

Tutorial 7 - Control de la línea de tiempo

Paso 1 de 11

En este tutorial vamos a añadir dos botones para reproducir y reanudar nuestra animación.

Aprenderemos cómo controlar tanto la línea de tiempo principal de una película, como las líneas de tiempo de clips anidados dentro de ella.

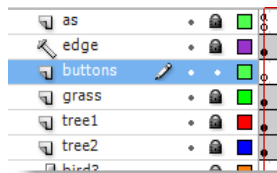
También aprenderemos cómo controlar la reproducción de animaciones creadas mediante programación.

Para hacer este tutorial, abrimos el archivo creado en el tutorial anterior (*tutorial6.fla*) y lo guardamos como *tutorial7.fla*.



Paso 2 de 11

En primer lugar vamos a crear una nueva capa llamada *buttons* por encima de la hierba, en la que colocaremos los botones con los que controlaremos la línea de tiempo.

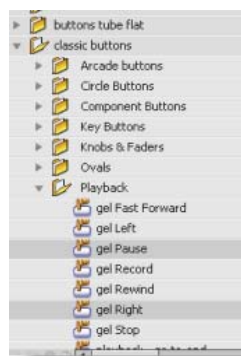


Para estos botones vamos a utilizar botones ya creados que podemos encontrar seleccionando **Ventana >**

Bibliotecas comunes > Botones. En otro tutorial aprenderemos a modificar y crear nuestros propios botones, y entraremos en detalle sobre su funcionamiento.

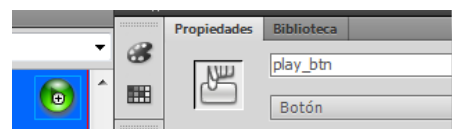
Arrastramos al escenario los botones *gel Pause* y *gel Right* que se encuentran en la carpeta **classic buttons > Playback**.

Ahora estos botones aparecerán en la biblioteca de nuestro documento. Por tanto ya podemos cerrar la biblioteca de los botones.



Seleccionamos la instancia de *gel Pause* en el escenario y le damos el nombre de instancia *pause_btn* en el inspector de **Propiedades**.

A la instancia del botón *gel Right* le damos el nombre *play_btn*



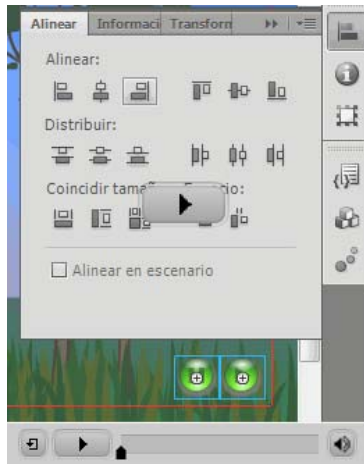
Cuando la línea de tiempo se esté reproduciendo queremos que esté visible el botón para pausar la película, y cuando la película esté pausada necesitaremos tener visible el botón para reanudarla.

Es decir, vamos a hacer que ambos botones se sitúen en el mismo lugar, y que se muestre uno u otro dependiendo del estado de la reproducción, como si fuera un conmutador.

Paso 3 de 11

Seleccionamos los botones con la tecla **Mayúsculas** pulsada para poder seleccionar ambos al mismo tiempo. Los situamos cerca de la esquina inferior derecha del escenario. Recordemos que los límites del escenario están definidos en este caso por el marco rojo.

Con ayuda del panel **Alinear**, colocamos ambos botones en el mismo lugar del escenario, uno encima del otro.



Abrimos el panel **Acciones (F9)**. Escribiremos tras la última línea que habíamos programado en el tutorial anterior, en el fotograma 1 de la capa as.

Para diferenciar ambos tutoriales y mejorar la legibilidad del código, vamos a introducir, además de alguna línea en blanco, un comentario que separe ambas partes del código y que nos indique en qué zona de la programación nos encontramos. Por ejemplo podemos escribir el siguiente comentario:

```
//control buttons
```

Cuando probamos la película, ésta se reproduce automáticamente, por lo que el botón que queremos que esté visible en un primer momento es el botón para pausar la animación. Por tanto, tendremos que hacer invisible el botón *play_btn*. Para ello escribiremos este código:

```
play_btn.visible = false;
```

La propiedad *visible* admite los valores *true* (visible) y *false* (invisible).

