

- Introducción.

El tema que vamos a estudiar es muy interesante: GENERACIÓN DINÁMICA DE CLIPS DE PELÍCULA mediante programación. ¿Qué quiere decir esto? Esto quiere decir que vamos a hacer que con código flash podamos hacer copias de objetos, es decir duplicados sin tener que utilizar las herramientas de dibujo de Flash.

1.- Creación de un duplicado de un clip de película

- Vamos a crear o pegar un objeto en el escenario y lo vamos a convertir en clip de película.
- Ponemos al clip de película un nombre de instancia (panel propiedades). Por ejemplo el nombre **bola**.
- Ahora introducimos este código en el fotograma de la película principal:

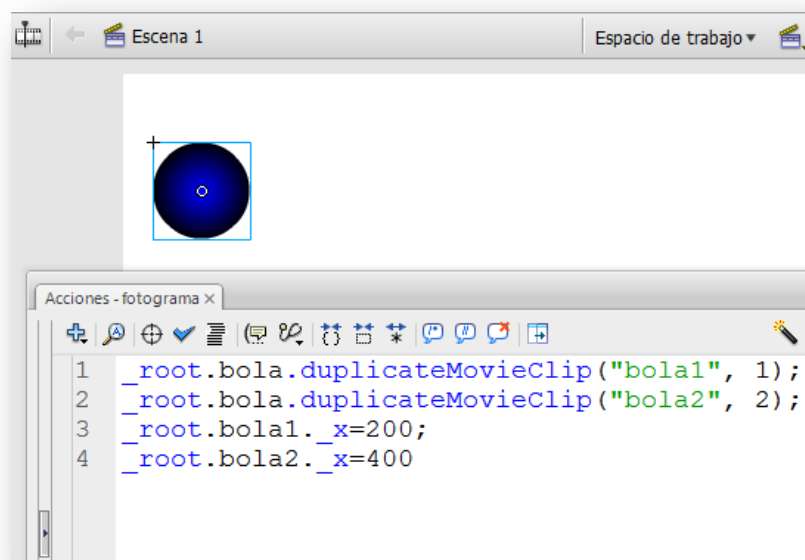


Imagen 1

Analicemos el código:

`_root.bola.duplicateMovieClip("bola1", 1);`

Con este código le decimos que duplique o cree un clon del clip `_root.bola`.

Este clon se llamará **`_root.bola1`** y heredará todas las propiedades de su padre. Es decir el color, el tamaño, sus coordenadas, su rotación y el código que lleve asociado.

El número 1 que ponemos al final se refiere al nivel de capa o también llamado nivel de profundidad. Y sirve para organizar los clips de película en capas. Es decir vemos el clip que está en la capa 2 delante del que está en la capa 1.

Si ejecutamos sólo veremos un objeto porque está encima de él ya que ha heredado sus **coordenadas _x e _y** y como están superpuestos uno encima del otro. Por eso para verlo en el escenario tendremos que situarlo en otro sitio. Esta es la razón del siguiente código:

```
_root.bola1._x=200;
```

```
_root.bola2._x=400;
```

Finalmente al ejecutar la aplicación veremos 3 clips de película, tres bolas. A la primera la llamaremos clip padre (o madre) y a los duplicados hijos o clones.

2.- Creación duplicaciones infinitas.

Ahora damos un paso más. Ya sabemos duplicar pero... ¿y si queremos **duplicaciones infinitas**?

- Vamos a crear en el fotograma de la película principal la variable: `_root.capa=1;`
- Creamos un botón que se llame **duplicar**.
- En el botón ponemos el siguiente código:

