



6-3-2018

Proyecto Unidad 1 – Unidad 2

SERPIENTES Y ESCALERAS

MANUAL TÉCNICO

Paola Guadalupe Herrera Quiroz

José Leonardo Amaya Escobedo

Índice

Introducción	2
Portada(Frame 1)	3
Jugadores(Frame 2).....	4
Juego Serpientes Y Escaleras(Frame 3)	6
Dado(Frame 3)	7
Movimiento de los jugadores(Frame 3)	8
Timer(Frame 3)	10
Resultados(Frame 4)	11
Guardar (Frame 4)	12
Conclusión	13

Introducción

En este manual técnico, se dan las especificaciones de cómo se ha realizado y codificado este proyecto que es parte de la unidad 1 y de la unidad 2 de la materia de Herramientas Multimedia. Primero se debe documentar el código dentro del mismo para ir teniendo la noción de que es lo que se está realizando en cada línea o en un conjunto de líneas en específico. Después de eso, en el manual técnico se da la idea en general de fragmentos completos de nuestro código. En este caso, es el juego de serpientes y escaleras, se explica el movimiento de cada peón, así como cuando llega a alguna escalera o serpiente.

Portada(Frame 1)

Para las animaciones de la portada, se convirtieron las palabras o frases en clips de película, para poder usar las fuentes que tenemos y darles los tweens a cada una. Algunas son con respecto al eje y y otras con respecto al x

```

1  stop();
2  import fl.transitions.TweenEvent;
3  import fl.transitions.easing.*;
4  import flash.events.MouseEvent;
5  import fl.transitions.Tween;
6
7
8  //se declaran las variables de los elementos que van a tener Tweens en la portada
9  var paola:Tween= new Tween(paola_mc,"x",Elastic.easeIn,-700,225,3,true);//Movimiento del movie clip Paola
10 var leo:Tween= new Tween(leo_mc,"x",Elastic.easeIn,1400,245,3,true);//Movimiento del movie clip Leo
11 var serpi:Tween= new Tween(serpi_mc,"y",Back.easeIn,-200,100,4,true);//Movimiento del movie clip Serpientes
12 var ymc:Tween= new Tween(y_mc,"y",Back.easeIn,-200,120,4,true);//movimiento del movie clip Y
13 var esca:Tween= new Tween(esca_mc,"y",Back.easeIn,-200,170,4,true);//movimiento del movie clip Escalera
14 var hm:Tween= new Tween(hm_mc,"x",Elastic.easeIn,-500,330,3,true);//movimiento del movie Clip Herramientas Multi
15 var iti:Tween= new Tween(iti_mc,"x",Elastic.easeIn,1400,430,3,true);//Movimiento del movie Clip ITI
16 var inicio:Tween= new Tween(inicio_btn,"y",Back.easeOut,1000,525,4,true);//Movimiento del botón de inicio
17
18 // ===== SIGUIENTE FRAME =====
19 function Finicio (event:MouseEvent):void{
20     gotoAndPlay(2);
21 }
22 inicio_btn.addEventListener(MouseEvent.CLICK,Finicio);

```

Jugadores(Frame 2)

Una vez que ya se ha presionado el botón de play, que es el que nos lleva al siguiente fotograma, tenemos una caja de texto y tres opciones de avatar. Para elegir alguna imagen, se debe ingresar el nombre de cada jugador. Este juego ha sido creado para exactamente 3 personas, si no se cubre esa cantidad, nunca aparecerá la opción de pasar al frame del juego.

Se debe ir ingresando nombre, y seleccionando el avatar que se quiere según el jugador. Cada uno de ellos, al ser elegidos desaparece para que no exista la opción de que se seleccione el mismo.

