el MovieClip `mcVideo`.
- *menuBg* – donde van los textos del interfaz.
- *seccs* – la carpeta de capas para las secciones.
- *txt+cir* – incluye el menú y los círculos animados.
- *AS* – la capa de programación.



Capa *cam*

Creamos desde la biblioteca un nuevo objeto *Video (ActionScript controlled)*, que por defecto mide 160x120 px y lo arrastramos dentro del MovieClip `mcVideo`, asignándole las dimensiones de 640x240 px y el nombre de instancia `vid`. Lo alineamos al centro del escenario y a su margen superior.

Capa *menuBg*

Dibujamos un rectángulo que cubra casi todo el escenario, excepto la parte del MovieClip `mcVideo` que queramos usar para detectar el movimiento. El resto de elementos decorativos se deja a elección del alumno.

Capa *secs*

El propósito de esta capa es agrupar las tres secciones a mostrar en tres capas individuales, por lo que puede obviarse. Lo importante es tener los tres MovieClips de las tres secciones en el escenario encima del objeto *Video* y de la capa *menuBg*. Cada sección empieza con un fotograma clave en blanco con la orden `stop()`, seguido de una animación hasta pararse en el último fotograma de dicha animación – con otro `stop()`.

Capa *txt+cir*

Aquí ubicamos el menú de navegación y los tres círculos animados. Cada item del menú de navegación es un círculo con un texto descriptivo, llamados `cir1`, `cir2` y `cir3`, respectivamente.

Los tres círculos animados son tres instancias del mismo MovieClip, una animación que consiste en rotar de forma aleatoria un gráfico mediante una capa guía, asignando esta acción en el fotograma 1:

```
gotoAndPlay(random(_totalframes));
```

Al final de la animación, al no haber ninguna orden de `stop()`, estaremos realizando un bucle que cada vez comienza en un fotograma distinto.

Colocaremos estos tres círculos fuera del escenario, para que aparezcan posteriormente al seleccionar alguna sección.

Capa *AS*

Esta es la responsable de toda la programación. Podemos dividirla en dos capas para diferenciar claramente la detección de movimiento y las acciones del resto de elementos.

Para empezar, preparamos la aplicación:

```
fscommand("fullscreen", true);
fscommand("allowscale", false);
fscommand("showmenu", false);
Stage.scaleMode = "noScale";
Stage.showMenu = false;
```

Monitorizado de la webcam

Vamos a realizar la monitorización más simple de todas: el cambio de color mediante diferencia RGB. No se han elegido otros métodos más precisos dado que el propósito es mostrar la filosofía de detección de movimiento.

En primer lugar importamos la clase que vamos a emplear:

```
import flash.display.BitmapData;
```

A continuación obtenemos la imagen de la webcam y la asignamos al objeto Video:

```
var cam:Camera = Camera.get();
mcVideo.vid.attachVideo(cam);
```

Definimos la zona a monitorizar:

```
var zona:BitmapData = new BitmapData(   →
640, 240);
```

Ahora creamos las zonas activas de la aplicación, que no son más que las coordenadas {x,y} de un pixel determinado:

```
var px1, d1, px2, d2, px3, d3 :Number;
var f1:Number = cir1._x + cir1._width/2;
var f2:Number = cir3._x + cir3._width/2;
var f3:Number = cir2._x + cir2._width/2;
var sensibilidad:Number = 1400000;
```

La variable `sensibilidad` indica cuánto cambio de color ha de detectarse para disparar un evento.

