

- **explosion.visible = false;**
explosion.loop = false;
Al momento de crear la explosión, se especifica que la explosión no se vea y que además no es una animación de *loop*.
- **addEventListener(EnterFrameEvent.ENTER_FRAME, checkCollisions);**
Se crea un evento que revisa en cada *frame* si existen colisiones entre cada uno de los pájaros y el husky.
- Y por supuesto, se escribe el método de este evento, el cual:
 - **if(badBird1.bounds.intersects(husky.bounds))**
Revisa si alguno de los pájaros tiene colisión con el husky. Recuerda que se está revisando si el rectángulo (en este caso imaginario) que contiene a los objetos relacionados, llega a tocarse en algún momento.
 - **TweenLite.to(this, 0.5, { scaleX:1.2, scaleY:1.2, onComplete:stabilizeScreen });**
Si existe una colisión, lo primero es hacer crecer la pantalla en una escala proporcional de 1.2 en los ejes X y Y. Esto es un efecto que puede desorientar al jugador, puedes experimentar áreas de oportunidad aquí. En este mismo enunciando, se revisa cuando el tween termina para invocar la función que restablece la escala a 1. Esta función se llama **stabilizeScreen**.
 - **explosion.x = badBird1.x;**
explosion.y = badBird1.y;
Después se posiciona el movieclip de la explosión en el punto exacto donde se encontraba el pájaro que chocó con el perro.
 - **TweenLite.killTweensOf(badBird1);**
Se invoca esta función para desactivar el tween restante del objeto, pues no necesitas que siga.
 - **badBird1.reuseBirds();**
Con esta línea se mueve al pájaro que colisionó a su posición de inicio invocando la función **reuseBirds()**;
 - **explosion.visible = true;**
explosion.currentFrame = 0;
Starling.juggler.add(explosion);
Se activa la visibilidad de la explosión, luego verificamos que el frame de la animación de la explosión sea el primero de la secuencia usando el comando **currentFrame**, y finalmente se agrega el objeto al juggler para que inicie la animación. Este proceso es sumamente rápido.
 - **explosion.addEventListener(Event.COMPLETE, explosionComplete);**
Al final de esta función, se coloca un evento que verifica cuando la secuencia de la animación termina, de manera que se pueda reciclar el objeto con el método **explosionComplete()**.
- **private function explosionComplete(e:Event):void**
En este método, simplemente se desactiva el evento que verifica el status de la explosión, luego desactiva la visibilidad y finalmente quita al objeto del juggler.