Paso 4 de 11

Un botón invisible no sólo no se verá, sino que además no estará activo. Sin embargo, un botón con un valor 0 de alfa podrá pulsarse aunque no sea visible. Es por ello que, cuando queremos desactivar un objeto además de hacerlo invisible, sea preferible utilizar la propiedad *visible* en lugar de la propiedad *alpha*.

El siguiente paso es añadir detectores del evento de ratón *CLICK* para ambos botones, llamando al botón *play_btn* a una función a la que llamaremos *playMovie*, y el botón *pause_btn* a una función llamada *stopMovie*:

```
pause_btn.addEventListener(MouseEvent.CLICK, stopMovie);
play_btn.addEventListener(MouseEvent.CLICK, playMovie);
```

La función *stopMovie* parará la película. Cuando la película esté parada necesitaremos tener visible el botón para reanudar la reproducción, mientras que ya no será necesario mostrar el botón para pausarla.

En el caso de la función *playMovie*, en primer lugar reanudará la reproducción de la película. Cuando la película se reproduzca de nuevo, necesitaremos tener visible el botón para pausarla, pero ya no será necesario tener visible el botón para reproducirla.

Las funciones *stop()* y *play()* detienen y reproducen respectivamente la línea principal de tiempo.

Por lo tanto, de momento las funciones *stopMovie* y *playMovie* quedarán de la siguiente manera:

```
function stopMovie(e:MouseEvent):void
{
    stop();
    play_btn.visible = true;
    pause_btn.visible = false;
}

function playMovie(e:MouseEvent):void
{
    play();
    pause_btn.visible = true;
    play_btn.visible = false;
}
```

Si hay alguna duda con esta parte de este código, recomendamos repasar el tutorial anterior.

Paso 5 de 11

La línea `play_btn.visible = false;` que habíamos colocado antes de las funciones (paso 3), se ejecutará directamente al probar la película cuando la cabeza lectora esté en el primer fotograma. Sin embargo, el resto de instrucciones sobre la propiedad `visible`, al estar dentro de una función, sólo serán ejecutadas cuando la función que las contiene sea llamada.

Seleccionamos **Control > Probar película (Ctrl+Intro)**.

Podemos comprobar por un lado cómo se alterna la visibilidad de los botones `play_btn` y `pause_btn` al pulsarlos.

Observamos que pulsando `pause_btn` se detiene la línea de tiempo principal, es decir, el avance de los pájaros. Sin embargo las alas siguen en movimiento, ya que la función `stop()` hace referencia a la línea de tiempo principal, pero no a las líneas de tiempo anidadas dentro de ella (como en este caso las alas).

Por otro lado, también continúa el avance de las nubes por el escenario, ya que su movimiento depende del evento `ENTER_FRAME`, y este evento es independiente del desplazamiento de la cabeza lectora en la línea de tiempo principal.

Para solucionar el problema del movimiento de las alas, primero tendremos que saber cómo acceder a ellas desde la programación en la línea de tiempo principal.

Para acceder a una línea de tiempo anidada dentro de otra, en la programación escribiremos el nombre de la instancia principal, seguida de un punto, y seguida del elemento que está contenido dentro de ella.

Por ejemplo, supongamos que queremos reproducir la línea de tiempo de la mano derecha que se encuentra dentro del clip del brazo derecho que a su vez pertenece a un cuerpo. Para ello tendríamos que escribir lo siguiente:

```
body.right_arm.right_hand.play();
```

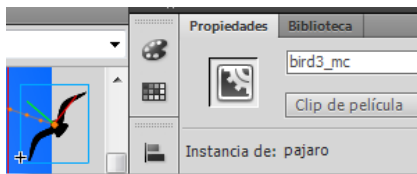
De esa forma la función `play()` haría referencia a esa mano. Por lo tanto, para poder acceder a las alas de cada pájaro tendremos que dar un nombre de instancia en primer lugar a cada pájaro, y después a las alas.

Paso 6 de 11

Para poder dar un nombre de instancia a los pájaros, desbloqueamos en primer lugar las capas que los contienen.

Seleccionamos cada pájaro en el escenario y les asignamos los nombres de instancia `bird1_mc`, `bird2_mc` y `bird3_mc`, siguiendo la numeración de los nombres de las capas en las que se encuentran.

Para seleccionar a los pájaros 2 y 3 tendremos que desplazar la cabeza lectora, ya que no aparecen en el escenario hasta los fotogramas 24 y 150 respectivamente.