Declaramos la función de detección:

```
function accion(sens:Number):Void {
 // refresco de la webcam
 // diferencia de color entre 2 instantes
 // detección del umbral de sensibilidad
 // actualización del color
};
```

Y la llamamos unas 30 veces por segundo:

```
setInterval(accion, 33, sensibilidad);
```

Para refrescar la imagen de la webcam debemos usar esta orden:

```
zona.draw(mcVideo);
```

Para hallar la diferencia de los valores de color entre dos instantes consecutivos en el tiempo:

```
d1 = Math.abs(px1-zona.getPixel(f1, 15));
d2 = Math.abs(px2-zona.getPixel(f2, 15));
d3 = Math.abs(px3-zona.getPixel(f3, 15));
```

Para detectar si hemos pasado el umbral definido en la variable `sensibilidad`:

```
if (d1 > sens) {
 muestra(secc1);
} else if(d2 > sens) {
 muestra(secc3);
} else if (d3 > sens) {
 muestra(secc2);
}
```

Para actualizar los valores de los pixels activos hay que restarlos de los que serán los valores actuales en la siguiente toma:

```
px1 = zona.getPixel(f1, 15);
px2 = zona.getPixel(f2, 15);
px3 = zona.getPixel(f3, 15);
```

Comportamiento del menú

Al cargar la aplicación, es conveniente ocultar las tres secciones inicialmente:

```
for (var i:Number = 1; i <= 3; ++i)     →
this["secc"+i]._visible = false;
```

Existen otras formas de hacerlo, por ejemplo importando los ítems desde la biblioteca con el método `attachMovie()`, pero no incidiremos en la eficiencia de una forma u otra.

Por último sólo falta crear la función para mostrar una sección, que básicamente es:

```
import mx.transitions.Tween;
import mx.transitions.easing.*;
function muestra(nombre):Void {
 // sacar el índice del ítem seleccionado
 // animar los 3 círculos hasta ese ítem
 // resetear las secciones
 // mostrar la sección seleccionada
};
```

Para saber el índice del ítem seleccionado lo que hacemos es sacar el último carácter del nombre del ítem:

```
var nom:String = nombre._name;
var n:String = nom.substr(nom.length-1,
nom.length);
```

Así, si seleccionamos el item2 del menú, que se llama `secc2`, `n` valdrá 2. Y sabremos entonces qué sección mostrar y hasta dónde animar los tres círculos.

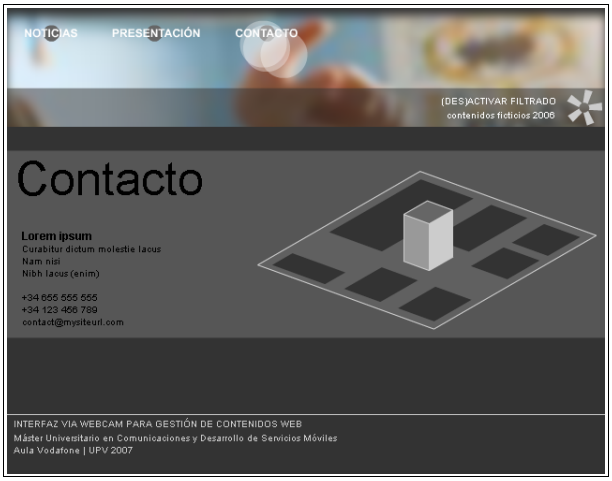
Para animar los tres círculos hasta la posición del item seleccionado emplearemos la clase Tween:

```
for (var i:Number = 1; i <= 3; ++i)      →
new Tween (_root["focus"+i], "_x",       →
Strong.easeOut, _root["focus"+i]._x,     →
_root["cir"+n]._x, 1.5, true);
```

Dado que cada sección tiene una animación, al seleccionar un item del menú debemos ocultar todas las secciones, llevándolas a su primer fotograma (donde había un `stop()`), y asignar un `play()` al MovieClip de la sección elegida:

```
if (nombre._currentframe !=              →
nombre._totalframes) {
 for (var i:Number = 1; i <= 3; ++i) {
 _root["secc"+i]._visible = false;
 _root["secc"+i].gotoAndStop(1);
}
nombre._visible = true;
nombre.play();
```

Y eso es todo. Ya podemos conectar la webcam al ordenador y lanzar la aplicación.



Para asignar acciones al ratón al pasar por encima de los items del menú:

```
cirN.onRollOver = function():Void {
 muestra(seccN);
};
```

Donde `N` es `1`, `2` o `3`.