En este frame se guardarán los nombres y el avatar seleccionado de cada jugador dentro de un Array, el cual nos ayudará más adelante dentro del juego a saber el turno de cada uno de estos 3 jugadores.

```
var nombresArray:Array = new Array();
var numeroArray:Array = new Array();
var avatarArray:Array = new Array();
var jugArray:Array = new Array();
var numJug:int = 0;
var num:int = 1;
var avatar:String;
nombres_txt.restrict = "A-Z || a-z";
numero.text = String(num);
jugadores.visible = false;
siguiente_btn.visible = false;

//array de los nombres
//array del numero de jugador
//array del avatar que eligió
//array de la posicion de cada jugador
//variable del numero de jugador
//para el numero de jug de la etiqueta
//para guardar el nombre del avatar
//restringir caracteres en nombres
//la etiqueta dependiendo el num de jug
//lista de jugadores con avatar invisible
//boton de pasar al juego
```

Jugadores(Frame 2)

Hay 3 funciones las cuales son llamadas con cada uno de los botones. Estos botones son los avatares que estan en el escenario.

Dentro de cada una de estas funciones se hace el procedimiento de “Pusheo” a lo que quiere decir que los datos los guardamos dentro de distintos arrays.

Hay 4 arrays en total

- 1.- Para guardar el nombre de los jugadores
- 2.- Guardar el numero del jugador(Quien eligió primero el avatar)
- 3.-Se guarda cuál fue el avatar que se eligió
- 4.-Guardamos la posición que tendrá el avatar. Esto porque después de que se elijan los 3 avatar nos saldrá una ventana mostrando en orden de elección a los avatares con su nombre.

Esto mismo para cada una de las 3 funciones.

```
function Fjugador1(event:MouseEvent):void           //cuando se presiona el avatar
{
    if (nombres_txt.text == "")                     //validar que haya un nombre tecleado
    {
        nombres_txt.text = "Ingresa nombre";
    }
    else if (nombres_txt.text=="Ingresa nombre")    //que"ingrese nombre"no lo tome como nomb
    {
        nombres_txt.text = "";
    }
    else if (numJug==2)                             //si ya es el tercer jugador, te pasa a la
    {                                                 //lista de los jugadores
        numero.visible = false;
        nombres_txt.visible = false;
        nombresArray.push(nombres_txt.text);        //el nombre se ingresa al array
        numJug++;                                   //se incrementa el num de jugador
        jug1.visible = false;                       //se pone invisible ese avatar
        numeroArray.push(numJug);                   //se agrega el num de jug al array
        ingresa.visible = false;
        trace(nombresArray);
        trace(numeroArray);
        jugadores.visible = true;
        siguiente_btn.visible = true;
        jugador1.text = String(nombresArray[0]);    //se imprimen los nombres de los jugadores
        jugador2.text = String(nombresArray[1]);    //en orden como se ingresaron
        jugador3.text = String(nombresArray[2]);
        avatar1.x = 508;                            //posiciones de los avatars en x, todas
        avatar2.x = 508;                            //iguales porque estan uno abajo del otro
        avatar3.x = 508;
        avatar = "rosa";                             //se le asigna nombre al avatar
        avatarArray.push(avatar);                   //se agrega el nomb del avatar al array
        trace(avatarArray);
    }
}
```

Juego Serpientes y Escaleras (Frame 3)