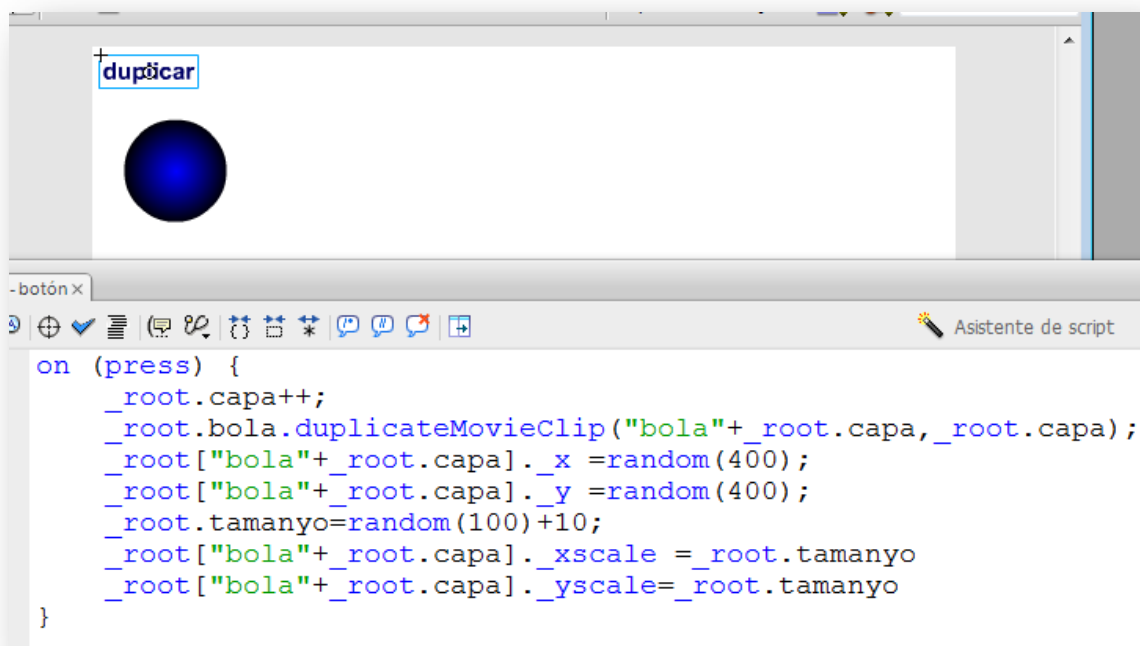


Imagen 2

Analicemos el nuevo código que está dentro del botón:

_root.capa++; Significa que cada vez que hagamos clic en el botón la variable `capa` suma una unidad más. La razón de esta acción es que cada clip duplicado debe estar en una capa diferente. No puede estar en la misma.

_root.bola.duplicateMovieClip("bola"+_root.capa,_root.capa); Con esta acción duplicamos el clip padre (`bola`) creando un hijo que tendrá por nombre `bola` más el número de capa que le toque. En este caso `"bola"` va entre comillas para decir al flash que queremos

concatenar (juntar) el texto "bola" con el número de capa para darle un nuevo nombre. Podríamos haber puesto cualquier nombre por ejemplo "pepe". La primera vez será "bola"+2 y esto traducido a programación es igual a bola2.

Y ahora: `_root["bola"+_root.capa]._x =random(400);`

Con este código que lleva corchetes le digo que ese clip se ponga en un lugar aleatorio de la pantalla en la coordenada x (horizontal). Y va en corchetes porque todos los clips duplicados quedan registrados como una matriz en flash. Por eso cuando llamamos a un clip duplicado lo podemos llamar con corchetes.

Con la acción **random(400)** estamos diciendo que elija un número aleatorio entre 0 y 400 para que cada vez que hagamos clip sitúe a los hijos en lugares diferentes de la pantalla.

¿Y si queremos que salgan clones de diferentes tamaños? Esta es la razón de las últimas líneas de código:

`_root.tamanyo=random(100)+10;` Con esta acción creamos una variable y almacenamos un número aleatorio entre el 0 y el 100. Y a este resultado le sumamos 10. ¿Por qué? Pues para evitar que salga tamaños muy pequeños entre el 0 y el 10 que no se visualizarían en la pantalla.

Y finalmente con el siguiente código asignamos las escalas (altura, anchura) los clips de película.

`_root["bola"+_root.capa]._xscale =_root.tamanyo;`

`_root["bola"+_root.capa]._yscale=_root.tamanyo;`

3.- Herencia de código.

Como hemos dicho al principio de este tutorial los clones también heredarán la inteligencia o el ADN de sus padres. Queremos decir el código.

Vamos a introducir en el clip **bola** (padre) el código para que pueda ser arrastrado:

```
on (press) {  
this.startDrag();  
}  
on (release) {  
this.stopDrag();  
}
```

Y ahora vamos a comprobar que al crear los duplicados cada uno pueda ser arrastrado.

4.- Ejercicio propuesto.

Propuestas:

1. Un actividad donde sale un número aleatorio entre el 1 y el 10 y el usuario debe crear tantos objetos como indica dicho número. El programa le podrá decir: *correcto, te has pasado o aún te faltan* cuando haga clic en el botón comprobar.
2. Propuesta libre.

duplicar