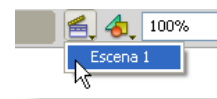


Si hacemos clic dos veces sobre cualquiera de los pájaros del escenario, entraremos a la edición del clip `bird` original.

Este clip de película principal tiene en su interior dos instancias del clip de película `wing`. Seleccionamos el ala izquierda y le damos el nombre de instancia `wing1_mc`. Al ala derecha le damos el nombre `wing2_mc`.



Volvemos a la escena principal pulsando sobre *Escena 1* en la barra de edición, o bien seleccionándola en el desplegable.



Paso 7 de 11

Si pulsamos dos veces sobre cualquiera de los otros pájaros, veremos que nos lleva al mismo clip *bird* original, al que ya hemos dado los nombres de instancia *wing1_mc* y *wing2_mc*.

Probemos a detener un ala de un pájaro. Para ello tendremos que añadir la siguiente línea de código dentro de función *stopMovie*:

```
bird1_mc.wing1_mc.stop();
```

de tal forma que la función *stopMovie* quede como sigue:

```
function stopMovie(e:MouseEvent):void
{
    stop();
    play_btn.visible = true;
    pause_btn.visible = false;
    bird1_mc.wing1_mc.stop();
}
```

Probamos la película y pulsamos sobre el botón *pause_btn*. El ala izquierda del primer pájaro se detendrá a la vez que la línea de tiempo principal. Aunque reanudem la película, este ala no volverá a reproducirse ya que no hemos escrito instrucciones para ello.

Antes de continuar, probemos a sustituir en la programación *bird1_mc* por *bird3_mc*, quedando la línea como sigue:

```
bird3_mc.wing1_mc.stop();
```

Si volvemos a probar la película, y pulsamos el botón *pause_btn* antes de que el pájaro 3 aparezca en escena, nos encontraremos con el siguiente error:

```
TypeError: Error #1009: No se puede acceder a una
propiedad o a un método de una referencia a un
objeto nulo.
    at tutorial7_fla:MainTimeline/detener()
```

Esto quiere decir que al pulsar el botón para detener la película y ejecutarse la función *stopMovie*, se hace referencia a un objeto que no existe en ese momento (*bird3_mc* en este caso), y no puede por tanto ejecutar la instrucción de detener el ala de ese pájaro.

Pese a que esto no impide el funcionamiento correcto de nuestra película, para evitar este error comprobaremos qué pájaros están en el escenario en el momento de pulsar los botones, y sólo ejecutaremos las instrucciones para detener las alas de esos pájaros.

Paso 8 de 11

Para saber qué pájaros se encuentran en el escenario en el momento de pulsar un botón será necesario saber en primer lugar en qué fotograma nos encontramos en ese momento.

Para conocer en qué fotograma se encuentra la cabeza lectora en un momento dado, utilizaremos la propiedad *currentFrame*.

Sabemos que el pájaro 1 está desde el primer fotograma y que desaparece en el fotograma 500. Por lo tanto, si la propiedad *currentFrame* devuelve un número menor que 501, el pájaro 1 estará en escena.

El pájaro 2 aparece en el fotograma 24 y desaparece en el 524, así que si la cabeza lectora se encuentra entre esos dos valores, también podremos asegurar que el pájaro 2 está en el escenario. De la misma forma, el pájaro 3 está presente entre los fotogramas 150 y 350.

Para considerar que un pájaro está en escena no es necesario que se encuentre dentro del marco visible del escenario, sino que lo que se tiene en cuenta es que esté presente en la línea de tiempo.

Teniendo en cuenta que < significa menor que, > significa mayor que, y && equivale al AND lógico, la función *stopMovie* quedará como sigue:

```
function stopMovie(e:MouseEvent):void
{
    stop();
    play_btn.visible = true;
    pause_btn.visible = false;

    if (currentFrame < 501)
    {
        bird1_mc.wing1_mc.stop();
        bird1_mc.wing2_mc.stop();
    }

    if (currentFrame > 23 && currentFrame < 525)
    {
        bird2_mc.wing1_mc.stop();
        bird2_mc.wing2_mc.stop();
    }

    if (currentFrame > 149 && currentFrame < 351)
    {
        bird3_mc.wing1_mc.stop();
        bird3_mc.wing2_mc.stop();
    }
}
```

Paso 9 de 11

Vamos a explicar uno de los condicionales para comprender mejor el código. Por ejemplo, con la condición

```
if (currentFrame > 23 && currentFrame < 525)
```

estamos comprobando si el fotograma actual es mayor que el fotograma 23 y, además, es menor que el fotograma 525.

