

5th May 2010

Crear un control de volumen en
ActionScript 3.0

Veíamos en un post anterior, que mediante el objeto SoundTransform controlamos el volumen de un sonido. Para hacer **una barra de volumen** como en la mayoría de players, lo que haremos es asociar la propiedad x del objeto que haga de pestaña deslizante, con la propiedad volume del objeto SoundTransform.

El único problema, es que por motivos de diseño, la barra deslizante puede tener un recorrido de 100 o 200 píxeles, y el volumen en AS3 tiene un valor que va de 0 a 1

[leer más...](#) [\[http://desarrolloparaweb.blogspot.com/2010/05/crear-un-control-de-volumen-en.html\]](http://desarrolloparaweb.blogspot.com/2010/05/crear-un-control-de-volumen-en.html)

Crear una barra deslizadora

Para crear una barra deslizadora, utilizaremos la función **startDrag**, pero con todos sus parámetros: en primer lugar nos pide si queremos bloquear el objeto en el centro, normalmente pondremos "false", porque esto nos permite que funcione si arrastramos desde una esquina de la pestaña y el objeto no coloque su centro de coordenadas en el punto del mouse.

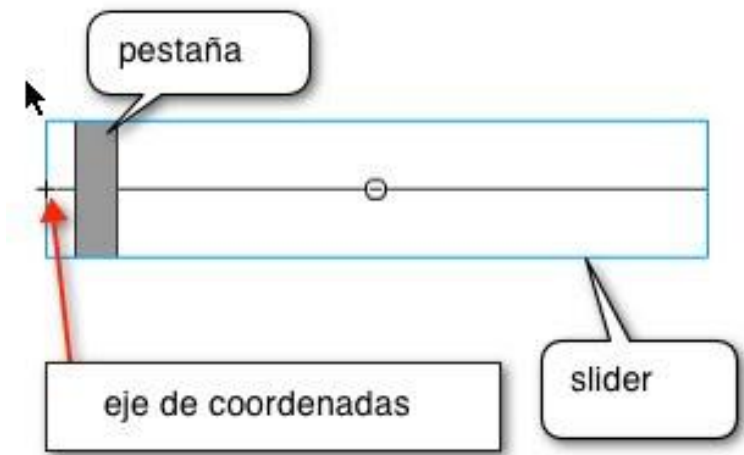
```
startDrag (
    flash.display.Sprite.startDrag([bloquearCentro:Boolean=false,limites:flash.geom.Rectangle=null]):void
```

[\[http://4.bp.blogspot.com/_G62iQFG_F58/S-GE5C6AIhI/AAAAAAAAABbQ/aq8ewCrrRuY/s1600/drag.jpg\]](http://4.bp.blogspot.com/_G62iQFG_F58/S-GE5C6AIhI/AAAAAAAAABbQ/aq8ewCrrRuY/s1600/drag.jpg)

En segundo lugar nos pide que le indiquemos los límites, pero se los debemos proporcionar en el formato **Rectangle** que es una clase dentro de la ruta **flash.geom**. En realidad no es nada complicado, puesto que la clase rectangle se utiliza para almacenar cuatro coordenadas (x, y, anchura y altura).

```
var rect:Rectangle = new Rectangle(0,0,200,0);
e.target.startDrag(false, rect);
```

Para que nos funcione la barra deslizadora, **la pestaña debe estar dentro de un MovieClip, para así aprovechar las coordenadas locales del MovieClip y no las del Stage**. Así que creamos un MovieClip que se llame pestaña, y luego lo metemos dentro de otro que se llame por ejemplo slider.



[\[http://4.bp.blogspot.com/_G62iQFG_F58/S-GG7INqZel/AAAAAAAAABbY/SICPQ1aqY9Y/s1600/slider.jpg\]](http://4.bp.blogspot.com/_G62iQFG_F58/S-GG7INqZel/AAAAAAAAABbY/SICPQ1aqY9Y/s1600/slider.jpg)

El MovieClip Slider está compuesto de una línea horizontal (un vector) y de un MovieClip (pestaña) que queremos que se pueda arrastrar en horizontal durante 200 píxeles. La acción se la podemos asignar tranquilamente a slider, y usar la propiedad **e.target** (que en éste caso será "pestaña" puesto que es el único que hay).

```
function arrastrar(e:MouseEvent):void{
var rect:Rectangle = new Rectangle(0,0,200,0);
e.target.startDrag(false,rect);
}
```

```
slider.addEventListener(MouseEvent.CLICK, arrastrar);
```

Para que funcione correctamente tenemos que crear el símbolo slider con **el punto de registro en la izquierda**. Esto hará que el valor 0 del eje de coordenadas esté en la izquierda.

