- Introducción.

Este tutorial es una continuación del anterior que trataba de cómo programar objetos que se puedan arrastrar.

En este tutorial vamos a añadir las siguientes funcionalidades:

- Controlar cuándo colisiona un objeto con otro.
- Detecta errores y aciertos y visualizarlos en la pantalla.
- Bloquear la palabra una vez acertada para que no pueda ser arrastrada.
- Mover clips aleatoriamente en la pantalla.

1.- Creación de los objetos de la pantalla.

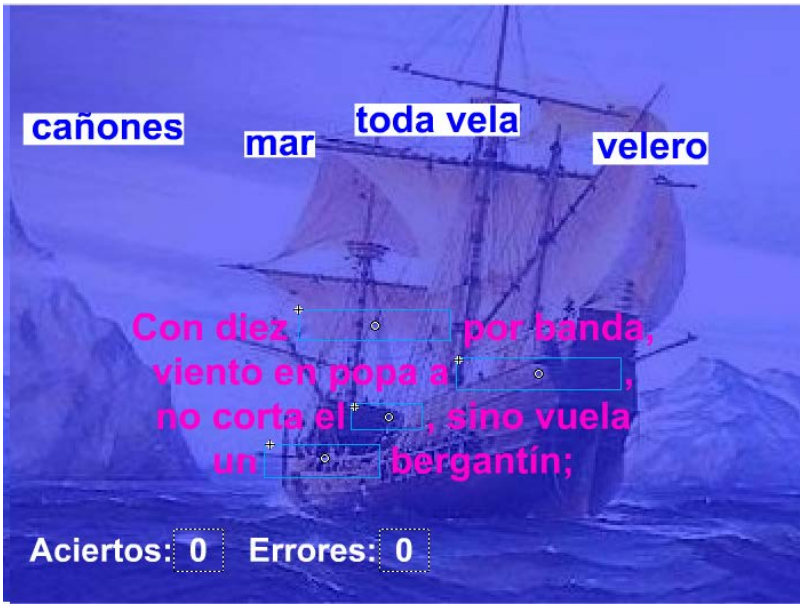

Imagen 1

En este primer paso crearemos todos los objetos de la pantalla: imagen de fondo, palabras, texto para los contadores y aciertos. Recordemos lo que en otros tutoriales hemos estudiado:

- Crear variables e inicializarlas.
- Crear cajas de texto dinámico y enlazarlas con las variables.
- Convertir texto a gráfico y a clip de película.

Observaciones:

La imagen de fondo (barco) se ha colocado en una capa inferior, y se ha convertido en clip de película para añadirle efecto de transparencia (alpha) y tintado mediante el panel de propiedades.

2.- Creación de los clips de película y sus nombres de instancia.

Una vez que hemos creado los elementos del escenario ahora vamos a convertir cada palabra a clip de película y a continuación activamos el panel de propiedades para dar a cada uno de los clips un **nombre de instancia**.

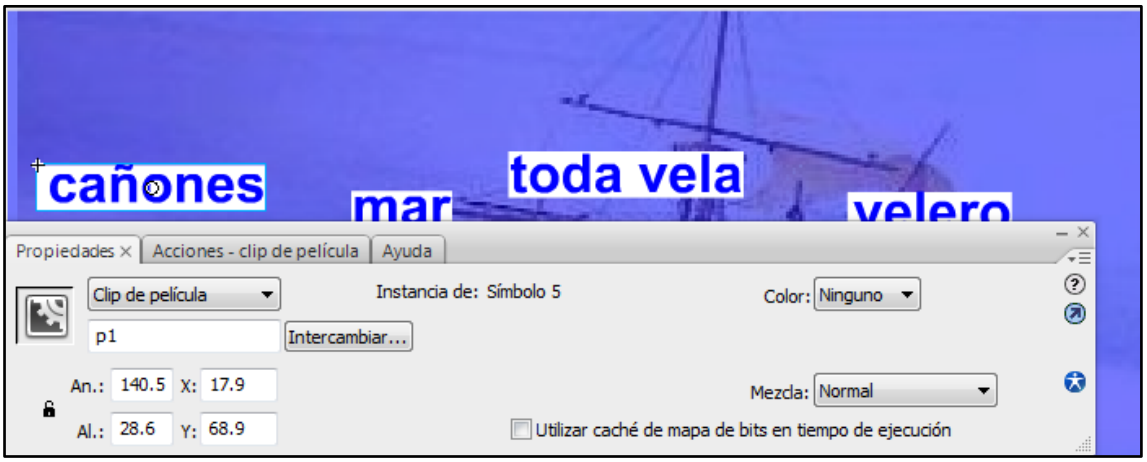

Imagen 2

En la imagen 2 podemos observar cómo hemos seleccionado el clip de película que contiene la palabra *cañones* y le hemos puesto como nombre de instancia **p1.** Haremos lo mismo con los clips de película restantes (p2, p3, p4…).

Pero aún necesitamos más clips de película. **Necesitamos saber cuándo una palabra colisiona con el hueco donde debe ir**. La forma más fácil es hacer una copia de cada clip y arrastrarlo al hueco adecuado. Cada una de estas copias de clip duplicados tendrán sus nombres de instancia. Podemos elegir cualquier nombre. En nuestro caso se ha elegido p11, p22, p33. Es decir el clip p1 ha de colisionar con el de p11 y así sucesivamente.