Aquí se muestran todas las variables que se usan a lo largo del frame 3 el cual ya sería el juego tal cual. La lógica que se utilizó para este juego fue sencillamente el de recorrer un Array de 101 posiciones, en dónde 100 de estas posiciones son instancias de las imágenes que están posicionadas en el escenario. Estas imágenes son las casillas de todo nuestro escenario. Y dentro de este mismo Array hay una posición de más, la cual se utilizó para un mejor manejo de este recorrimiento.

```
nombre.text = String(nombresArray[cont]); //Escribir el nombre del jugador en juego
var cont:int = 0; //Contador para el orden de los nombres
var contT:int=0; //contador del timer
var movJugador1:Tween; //Variable para hacer el movimiento del cambio de casilla del jugador en turno
var numRandom:int; //Numero random del 1 al 6
var casillas:int = 0; //Variable para guardar las casillas recorridas por el jugador 1
var casillas1: int = 0; //Variable para guardar las casillas recorridas por el jugador 2
var casillas2: int = 0; //Variable para guardar las casillas recorridas por el jugador 3
var Vcasillas: Array = new Array(casillas,casillas1,casillas2); //Vector en donde se guardan las variables de las casilla.
var cero:String; //para que las tarjetas empiecen en 1 y no en cero
var timer:Timer = new Timer(1000,contT++); //Timer para que haya un tiempo de espera cuando se encuentra Serpientes o es:
var variablepapu:int; //Variable que se va a ir aumentando cada segundo en el Timer
var resta:int=0; //Variable obtener el resultado que sobra en caso de que se obtenga un valor mayor a 100
var rebote:Boolean; //Variable para hacer el efecto de rebote en caso de que pase de 100 el valor de casillas recorridas

//Array de todas las tarjetas para usar su ubicación más adelante.
var Vtarjetas:Array = [cero,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,t20,t21,t22,t23,t24,t25,
t26,t27,t28,t29,t30,t31,t32,t33,t34,t35,t36,t37,t38,t39,t40,t41,t42,t43,t44,t45,t46,t47,t48,t49,t50,
t51,t52,t53,t54,t55,t56,t57,t58,t59,t60,t61,t62,t63,t64,t65,t66,t67,t68,t69,t70,t71,t72,t73,t74,t75,
t76,t77,t78,t79,t80,t81,t82,t83,t84,t85,t86,t87,t88,t89,t90,t91,t92,t93,t94,t95,t96,t97,t98,t99,t100];
```

```
//Imagen jugador1;
if (avatarArray[0] == "rosa")
{
    avatar1.y = 168;
    trace("ya llegue");
    jugArray.push(j1);
}
else if (avatarArray[0]=="verde")
{
    avatar2.y = 168;
    jugArray.push(j2);
}
else if (avatarArray[0]=="morado")
{
    avatar3.y = 168;
    jugArray.push(j3);
}
//Imagen jugador2
if (avatarArray[1] == "rosa")
{
    avatar1.y = 306;
    trace("ya llegue");
    jugArray.push(j1);
}
else if (avatarArray[1]=="verde")
{
    avatar2.y = 306;
    jugArray.push(j2);
}
else if (avatarArray[1]=="morado")
{
    avatar3.y = 306;
    jugArray.push(j3);
}
```

```
//Si la posición 0 del array de los avatar
//es rosa, se le asigna cierta posición en
//la lista de los jugadores, que sería para
//el primer jugador, o sea el primero en la lista

//se guarda la instancia de la figura que se
//va a mover en el tablero, para usar el array
//SE HACE EN TODOS LOS CASOS DE NOMBRES DE AVATAR
//Y POSICIONES EN EL ARRAY
```

Dado(Frame 3)

La primera función con la que nos topamos en el código es el del dado. Esta función se ejecuta cuando se presiona un dado que está en el escenario. La función del dado es el de conseguir un numero aleatorio que esté en el rango del 1 al 6 y a su vez llamar a otra función llamada “moverJugador” la cual consiste en los desplazamientos que hay dentro del tablero.

```
function NumeroRandom(event:MouseEvent):void
{
    numRandom = (Math.random()*6)+1;//Se asigna un número hasta el 6 y a e
    //numRandom=100
    trace(numRandom);//Se muestra el numero aleatorio en la consola
    txt_dado.text = numRandom + "";//Se muestra el numero random en pantall
    moverJugador();//Se llama a la funcion en donde se ejecutan los movimi
}
btn_dado.addEventListener(MouseEvent.CLICK,NumeroRandom); //Boton del dado
```


Movimientos De Los Jugadores (Frame 3)

La siguiente función es la de “moverJugador” en esta función lo que se hace básicamente es tomar el valor del número random que obtuvimos en la tirada y este número sumárselo a una variable que hay por cada jugador la cual consiste en ir sumando cada uno de los números random que van obteniendo. Así mismo con esta variable sabremos la casilla actual en la que está el jugador.

También dentro de esta función existen las condiciones de los casos “especiales”. El primer caso especial con el que nos encontramos es con el caso de que si el número de casillas del jugador actual es mayor a 100 se ejecutará a lo que nosotros llamamos **Un rebote**. Este rebote es el de retroceder la cantidad de casillas por las cuales se pasó de 100.