Asociar la propiedad x con la propiedad volume

Una vez tenemos la pestaña con la función startDrag, ésta se podrá arrastrar de tal manera que su propiedad x tendrá un valor que irá entre 0 y 200, para poder aplicarlo al volumen lo dividiremos entre 200 para obtener un valor que va de 0 a 1. Podemos crear un campo de texto para visualizar el volumen:

```
nuevoVolumen = slider.pt.x/200; // la pestaña tiene el nombre de instancia pt
info_txt.text = String(nuevoVolumen);
myTransform.volume = nuevoVolumen;
sch.soundTransform = myTransform;
```

Como comentaba anteriormente, una vez cambiamos la propiedad volume, deberemos volver a asignar el objeto SoundTransform a la propiedad soundTransform del objeto SoundChannel como hacemos en la última línea. Por último deberemos tener en cuenta, que para que esté continuamente funcionando, estas acciones deberán ejecutarse mediante un ENTER_FRAME

A continuación os dejo el código completo del ejercicio

```
var so:Sound = new Sound();
var sch:SoundChannel = new SoundChannel();
var myTransform = new SoundTransform();
var req:URLRequest = new URLRequest("bomb.mp3");
so.load(req);
```

```
sch = so.play();
myTransform.volume = 0.2;
sch.soundTransform = myTransform;
```

```
var posicion:Number = 0;
function mute(e:MouseEvent):void{
    posicion = sch.position;
    sch.stop();
}
```

```
mute_btn.addEventListener(MouseEvent.CLICK, mute);
```

```
function reproducir(e:MouseEvent):void{
    sch = so.play(posicion);
}
```

```
play_btn.addEventListener(MouseEvent.CLICK, reproducir);
function volumenTope(e:MouseEvent):void{
    myTransform.volume = 1;
    sch.soundTransform = myTransform;
}
```

```
stage.addEventListener(MouseEvent.CLICK, volumenTope);
```

```
// sound complete
```

```
sch.addEventListener(Event.SOUND_COMPLETE, soundCompleteHandler);  
function soundCompleteHandler(e:Event):void {  
    trace("ya está");  
    posicion = 0;  
    sch = so.play(posicion);  
}
```

```
var tiempo:Number = 0;  
function tempo(e:Event):void{  
    tiempo = (Math.round(sch.position));  
    tiempo = tiempo/1000;  
    time_txt.text = String(tiempo);  
}
```

```
stage.addEventListener(Event.ENTER_FRAME, tempo);
```

```
function arrastrar(e:MouseEvent):void{  
    var rect:Rectangle = new Rectangle(0,0,200,0);  
    e.target.startDrag(false,rect);  
}
```

```
function soltar(e:MouseEvent):void{  
    stopDrag();  
}
```

```
slider.addEventListener(MouseEvent.MOUSE_DOWN, arrastrar);  
slider.addEventListener(MouseEvent.MOUSE_UP, soltar);
```


```
var nuevoVolumen:Number = 0;  
function modificarSonido(e:Event):void{  
    nuevoVolumen = slider.pt.x/200;  
    info_txt.text = String(nuevoVolumen);  
    myTransform.volume = nuevoVolumen;  
    sch.soundTransform = myTransform;  
  
}
```

```
stage.addEventListener(Event.ENTER_FRAME, modificarSonido);
```

Publicado 5th May 2010 por [José Balaguer](#)
Etiquetas: [ActionScript 3.0](#)

3


 Ver comentarios



Ivancho [6 de mayo de 2010, 15:06](#)

Muchas Gracias por tu aporte


[Responder](#)



Julian [14 de julio de 2010, 10:57](#)

Muchas gracias por el aporte, he visto cosas muy buenas en este blog y me han ayudado mucho gracias.

[Responder](#)




Anonymous [10 de julio de 2012, 7:52](#)


Hola por que este error??

TypeError: Error #1010: Un término no está definido y no tiene propiedades.
at controldevolumen fla::MainTimeline/modificarSonido()

[Responder](#)

Introduce tu comentario...

Comentar como:

Christian (Gooꝑ 

Cerrar sesión

Publicar

Vista previa

☐ Avisarme