Una vez tengamos los clips duplicados les haremos invisibles ya que serán los huecos del texto. Para ello seleccionamos cada clip de película y mediante el panel de propiedades asignamos la propiedad **alfa** a 0 como podemos observar en la imagen 3. Observa la selección del hueco (clip de película p11).

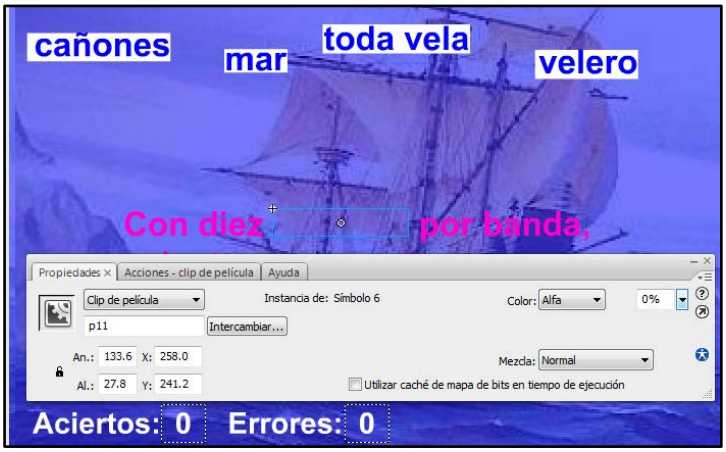

Imagen 3

3. Inyectando el código necesario.

```
on (press) {// cuando se presiona el botón izquierdo del ratón
    _root.p1.startDrag();// empezar a arrastrar


}

on (release) {// cuando se suelta el botón izquierdo del ratón

    if (_root.p1.hitTest(_root.p11)) { //¿hay colisión entre esos dos clips?
        _root.aciertos++; //aumenta el número de aciertos.
        _root.p1.stopDrag();// parar de arrastrar
        _root.p1._x = _root.p11._x;// la coordenada x de la primera palabra es igual a la del clip del hueco
        _root.p1._y = _root.p11._y;// la coordenada y de la primera palabra es igual a la del clip del hueco
        _root.p1._name = "p1fin";// el nombre del clip cambia para que ya no pueda ser arrastrado.

    } else if (this._name<>"p1fin") {//si el nombre de clip es diferente a p1fin
        _root.errores++;//aumnta errores
        _root.p1._x=random(500);//mueve la palabra a una coordenada x aleatoria
        _root.p1._y=random(200);//mueve la palabra a una coordenada y aleatoria
        _root.p1.stopDrag();// parar de arrastrar
    }

}
```

Imagen 4

El código que podemos observar en la imagen 4 lo introduciremos en cada uno de los clips que deben ser arrastrados. Pero no en los *clips hueco.*

Los comentarios sobre la función de cada acción están en color rosa. Recordemos que para poder poner comentarios en el código solamente es necesario escribir al principio del comentario dos barras inclinadas //

El código están distribuido en dos eventos de ratón: **on (press) {}** y **on (release) {}.** En el primero, cuando se presiona el ratón, hacemos que el clip empiece a ser arrastrado.

Pero el segundo evento es el que realiza todas las funciones restantes y ocurrirá cuando el usuario suelte el botón del ratón. En este preciso momento se pasa a una sentencia condicional **if () {}**. Entre paréntesis se pone la condición y entre las llaves lo que se ejecutará si se cumple dicha condición.

Como podemos observar la condición evaluada es **si el clip p1 colisiona con el p11**. Utilizamos **hitTest()** para comprobarlo. Y si es cierto que hay colisión se ejecutamos las acciones para:

- Aumentar el número de aciertos
- Hacer que se quede quieto el clip en las mismas coordenadas que el clip del hueco.
- Cambiamos el nombre de la instancia del clip para que no pueda ser arrastrado nuevamente.

Siguiendo con el código también evaluamos lo que ocurrirá si no se produce la colisión, es decir si soltamos el clip en cualquier otro sitio del escenario. Po ello utilizamos la sentencia: **else if () {}**, que significa: *en caso contrario si ocurre otra condición…*

Y esta condición nos servirá para saber si el usuario soltó el botón en el clip P1 pero sin colisionar con el clip hueco que le correspondía (p11). En este caso se ejecutarían las siguientes acicones:

- Aumentar el número de errores.
- Enviar el clip aleatoriamente a un lugar distinto de la pantalla para dar la posibilidad de repetir la interacción con este clip.
- Dejar quieto el clip.

Para finalizar recordemos que para que las variables aciertos y errores funcionen, antes hay que inicializarlas y estas acciones las introduciremos en el primer fotograma de la película:

- _root.acciones=0;
- _root.aciertos=0;

4.- Ejercicio propuesto.

Se propone crear un ejercicio semejante al anterior pero combinando texto con imagen y añadiendo al menos una de las funcionalidades siguientes:

- Control de tiempo
- Límite de errores
- Refuerzo final una vez resulto el ejercicio completo.

Suerte y nos vemos en el foro.