Ejemplo.- En caso de que nosotros estemos posicionados en la casilla 95 y nos sale una tirada de 6. Nos quedaríamos en la casilla 101 pero esta no existe dentro del escenario así que lo que se hace es que se posicione en la casilla 99 ya que fue 1 por lo que se pasó de 100.

Solamente se gana cuando el número de casillas sea exactamente a 100

```
function moverJugador()
{
    Vcasillas[cont] += numRandom;
    trace("El numero del Array es "+Vcasillas[cont]);
    if (Vcasillas[cont]>100){
        rebote=true;
        resta=(Vcasillas[cont]-100);
        trace(resta);
        movJugador1 = new Tween(jugArray[cont], "y", None.easeOut, jugArray[cont].y, t100.y, 1, true);
        movJugador1 = new Tween(jugArray[cont], "x", None.easeOut, jugArray[cont].x, t100.x, 1, true);
        //timer.start();
        //===== REBOTE =====
        trace("Rebote")
        Vcasillas[cont] = (100-resta);
        trace(Vcasillas[cont]);
        movJugador1 = new Tween(jugArray[cont], "y", None.easeOut, jugArray[cont].y, Vtarjetas[Vcasillas[cont]].y, 1, true);
        movJugador1 = new Tween(jugArray[cont], "x", None.easeOut, jugArray[cont].x, Vtarjetas[Vcasillas[cont]].x, 1, true);
    }
    if (cont==2){
        cont=0;
        nombre.text = String(nombresArray[cont]);
    }
    else {
        cont++;
        nombre.text = String(nombresArray[cont]);
    }
}
else{
    rebote=false;
    movJugador1 = new Tween(jugArray[cont], "y", None.easeOut, jugArray[cont].y, Vtarjetas[Vcasillas[cont]].y, 1, true);
    movJugador1 = new Tween(jugArray[cont], "x", None.easeOut, jugArray[cont].x, Vtarjetas[Vcasillas[cont]].x, 1, true);
}
```

Movimientos De Los Jugadores (Frame 3)

```

if (Vcasillas[cont]==7 ){
    timer.start();
}
else if (Vcasillas[cont] == 97){
    timer.start();
}
else if (Vcasillas[cont] == 88){
    timer.start();
}
else if (Vcasillas[cont] == 65){
    timer.start();
}
else if (Vcasillas[cont] == 62){
    timer.start();
}
else if (Vcasillas[cont] == 77){
    timer.start();
}
else if (Vcasillas[cont] == 54){
    timer.start();
}
else if (Vcasillas[cont] == 18){
    timer.start();
}
else if (Vcasillas[cont] == 21 ){
    timer.start();
}
else if (Vcasillas[cont] == 46 ){
    timer.start();
}
else if (Vcasillas[cont] == 100 && rebote==false){
    timer.start();
}

if (cont==2){
    cont=0;
    nombre.text = String(nombresArray[cont]);
}

```

Y los otros casos especiales es el de cada uno de las serpientes y escaleras que existe en el escenario. En caso de que llegue a alguno de estos casos especiales se va a inicializar un timer el cual estará a cargo de los movimientos de cada uno de esos casos especiales.

Timer (Frame 3)

```
function Ftiempo (tiempoevent:TimerEvent):void{
    variablepapu++;
    trace(variablepapu)
    trace("La casilla es "+Vcasillas[cont]);
    btn_dado.visible = false;
    if (variablepapu == 3){
        timer.stop();
        trace("ya pare");
        //===== ESCALERA 7 =====
        if (Vcasillas[cont]==7){
            trace("Obvio papu");
            movJugador1 = new Tween(jugArray[cont], "y", None.easeOut, jugArray[cont].y, t14.y, 1, true);
            trace("Me voy para arriba")
            Vcasillas[cont] = 14
            variablepapu=0;
            btn_dado.visible = true;
        }
        //===== SERPIENTE 97 =====
    }
}
```

Cómo se dijo anteriormente, cada uno de los casos especiales hará que empiece a correr un timer. Este timer es el responsable de que se hagan los movimientos “Tweens” correctos para cada uno de los casos. Por ejemplo, vemos en el código de arriba que en caso de que caiga en la casilla 7 va a subir hasta la casilla 14 y que ahora el numero de la variable de las casillas recorridas será 14 ya que está en la casilla 14.