Si se cumplen ambos requisitos, entonces la condición completa se cumple, y pasarán a ejecutarse las instrucciones que estén entre las llaves de ese `if`. Si alguna de las partes de la condición no se cumple, no se ejecutarán las instrucciones correspondientes.

Según en qué fotograma estemos, puede que se cumplan las condiciones que hemos escrito para todos los pájaros (por ejemplo el fotograma 200), para sólo alguno de ellos (por ejemplo el fotograma 15), o para ninguno de los 3 (por ejemplo el fotograma 580).

Si probamos la película, veremos que ahora se paran correctamente las alas de los tres pájaros, y no se nos informa de ningún error. Ahora nos falta volver a activar el movimiento de las alas cuando volvemos a reproducir la película.

Para activar de nuevo el movimiento de las alas, copiamos en la función `playMovie` las sentencias condicionales que hemos utilizado en la función `stopMovie`. Después, simplemente sustituimos el método `stop()` por el método `play()`, por ejemplo:

```
if (currentFrame < 501)
{
    bird1_mc.wing1_mc.play();
    bird1_mc.wing2_mc.play();
}
```

Probamos de nuevo la película. Ahora con los botones `play_mc` y `pause_mc` podemos detener y reanudar tanto el avance de los pájaros por el escenario, como el movimiento de sus alas.

Además de parar con `stop()` y reanudar con `play()`, existen otros métodos interesantes a la hora de controlar una línea de tiempo, como por ejemplo:

- `nextFrame()` y `prevFrame()`, para avanzar y retroceder un fotograma
- `gotoAndPlay()` y `gotoAndStop()`, para ir a un fotograma concreto, y una vez allí continuar o detenemos, escribiendo dentro del paréntesis el número de fotograma al que queremos ir o su nombre (etiqueta).

Paso 10 de 11

Hasta aquí hemos visto cómo podemos controlar la cabeza lectora tanto de la línea de tiempo principal como de líneas de tiempo anidadas.

Ahora nos queda detener y reanudar el movimiento de las nubes, ya que este movimiento es independiente del desplazamiento de la cabeza lectora en la línea de tiempo principal.

El movimiento de cada nube lo habíamos creado en el tutorial anterior con el siguiente código:

```
cloud1_mc.addEventListener(Event.ENTER_FRAME, wind);
cloud2_mc.addEventListener(Event.ENTER_FRAME, wind);
```

Para que las nubes se detengan, será suficiente con eliminar los detectores del evento `ENTER_FRAME`. Para eliminarlos basta con copiar el mismo código que se usó para crearlos, y sustituir `addEventListener` por `removeEventListener`.

```
cloud1_mc.removeEventListener(Event.ENTER_FRAME,
wind);
cloud2_mc.removeEventListener(Event.ENTER_FRAME,
wind);
```

Por lo tanto, añadiremos las dos sentencias con `removeEventListener` a la función `stopMovie` antes de la última llave que marca el final de la función. Para que el movimiento se reanude después, añadiremos de nuevo en la función `playMovie` las dos sentencias con `addEventListener`.

Si además de querer detener el movimiento, quisiéramos que las nubes dejaran de ser arrastrables, tendríamos que seguir el mismo procedimiento, es decir, copiar la misma sentencia que utilizamos para agregar el listener y sustituir `add` por `remove`. En este caso vamos a permitir que las nubes puedan arrastrarse aunque la línea de tiempo esté detenida, así que no vamos a eliminar esos listeners.

El método `removeEventListener` puede ser muy útil en diversas circunstancias, por ejemplo para eliminar la interactividad de un botón o un clip sin necesidad de esconderlo ni eliminarlo del escenario.

Seleccionamos **Control > Probar película**. Ahora podemos detener y reanudar todas las animaciones (pájaros, alas y nubes).

Paso 11 de 11

Para complementar los conceptos desarrollados en este tutorial, se recomienda hacer las siguientes actividades:

1. Añade botones para avanzar y retroceder un fotograma
2. Añade un botón que vaya al principio de la película
3. Haz que la nube se puedan arrastrar sólo cuando la película esté detenida.