También dentro de cada una de estos casos ponemos en invisible el dado. Esto para que se espere en terminar el tween del traslado de una casilla a otra y que no pueda tirar el siguiente jugador hasta que esto termine.

Esta misma condición es la misma para cada uno de los casos “especiales”

Resultados(Frame 4)

Cuando algún jugador llega exactamente a la meta, se hace una comparación de la posición en las casillas de cada jugador y la que vaya siendo mayor es la que va quedando en primer lugar, se comparan las otras dos y es la que queda en segundo y la otra en tercero. Esto se realizó con condiciones IF, en las cuales llevaba ELSE IF, si la primera no se cumple, se pasa a verificar la que sigue y así hasta encontrar cual se va a hacer. Dentro de ésta se usa el Array que nos guardó anteriormente los nombres de los jugadores, y se imprime en el texto dinámico el nombre.

Si la posición 0 de los jugadores, es decir, el primer jugador que se ingresó, es la que quedó en primer lugar, en la etiqueta de primer lugar se imprime el nombre del Array en la posición 0, porque están relacionados. Ahí mismo, se le da valor a la posición y de las imágenes del avatar, ya que en x estaba predeterminada la misma, porque estarán en una columna. Las posiciones x y y se asignaron de acuerdo a la posición de la etiqueta del nombre.

```
jugArray[0].x=565;
jugArray[1].x=565;
jugArray[2].x=565;

if (Vcasillas[0] > Vcasillas[1] && Vcasillas[0] > Vcasillas[2]){
    primero_txt.text=String(nombresArray[0]);
    jugArray[0].y=195;

    if (Vcasillas[1] > Vcasillas[2]){
        segundo_txt.text=String(nombresArray[1])
        tercero_txt.text=String(nombresArray[2])
        jugArray[1].y=330;
        jugArray[2].y=455;
    }
    else{
        segundo_txt.text=String(nombresArray[2])
        tercero_txt.text=String(nombresArray[1])
        jugArray[1].y=455;
        jugArray[2].y=330;
    }
}
```

Guardar(Frame 4)

Se programa para que el botón de “guardar” cree un archivo PDF mediante un FileReference, que es para una variable que se podrá guardar con cualquier extensión. En la variable, nuestro caso es archivo, se utiliza un save, que es la que permite el cambio de los textos al documento, en la cual los parámetros serán: las instancias de textos que se quieran exportar y el nombre que se le dará predeterminado al archivo, junto con la extensión.

```
//===== GUARDAR =====
import flash.net.FileReference;           //Permite guardar el texto insertado en un TextInput
function Fguardar(event:MouseEvent):void //funcion para guardar de acuerdo al boton
{
    var archivo:FileReference = new FileReference(); //se crea un archivo FileReference
                                                    //que permite guardar textos de flash
                                                    //a algun documento, en este caso PDF
    archivo.save ("Resultados"+"\r\n"+"Primero:"+" "+(primero_txt.text)+"\r\n"+"Segundo:"+" "+(s:
                                                    // \r\n brinca al siguiente renglon
                                                    //Se pone entre comillas lo que se
                                                    //quiere agregar + la instancia de
                                                    //donde se tiene el texto que se quiere
                                                    //traspasar a otro archivo
}
guardar_btn.addEventListener(MouseEvent.CLICK, Fguardar);
```

Conclusión

Se elabora el manual técnico para explicar de una manera más general el cómo y porqué de nuestras líneas de código. Se detallan por fragmentos desarrollados y el método que se usó, qué es lo que hacen esas líneas y como. Una de las partes más importantes del proyecto es saber colaborar en equipo, el manual a la vez es de ayuda para que nuestro colaborador entienda que estamos haciendo e incluso nosotros mismos para futuros cambios o modificaciones.